

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Саратовский государственный технический университет
имени Гагарина Ю.А.»

Филиал федерального государственного бюджетного образовательного
учреждения высшего образования
«Саратовский государственный технический университет
имени Гагарина Ю.А.» в г. Петровске



УТВЕРЖДАЮ
Директор филиала СГТУ
имени Гагарина Ю.А. в г.Петровске
Е.А.Бесшапошникова
«30» июня 2025 г.

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ

по дисциплине

МДК 04.02. «Обеспечение качества функционирования
компьютерных систем»

направление подготовки

09.02.07 «Информационные системы и программирование»

Методические указания рассмотрены
на заседании предметной (цикловой)
комиссии общепрофессиональных дисциплин
и профессиональных модулей
«16» июня 2025 года, протокол №13

Председатель ПЦК  /Ю.А.Табарова/

Петровск 2025

Пояснительная записка

Методические указания по выполнению практических работ подготовлены на основе рабочей программы учебной дисциплины МДК.04.02 «Обеспечение качества функционирования компьютерных систем», разработанной на основе ФГОС СПО по специальности 09.02.07 «Информационные системы и программирование» и соответствующих общих (ОК) и профессиональных (ПК) компетенций:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 4.1. Осуществлять установку, настройку и обслуживание программного обеспечения компьютерных систем

ПК 4.2. Осуществлять измерения эксплуатационных характеристик программного обеспечения компьютерных систем

ПК 4.3. Выполнять работы по модификации отдельных компонент программного обеспечения в соответствии с потребностями заказчика

ПК 4.4. Обеспечивать защиту программного обеспечения компьютерных систем программными средствами.

При выполнении практических работ студент должен *знать*:

- основные методы и средства эффективного анализа функционирования программного обеспечения;
- основные виды работ на этапе сопровождения ПО;
- основные принципы контроля конфигурации и поддержки целостности конфигурации ПО;
- основные средства и методы защиты компьютерных систем программными и аппаратными средствами

При выполнении практических работ студент должен *уметь*:

- подбирать и настраивать конфигурацию программного обеспечения компьютерных систем;
- проводить установку программного обеспечения компьютерных систем;
- производить настройку отдельных компонентов программного обеспечения компьютерных систем;
- измерять и анализировать эксплуатационные характеристики качества программного обеспечения;
- определять направления модификации программного продукта;

- разрабатывать и настраивать программные модули программного продукта;
- использовать методы защиты программного обеспечения компьютерных систем;
- анализировать риски и характеристики качества программного обеспечения;
- выбирать и использовать методы и средства защиты компьютерных систем программными и аппаратными средствами

Содержание практических работ определено рабочей программой и тематическим планированием, соответствует теоретическому материалу изучаемых разделов учебной дисциплины.

Объем практических работ по дисциплине определяется учебным планом по данной специальности.

Продолжительность практической работы – 2 академических часа. Перед проведением практической работы преподавателем организуется инструктаж, а по ее окончании – обсуждение итогов.

Комплект методических указаний по выполнению практических работ по дисциплине МДК.04.02 «Обеспечение качества функционирования компьютерных систем» содержит 52 практических занятия.

**Перечень практических работ
по дисциплине МДК.04.02 «Обеспечение качества функционирования
компьютерных систем»**

ПРАКТИЧЕСКАЯ РАБОТА № 1

Тема: Расчет сетевых графиков

ПРАКТИЧЕСКАЯ РАБОТА № 2

Тема: Расчет сетевых графиков

ПРАКТИЧЕСКАЯ РАБОТА № 3

Тема: Расчет сетевых графиков

ПРАКТИЧЕСКАЯ РАБОТА № 4

Тема: Расчет продолжительности работ и критического пути

ПРАКТИЧЕСКАЯ РАБОТА № 5

Тема: Расчет продолжительности работ и критического пути

ПРАКТИЧЕСКАЯ РАБОТА № 6

Тема: Вычисление детерминированных и вероятностных показателей времени

ПРАКТИЧЕСКАЯ РАБОТА № 7

Тема: Вычисление детерминированных и вероятностных показателей времени

ПРАКТИЧЕСКАЯ РАБОТА № 8

Тема: Сокращение времени и затрат

ПРАКТИЧЕСКАЯ РАБОТА № 9

Тема: Сокращение времени и затрат

ПРАКТИЧЕСКАЯ РАБОТА № 10

Тема: Работа с Календарем проекта в программе MS Project

ПРАКТИЧЕСКАЯ РАБОТА № 11

Тема: Работа с Календарем проекта в программе MS Project

ПРАКТИЧЕСКАЯ РАБОТА № 12

Тема: Определение состава задач проекта, структурирование списка задач, СДР-коды, ввод задач в программе MS Project

ПРАКТИЧЕСКАЯ РАБОТА № 13

Тема: Определение состава задач проекта, структурирование списка задач, СДР-коды, ввод задач в программе MS Project

ПРАКТИЧЕСКАЯ РАБОТА № 14

Тема: Определение состава задач проекта, структурирование списка задач, СДР-коды, ввод задач в программе MS Project

ПРАКТИЧЕСКАЯ РАБОТА № 15

Тема: Выполнение ресурсного и бюджетного планирования в программе MS Project

ПРАКТИЧЕСКАЯ РАБОТА № 16

Тема: Выполнение ресурсного и бюджетного планирования в программе MS Project

ПРАКТИЧЕСКАЯ РАБОТА № 17

Тема: Выполнение ресурсного и бюджетного планирования в программе MS Project

ПРАКТИЧЕСКАЯ РАБОТА № 18

Тема: Анализ и планирование рисков

ПРАКТИЧЕСКАЯ РАБОТА № 19

Тема: Анализ и планирование рисков

ПРАКТИЧЕСКАЯ РАБОТА № 20

Тема: Анализ и планирование рисков

ПРАКТИЧЕСКАЯ РАБОТА № 21

Тема: Анализ и планирование рисков

ПРАКТИЧЕСКАЯ РАБОТА № 22

Тема: Выявление первичных и вторичных ошибок

ПРАКТИЧЕСКАЯ РАБОТА № 23

Тема: Выявление первичных и вторичных ошибок

ПРАКТИЧЕСКАЯ РАБОТА № 24

Тема: Выявление первичных и вторичных ошибок

ПРАКТИЧЕСКАЯ РАБОТА № 25

Тема: Управление базовым планом с помощью инструментов в программе MS Project

ПРАКТИЧЕСКАЯ РАБОТА № 26

Тема: Управление базовым планом с помощью инструментов в программе MS Project

ПРАКТИЧЕСКАЯ РАБОТА № 27

Тема: Работа с отчетами и представлениями

ПРАКТИЧЕСКАЯ РАБОТА № 28

Тема: Работа с отчетами и представлениями

ПРАКТИЧЕСКАЯ РАБОТА № 29

Тема: Тестирование программных продуктов

ПРАКТИЧЕСКАЯ РАБОТА № 30

Тема: Тестирование программных продуктов

ПРАКТИЧЕСКАЯ РАБОТА № 31

Тема: Тестирование программных продуктов

ПРАКТИЧЕСКАЯ РАБОТА № 32

Тема: Тестирование программных продуктов

ПРАКТИЧЕСКАЯ РАБОТА № 33

Тема: Математическое моделирование методов шифрования

ПРАКТИЧЕСКАЯ РАБОТА № 34

Тема: Математическое моделирование методов шифрования

ПРАКТИЧЕСКАЯ РАБОТА № 35

Тема: Математическое моделирование методов шифрования

ПРАКТИЧЕСКАЯ РАБОТА № 36

Тема: Математическое моделирование методов шифрования

ПРАКТИЧЕСКАЯ РАБОТА № 37

Тема: Составление программ шифрования методом замены

ПРАКТИЧЕСКАЯ РАБОТА № 38

Тема: Составление программ шифрования методом замены

ПРАКТИЧЕСКАЯ РАБОТА № 39

Тема: Составление программ шифрования методом замены

ПРАКТИЧЕСКАЯ РАБОТА № 40

Тема: Составление программ шифрования методом замены

ПРАКТИЧЕСКАЯ РАБОТА № 41

Тема: Составление программ шифрования методом перестановки

ПРАКТИЧЕСКАЯ РАБОТА № 42

Тема: Составление программ шифрования методом перестановки

ПРАКТИЧЕСКАЯ РАБОТА № 43

Тема: Составление программ шифрования методом перестановки

ПРАКТИЧЕСКАЯ РАБОТА № 44

Тема: Составление программ шифрования методом перестановки

ПРАКТИЧЕСКАЯ РАБОТА № 45

Тема: Составление программ шифрования через картинку

ПРАКТИЧЕСКАЯ РАБОТА № 46

Тема: Составление программ шифрования через картинку

ПРАКТИЧЕСКАЯ РАБОТА № 47

Тема: Составление программ шифрования через картинку

ПРАКТИЧЕСКАЯ РАБОТА № 48

Тема: Составление программ шифрования через картинку

ПРАКТИЧЕСКАЯ РАБОТА № 49

Тема: Установка и настройка антивирусных программ

ПРАКТИЧЕСКАЯ РАБОТА № 50

Тема: Обнаружение вируса и устранение последствий его влияния

ПРАКТИЧЕСКАЯ РАБОТА № 51

Тема: Использование программ дефрагментации дисков

ПРАКТИЧЕСКАЯ РАБОТА № 52

Тема: Работа с программой восстановления файлов и очистки дисков

ИНСТРУКЦИИ ДЛЯ ОБУЧАЮЩИХСЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ

Прежде чем приступить к выполнению заданий, внимательно прочитайте данные рекомендации. Практические работы включают в себя задания следующих видов.

1. Работа за компьютером

В ходе выполнения практических работ студент должен:

- выполнять требования по охране труда
- соблюдать инструкцию по правилам и мерам безопасности в кабинете информационных технологий
- строго выполнять весь объем работы, указанный в задании
- соблюдать требования эксплуатации компьютерной техники (правила включения и выключения)
- предоставить отчет о проделанной работе по окончании выполненной работы, который должен содержать:

1. Название работы.
2. Цель работы.
3. Задание и его решение.
4. Вывод о проделанной работе.

Текст отчета по практической работе должен быть набран на компьютере шрифтом Times New Roman размером 14 пт. (при оформлении текста используется текстовый редактор MS Word). Шрифт, используемый в иллюстративном материале (таблицы и рисунки), рекомендуется уменьшить до 12 пт. Межстрочный интервал в основном тексте - полуторный. В иллюстративном материале межстрочный интервал рекомендуется сделать одинарным. Поля страницы должны быть: левое поле - 30 мм; правое поле – 15 мм; верхнее и нижнее поле - 20 мм.

Каждый абзац должен начинаться с красной строки. Отступ абзаца – 1,25 см от левой границы текста.

Студент должен выполнить практическую работу самостоятельно (или в группе, если это предусмотрено заданием). Практическая работа выполняется согласно заданию и методическим рекомендациям. После выполнения практической работы обучающийся самостоятельно себя контролирует путем ответов на вопросы. Результат работы представляется преподавателю в виде файла (файлов) в личном каталоге, защищается обучающимися.

По ходу выполнения работы при возникновении вопросов обучающийся может получить консультацию у преподавателя или самостоятельно воспользоваться лекционным материалом, рекомендуемой литературой.

2. Ответ на поставленные вопросы (с аргументацией)

Прочитайте вопрос и вникните в него.

Для удобства подчеркните ту, фразу, которая, по вашему мнению, является главной. Это поможет вам быстрее сориентироваться при ответе на вопрос.

Если вы считаете, что можете ответить на вопрос без помощи лекции и дополнительной литературы – приступайте. Если же вопрос заставляет вас

сомневаться, откройте лекционную тетрадь (учебник или дополнительную литературу), прочитайте необходимый пункт, вникните в содержание и после этого приступайте за работу.

ГЛАВНОЕ! Не переписывайте отрывки лекции в рабочую тетрадь! Четко отвечайте на ПОСТАВЛЕННЫЙ вопрос!

Не забудьте привести аргументацию (обоснование) вашей позиции, если вопрос предполагает личностное отношение к проблеме.

3. Заполнение таблиц и схем

Прочитайте название таблицы или схемы.

Исходя из названия, вы поймете цель предстоящей работы.

Воспользуйтесь материалами лекций или другими источниками, чтобы заполнить таблицу (схему).

Используйте цветные графические материалы для выделения строк, столбцов или элементов схем.

Особое внимание обращайтесь на четкость при отборе материала: делайте записи кратко и четко!

4. Поиск информации в сети — использование web-браузеров, баз данных, пользование информационно-поисковыми и информационно-справочными системами, автоматизированными библиотечными системами, электронными журналами. Поиск и обработка информации включает подготовку фрагмента практического занятия.

ПРАКТИЧЕСКАЯ РАБОТА № 1

Тема: Расчет сетевых графиков

Цель работы: научиться использовать методы сетевого планирования для решения задач управления проектами, например, контроля сроков выполнения работ, выявления критических операций.

Оборудование: ПК, интернет, программное обеспечение – операционная система MS Project, MS Word, инструкции по выполнению работы.

Справочный материал:

Понятие сетевого графика — диаграммы, представляющей последовательность и взаимосвязь работ или задач в проекте. Основные компоненты: вершины (узлы) — события или этапы проекта, рёбра (дуги) — задачи или работы, связывающие события.

Типы сетевых графиков:

1. *Диаграмма предшествующих работ* — узлы представляют работы, а рёбра — зависимости между работами.

2. *Диаграмма стрел* — узлы представляют события, а рёбра — работы.

Метод критического пути (Critical Path Method, CPM) — определение наиболее длительного пути через сетевой график, который определяет минимальное время завершения проекта. Задачи на критическом пути не имеют временного резерва.

Расчёт временных характеристик сетевого графика:

1. *Раннее начало (ES) и ранний конец (EF)* — определяют, когда задача может начаться и закончиться при условии, что все предшествующие задачи завершены как можно раньше.

2. *Позднее начало (LS) и поздний конец (LF)* — определяют, когда задача может начаться и закончиться без задержки проекта.

3. *Временной резерв (Float)* — разница между поздним и ранним началом или концом задачи, показывает, сколько времени можно задержать задачу без влияния на проект.

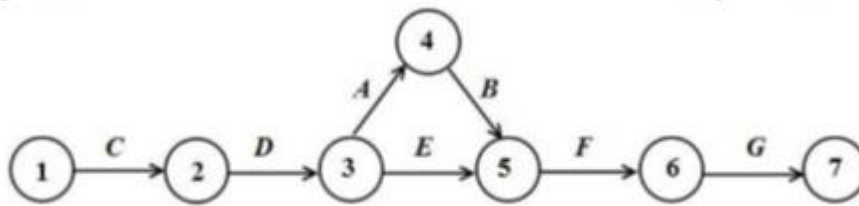
Содержание работы:

Задание 1. Необходимо построить сетевую модель программы опроса общественного мнения, включающая разработку (А; 1 день) и распечатку анкет (В; 0.5 дня), прием на работу (С; 2 дня) и обучение персонала (D; 2 дня), выбор опрашиваемых лиц (Е; 2 дня), рассылку им анкет (F; 1 день) и анализ полученных данных (G; 5 дней).

Решение.

Из условия задачи нам известно содержание работ, но явно не указаны взаимосвязи между работами. Поэтому для их установления необходимо проанализировать смысл каждой конкретной работы и выяснить, какие из остальных работ должны ей непосредственно предшествовать. Исходной работой, начинающей сетевой график, в данном случае является «прием на работу» (С), поскольку все остальные работы должны выполняться уже принятыми на работу сотрудниками. Перед выполнением всех работ по опросу общественного мнения сотрудников необходимо обучить персонал (D). Перед тем как разослать анкеты (F), их надо разработать (А), распечатать (В) и

выбрать опрашиваемых лиц (Е), причем работу с анкетами и выбор лиц можно выполнять одновременно. Завершающей работой проекта является анализ полученных данных (G), который нельзя выполнить без предварительной рассылки анкет (F). В результате этих рассуждений сетевая модель имеет следующий вид.



Задание 2. Необходимо построить сетевую модель, включающую работы А, В, С, ..., L, которая отображает следующее упорядочение работ:

1) А, В и С – исходные операции проекта;

2) А и В предшествуют D;

3) В предшествует Е, F и H;

4) F и С предшествует G;

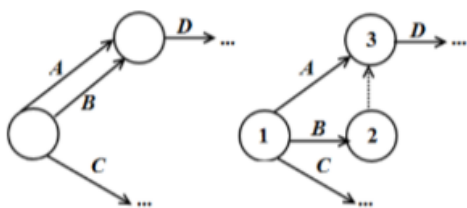
5) Е и H предшествуют I и J;

6) С, D, F и J предшествуют K;

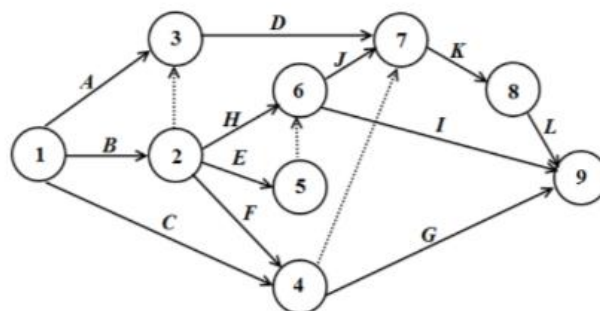
7) K предшествует L.

Решение.

В пункте 1 явно указано, что А, В и С являются исходными работами, поэтому изобразим их тремя стрелками, выходящими из исходного события 1 В пункте 2 условие означает, что стрелки работ А и В должны окончиться в одном событии, из которого выйдет стрелка работы D. Но поскольку стрелки работ А и В также и начинаются в одном событии, то имеет место параллельность работ, которая недопустима правилами построения сетевых моделей.



Для ее устранения введем дополнительное событие 2, в которое войдет работа В, после чего соединим события 2 и 3, в которые входят работы А и В пунктирной стрелкой фиктивной работы. В данном случае фиктивная работа (2,3) не соответствует никакой реальной работе, а лишь отображает логическую связь между работами В и D. Дальнейшее построение рассмотрим с помощью рисунка.



Согласно пункту 3 условия задачи из события 2 выходят три стрелки работ E, F и H. Согласно пункту 4 условия задачи стрелки работ C и F должны войти в общее событие, из которого выйдет стрелка работы G. Проблема с параллельностью работ E и H (пункт 5 условия задачи решается путем введения дополнительного события 5 и фиктивной работы (5,6)). Для отображения в сетевой модели пункта 6 условия задачи введем стрелки работ D и J в событие 7, а связь работ F и C с работой K отобразим с помощью фиктивной работы (4,7). Стрелки работ F и C нельзя было напрямую вводить в событие 7, потому что после них должна следовать работа G, которая с работами D и J никак не связана. Стрелка работы L выходит из события 8, т.е. после окончания работы K в соответствии с пунктом 7 условия задачи.

Поскольку в условии не указано, что работы L, I и G предшествуют каким-либо другим работам, то эти работы являются завершающими и их стрелки войдут в завершающее событие 9. Нумерацию событий проводят после построения сетевого графика, следя за тем, чтобы номер начального события каждой работы был меньше номера ее конечного события.

Задание 3. Построить сетевой график для максимальной ($t_{\text{пес}}$) продолжительности всех его работ, рассчитать наиболее ранние и наиболее поздние сроки наступления событий, найти критический путь, определить полные и независимые резервы времени всех работ и коэффициенты напряженности некритических дуг.

ПРАКТИЧЕСКАЯ РАБОТА № 2

Тема: Расчет сетевых графиков

Цель работы: научиться использовать методы сетевого планирования для решения задач управления проектами, например, контроля сроков выполнения работ, выявления критических операций.

Оборудование: ПК, интернет, программное обеспечение – операционная система MS Project, MS Word, инструкции по выполнению работы.

Содержание работ:

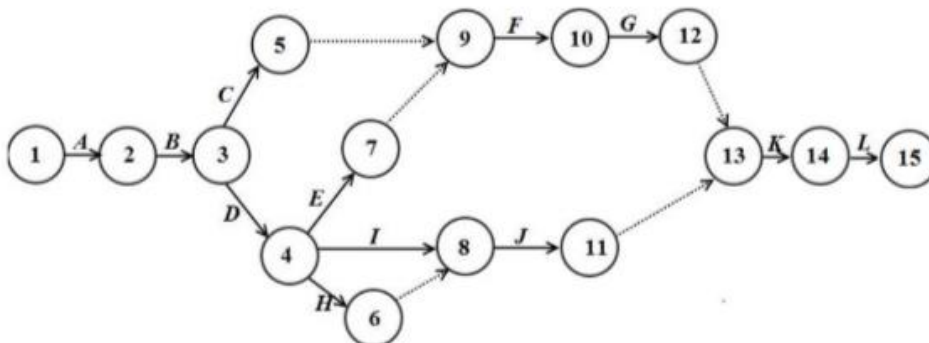
Задание 1. Промышленная фирма заключила контракт о производстве партии станков, предназначенных к использованию крупным предприятием обувной промышленности для массового производства обуви. Ниже перечислены операции, которые необходимо выполнить в процессе разработки и производства этих станков. Постройте сетевую модель.

Таблица. Список операций

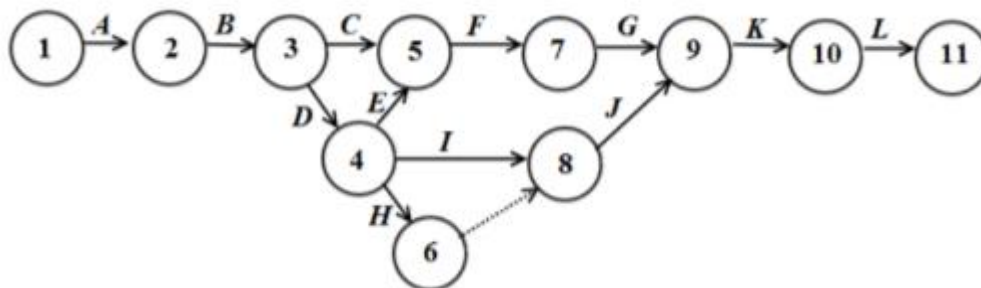
	Операции	Непосредственно предшествующая операция
A	Составление сметы затрат	–
B	Согласованные оценки	A
C	Покупка собственного оборудования	B
D	Подготовка конструкторских проектов	B
E	Строительство основного цеха	D
F	Монтаж оборудования	C,E
G	Испытания оборудования	F
H	Определение типа модели	D
I	Проектирование внешнего корпуса	D
J	Создание внешнего корпуса	H,I
K	Конечная сборка	G,J
L	Контрольная проверка	K

Решение.

Сетевая модель должна начинаться с единственного начального события, которое показано на рисунке кружочком, и заканчиваться единственным конечным событием. Построение модели мы начали с первого события. С этого события начинаются все операции, которым не предшествуют никакие виды работ. Начинать построение полезно с примерного эскиза будущей модели:



В соответствии с приведенной выше таблицей необходимо тщательно, переходя от одной операции к другой, проверить построенную в первом приближении модель. В случае необходимости следует провести его корректировку, а затем для совершенствования схемы построить новый. В данном случае можно исключить все фиктивные логические операции и оставить одну фиктивную операцию идентификации.



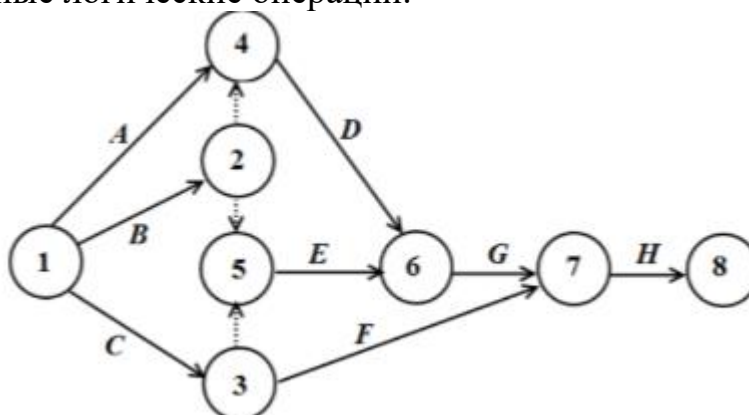
Задание 2. Промышленная фирма является участником другого проекта, детали которого приведены ниже. Постройте сетевую модель.

Таблица. Список операций

Операции	Непосредственно предшествующая операция
A	—
B	—
C	—
D	A,B
E	B,C
F	C
G	D,E
H	F,G

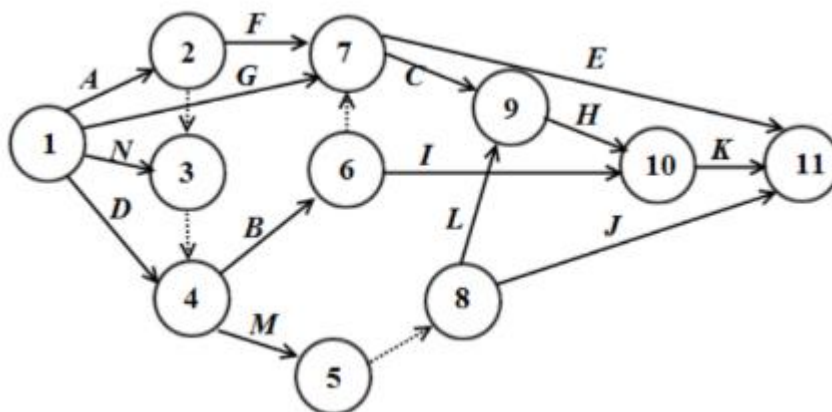
Решение.

Построение начинаем с начального события, обозначенного кружком 1. Из таблицы следует, что существуют три операции – А, В и С, которым не предшествует ни одна из операций. Поэтому из начального события выходят три стрелки. На первый взгляд таблица операций выглядит чрезвычайно простой, однако отразить присущую ей логику с помощью сетевой модели достаточно трудно, вследствие чего мы вынуждены использовать три фиктивные логические операции.



Задание 3. Используя данные о непосредственно предшествующих работах (таблица), перечислите работы, которые неверно отображены на сетевом графике (рисунок), устранимые найденные ошибки.

Название	Непосредственно предшествующие работы	Длительность, ед. времени
<i>A</i>	—	8
<i>B</i>	<i>D</i>	5
<i>C</i>	<i>B, F, G</i>	4
<i>D</i>	—	7
<i>E</i>	<i>B, F, G</i>	7
<i>F</i>	<i>A, N</i>	3
<i>G</i>	—	4
<i>H</i>	<i>C, L</i>	6
<i>I</i>	<i>B, G</i>	1
<i>J</i>	<i>I, M</i>	11
<i>K</i>	<i>H, I, M</i>	5
<i>L</i>	<i>I, M</i>	3
<i>M</i>	<i>D</i>	2
<i>N</i>	—	5



ПРАКТИЧЕСКАЯ РАБОТА № 3

Тема: Расчет сетевых графиков

Цель работы: научиться использовать методы сетевого планирования для решения задач управления проектами, например, контроля сроков выполнения работ, выявления критических операций.

Оборудование: ПК, интернет, программное обеспечение – операционная система MS Project, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Постройте сетевую модель проекта пусконаладки компьютерной системы, используя упорядочение работ из таблицы. Постройте линейную диаграмму Ганта и по ней определите критическое время.

Работа	Непосредственно предшествующая работа	Время выполнения
<i>A</i>	-	4
<i>B</i>	-	7
<i>C</i>	<i>A</i>	3
<i>D</i>	<i>B, C</i>	4
<i>E</i>	<i>D</i>	3
<i>F</i>	<i>E</i>	2
<i>G</i>	<i>B, C</i>	10
<i>H</i>	<i>F, G</i>	4

Задание 2. Постройте сетевую модель разработки и производства станков, используя упорядочение работ из таблицы. Постройте линейную диаграмму Ганта и по ней определите критическое время.

Работа	Непосредственно предшествующие работы	Время, ед. времени
<i>A</i> – составление сметы затрат	–	4
<i>B</i> – согласование оценок	<i>A</i>	7
<i>C</i> – покупка собственного оборудования	<i>B</i>	2
<i>D</i> – подготовка конструкторских проектов	<i>B</i>	3
<i>E</i> – строительство основного цеха	<i>D</i>	2
<i>F</i> – монтаж оборудования	<i>C, E</i>	6
<i>G</i> – испытание оборудования	<i>F</i>	3
<i>H</i> – определение типа модели	<i>D</i>	10
<i>I</i> – проектирование внешнего корпуса	<i>D</i>	8
<i>J</i> – создание внешнего корпуса	<i>H, I</i>	5
<i>K</i> – конечная сборка	<i>G, J</i>	4
<i>L</i> – контрольная проверка	<i>K</i>	8

ПРАКТИЧЕСКАЯ РАБОТА № 4

Тема: Расчет продолжительности работ и критического пути

Цель работы: получение навыков составления сетевых графиков, расчета раннего и позднего времени начала работ, нахождения критического пути.

Оборудование: ПК, интернет, программное обеспечение – операционная система MS Project, MS Word, инструкции по выполнению работы.

Справочный материал:

Метод критического пути (CPM) позволяет рассчитать продолжительность проекта дважды: по прямому пути (от даты начала) и по обратному пути (от даты окончания). Некоторые этапы расчёта:

1. Определение ранних сроков выполнения работ — раннего начала и раннего окончания.
2. Определение общей продолжительности проекта — она равна позднему окончанию завершающей работы.
3. Расчёт поздних сроков выполнения работ — позднего окончания и позднего начала каждой работы. Расчёт ведётся от позднего окончания завершающей работы, равного величине общей продолжительности проекта.
4. Определение резерва времени для каждой работы — он вычисляется как разница между поздним и ранним началом или поздним и ранним окончанием задачи.

Критический путь — это самая длинная последовательность работ, определяющая продолжительность проекта. Некоторые особенности расчёта:

1. Вычисление ранних временных параметров — прямой ход стартует от начальной операции и продвигается к финальной. Раннее начало операции соответствует максимальному из ранних окончаний всех предшествующих операций, для стартовой операции раннее начало принимается равным нулю. Раннее окончание рассчитывается сложением раннего начала и длительности операции.
2. Вычисление поздних временных параметров — обратный ход начинается от финальной операции и движется к стартовой. Позднее окончание операции равно минимальному из поздних начал всех последующих операций, для финальной операции позднее окончание равно раннему. Позднее начало получается вычитанием длительности операции из позднего окончания.
3. Выделение критического пути — его выделяют визуально на сетевой диаграмме, например, меняют цвет стрелок, обозначающих последовательность действий. Суммарное время для выполнения всех задач на критическом пути — это и есть срок проекта.

Содержание работы:

Задание 1. Составить проект Внедрение бухгалтерской системы для небольшой бухгалтерии, содержащей порядка 10 рабочих мест.

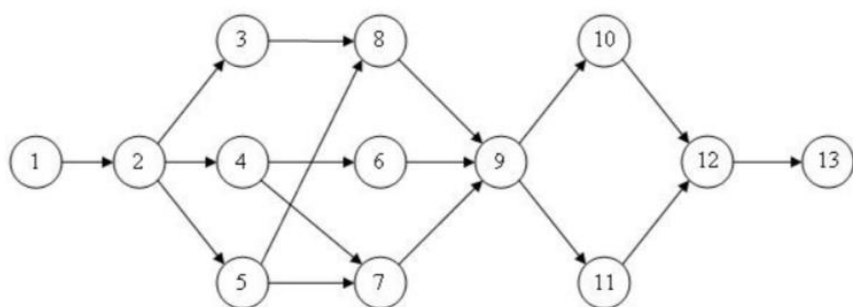
1. Составление перечня работ. В результате анализа выделяем перечень работ по проекту и оцениваем их длительность в днях. Результаты заносим в таблицу 1:

№	Название работы	Длительность
1	Начало проекта	-
2	Выбор системы	15
3	Приобретение программного обеспечения	7
4	Составление проекта сети	7
5	Приобретение компьютеров и сетевого оборудования	15
6	Обучение администратора и программиста	30
7	Монтаж локальной сети	20
8	Установка ПО на компьютеры	5
9	Установка сетевого ПО, настройка сети	25
10	Ввод начальных данных в информационную базу	40
11	Обучение персонала	30
12	Передача в эксплуатацию	5
13	Конец проекта	-

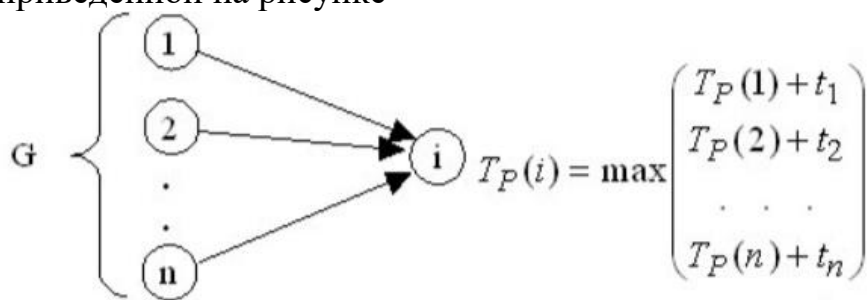
2. Определение взаимосвязей между работами. Для каждой работы из таблицы 1 требуется установить номера тех работ, до окончания которых она не может быть начата. Результат заносится в таблицу 2:

№	Название работы	Предшественники
1	Начало проекта	-
2	Выбор системы	1
3	Приобретение программного обеспечения	2
4	Составление проекта сети	2
5	Приобретение компьютеров и сетевого оборудования	2
6	Обучение администратора и программиста	4
7	Монтаж локальной сети	4; 5
8	Установка ПО на компьютеры	3; 5
9	Установка сетевого ПО, настройка сети	6; 7; 8
10	Ввод начальных данных в информационную базу	9
11	Обучение персонала	9
12	Передача в эксплуатацию	10; 11
13	Конец проекта	-

3. Составление сетевого графика работ. Каждая из работ таблицы 2 на сетевом графике обозначается кружком, в который заносится ее номер. Кружки соединяются стрелками. Стрелка соответствует одному из чисел столбца *Предшественники* и соединяет работу-предшественник с работой-последователем. Результат изображен на рисунке 1.



4. Вычисление раннего времени начала работ. В соответствии со схемой, приведенной на рисунке



вычисляем раннее время начала работ с учетом их длительностей из таблицы 1 и связей, задаваемых сетевым графиком на рисунке 1. Вычисления начинаются с первой и заканчиваются последней работой проекта. Последовательность вычислений приведена в таблице 3. Результат показывает, что длительность проекта равна 122 дня.

№	Раннее время начала	Длитель-ность		
1	-	-	-	0
2	1 0	0	0+0	0
3	2 0	15	0+15	15
4	2 0	15	0+15	15
5	2 0	15	0+15	15
6	4 15	7	15+7	22
7	4 15	7	Max(15 + 7;	30
	5 15	15	15 + 15)	
8	3 15	7	Max(15 + 7;	30
	5 15	15	15 + 15)	
9	6 22	30	Max(22+30;	52
	7 30	20	30+20;	
	8 30	5	30+5)	
10	9 52	25	52+25	77
11	9 52	25	52+25	77
12	10 77	40	Max(77+40;	117
	11 77	30	77+30)	
13	12 117	5	117+5	122

5. Вычисление позднего времени начала работ. Используя длительности работ и сетевой график, вычисляем позднее время начала работ. Вычисления начинаются с последней работы проекта, ведутся в обратном порядке и заканчиваются первой работой. Результаты вычислений приведены в таблицу 4.

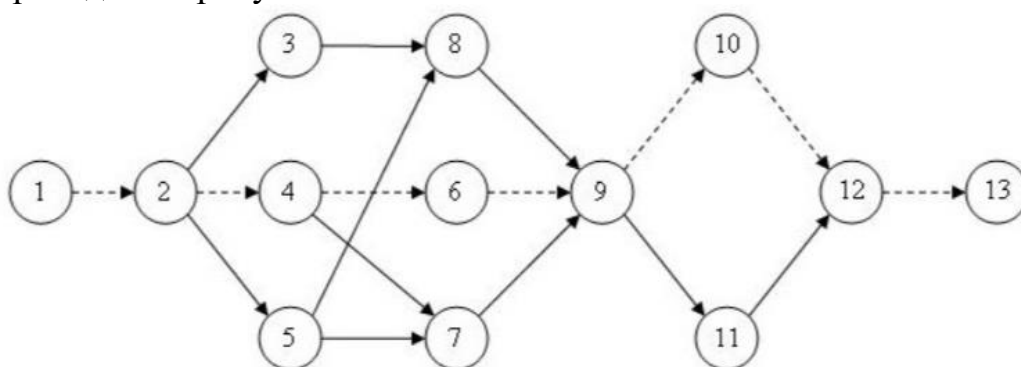
№	Позднее время начала				
13	-	-	-	-	122
12	13	122	5	122-5	117
11	12	117	30	117-30	87
10	12	117	40	117-40	77
9	10	77	25	Min(77-25;	52
	11	87		87-25)	
8	9	52	5	52-5	47
7	9	52	20	52-20	32
6	9	52	30	52-30	22
5	7	32	15	Min(32-15;	17
	8	47		47-15)	
4	6	22	7	Min(22-7;	15
	7	32		32-7)	
3	8	47	7	47-7	40
2	3	40	15	Min(40-15;	0
	4	15		15-15;	
	5	17		17-15	
1	2	0	0	0-0	0

6. Вычисление резерва времени работ. Резерв времени равен разнице между поздним и ранним временем начала работ. Занесем в таблицу 5 и значения и разность (резерв).

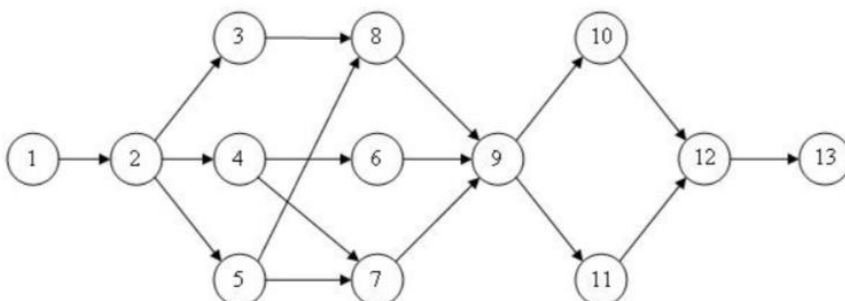
Работа №	Раннее время начала	Позднее время начала	Резерв времени
1	0	0	0
2	0	0	0
3	15	40	25
4	15	15	0
5	15	17	2
6	22	22	0
7	30	32	2

8	30	47	17
9	52	52	0
10	77	77	0
11	77	87	10
12	117	122	5
13	122	122	0

7. Нахождение критического пути проекта внедрения информационной сети. Критический путь состоит из работ с нулевым резервом времени. Обозначим на сетевом графике критический путь пунктирными стрелками. Результат приведен на рисунке.

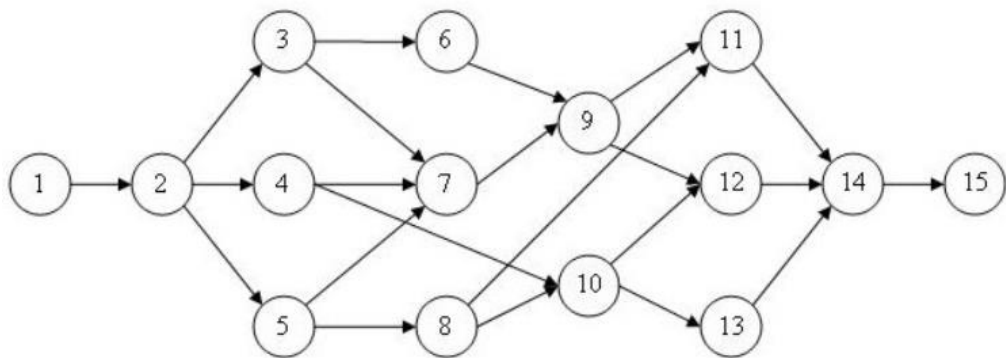


Задание 2. Вычислить критический путь по сетевому графику, изображенному на рисунке со значениями длительностей работ, приведенными в таблице.



	1	2	3	4	5	6	7	8	9	10	11	12	13
1	-	60	25	15	30	20	15	70	30	90	20	5	-
2	-	30	15	20	25	30	45	45	25	60	40	7	-
3	-	44	22	17	34	28	23	64	26	78	32	9	-
4	-	24	17	18	26	19	31	69	33	82	15	3	-
5	-	36	24	19	25	12	18	55	24	80	22	4	-
6	-	57	32	24	18	28	21	59	24	35	27	6	-
7	-	29	18	14	37	16	44	38	29	51	22	8	-
8	-	37	20	17	18	21	16	40	27	28	30	7	-
9	-	18	19	31	30	35	29	38	21	43	19	4	-
10	-	21	27	14	34	22	35	30	27	12	40	6	-

Задание 3. Последовательность работ проекта описывается сетевым графиком, изображенным на рисунке



Найти критический путь при условии, что длительности работ заданы одним из вариантов в таблице.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	-	4	7	8	6	4	6	5	3	7	5	4	3	2	-
2	-	7	10	8	9	7	5	6	5	3	4	3	5	5	-
3	-	3	6	4	7	5	4	5	4	6	3	6	7	4	-
4	-	2	2	3	2	4	4	5	4	3	3	4	5	7	-
5	-	7	6	5	4	4	5	6	7	5	5	8	6	3	-
6	-	5	7	6	7	8	9	8	9	9	7	6	6	5	-
7	-	4	4	5	6	6	5	4	5	5	7	6	8	5	-
8	-	7	5	4	3	3	4	5	4	4	2	3	2	2	-
9	-	2	3	4	5	7	3	5	6	4	6	8	7	4	-
10	-	4	7	3	2	7	5	4	7	2	4	7	10	6	-

ПРАКТИЧЕСКАЯ РАБОТА № 5

Тема: Расчет продолжительности работ и критического пути

Цель работы: научиться определять, какие работы (операции) в проекте являются критическими, и построить план проведения всех работ так, чтобы выдержать плановые сроки при минимальных затратах.

Оборудование: ПК, интернет, программное обеспечение – операционная система MS Project, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. По заданному перечню работ, построить сетевой график. Определить продолжительности полных путей графика. Определить и выделить критический путь. Определить резерв времени каждого пути. Определить коэффициенты напряженности пути. Определить ранние и поздние сроки начала и окончания работы. Определить полный резерв времени каждой работы.

Исходные данные

Пути сетевого графика:

L1: 0-1, 1-2, 2-5, 5-7, 7-8

L2: 0-1, 1-4, 4-7, 7-8

L3: 0-1, 1-4, 4-5, 5-7, 7-8

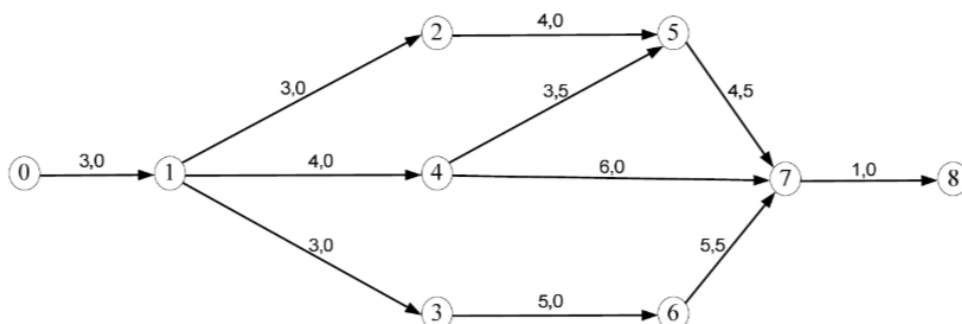
L4: 0-1, 1-3, 3-6, 6-7, 7-8

Продолжительность работ:

0-1	3,0
1-2	3,0
1-3	3,0
1-4	4,0
2-5	4,0
3-6	5,0
4-5	3,5
4-7	6,0
5-7	4,5
6-7	5,5
7-8	1,0

Решение.

1. Сетевой график



2 Продолжительности полных путей графика.

L1: 0-1, 1-2, 2-5, 5-7, 7-8; $TL1 = 3,0 + 3,0 + 4,0 + 4,5 + 1,0 = 15,5$.

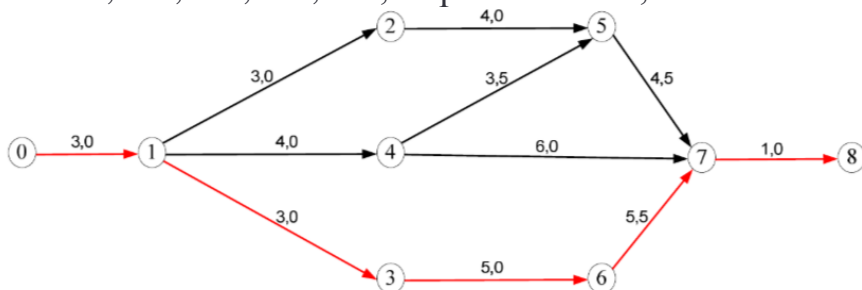
L2: 0-1, 1-4, 4-7, 7-8; $TL2 = 3,0 + 4,0 + 6,0 + 1,0 = 14,0$.

L3: 0-1, 1-4, 4-5, 5-7, 7-8 $TL3 = 3,0 + 4,0 + 3,5 + 4,5 + 1,0 = 16,0$.

L4: 0-1, 1-3, 3-6, 6-7, 7-8; $TL4 = 3,0 + 3,0 + 5,0 + 5,5 + 1,0 = 17,5$.

3 Критический путь:

L4: 0-1, 1-3, 3-6, 6-7, 7-8; $T_{кр} = TL4 = 17,5$.



4 Резервы времени каждого пути определим по формуле $RL = T_{кр} - TL$:

$RL1 = T_{кр} - TL1 = 17,5 - 15,5 = 2,0$;

$RL2 = T_{кр} - TL2 = 17,5 - 14,0 = 3,5$;

$RL3 = T_{кр} - TL3 = 17,5 - 16,0 = 1,5$;

$RL4 = T_{кр} - TL4 = 17,5 - 17,5 = 0$ – критический путь резерва времени не имеет.

5. Определим коэффициенты напряженности пути по формуле

$$K_{H_j} = \frac{T_{L_j}}{T_{кр}}$$

$$K_{H_1} = \frac{T_{L_1}}{T_{кр}} = \frac{15,5}{17,5} = 0,89;$$

$$K_{H_2} = \frac{T_{L_2}}{T_{кр}} = \frac{14,0}{17,5} = 0,80;$$

$$K_{H_3} = \frac{T_{L_3}}{T_{кр}} = \frac{16,0}{17,5} = 0,91;$$

$$K_{H_4} = \frac{T_{L_4}}{T_{кр}} = \frac{17,5}{17,5} = 1,0.$$

6 и 7 Определим ранние и поздние сроки начала и окончания работы и полный резерв времени каждой работы. Результаты расчётов сведём в таблицу. Критические работы в таблице выделены.

Шифр работ	Продолжительность, ч	Ранние сроки		Поздние сроки		Полный резерв времени
		начало	окончание	начало	окончание	
0-1	3,0	0	3,0	0	3,0	0
1-2	3,0	3,0	6,0	5,0	8,0	2,0
1-3	3,0	3,0	6,0	3,0	6,0	0
1-4	4,0	3,0	7,0	4,5	8,5	1,5
2-5	4,0	6,0	10,0	8,0	12,0	2,0
3-6	5,0	6,0	11,0	6,0	11,0	0
4-5	3,5	7,0	10,5	8,5	12,0	1,5
4-7	6,0	7,0	13,0	10,5	16,5	3,5
5-7	4,5	10,5	15,0	12,0	16,5	1,5
6-7	5,5	11,0	16,5	11,0	16,5	0
7-8	1,0	16,5	17,5	16,5	17,5	0

Работы, лежащие на критическом пути, резерва времени не имеют.

Задание 2. Компания планирует внедрение новой информационной системы для управления проектами. Для этого необходимо выполнить следующие этапы:

1. Анализ требований и проектирование системы – 2 недели.
2. Разработка программного обеспечения – 4 недели.
3. Тестирование системы – 3 недели.
4. Обучение сотрудников – 1 неделя.
5. Перенос данных в новую систему – 2 недели.
6. Настройка интеграции с существующими системами – 2 недели.
7. Проведение пилотного запуска – 1 неделя.
8. Полная реализация и запуск системы – 1 неделя.

Необходимо:

1. Рассчитать общую продолжительность проекта.
2. Построить сетевой график.
3. Определить критический путь проекта, то есть последовательность задач, определяющую максимальную продолжительность проекта.
4. Предложить возможные способы сокращения времени выполнения проекта без изменения его содержания.

ПРАКТИЧЕСКАЯ РАБОТА № 6

Тема: Вычисление детерминированных и вероятностных показателей времени

Цель работы: научиться применять методы оценки времени выполнения задач проекта, используя как детерминированные показатели, так и вероятностные оценки для учета неопределенности, освоить расчет сроков выполнения проекта и анализ рисков задержек

Оборудование: ПК, интернет, программное обеспечение – операционная система MS Project, MS Word, инструкции по выполнению работы.

Справочный материал:

Детерминированный сетевой график – такой сетевой график, в котором количество работ и их продолжительность определяется по заранее установленным нормативам (стандартам).

Сложность технических решений на производстве и трудность в координации выполнения больших комплексов работ определили актуальность таких сетей.

Характерные особенности:

- Известно заранее количество событий и работ;
- Известны продолжительности каждой из работ;

Соответственно, при такой ситуации, не вызывает труда у проектной команды вычислить количество путей, их продолжительности, имеющиеся резервы для событий и работ без дополнительных трудностей.

Однако, в большинстве случаев при планировании и управлении инновационных проектов, когда имеются обязательно какие-то элементы неясности и неопределенности, необходимо применять вероятностные сети.

Вероятностный сетевой график – такой сетевой график, который включает работы/операции с элементами неопределенности в оценках их количественных и качественных характеристик (в т.ч. продолжительности). Для расчета и построения таких графиков, которые содержат элементы неопределенности, применяют теорию вероятности.

Цель – получить ожидаемое время выполнения работ на основании ряда возможных оценок.

При расчете продолжительности работ в вероятностном графике, нам необходимо получить три временные оценки:

1. Предельное значение (Минимальное), имеющее место при наиболее благоприятных условиях выполнения работы/операции;
2. Предельное значение (Максимальное), имеющее место при наихудшем стечении обстоятельств при выполнении работы/операции;
3. Среднее значение (Наиболее вероятное), имеющее место при нормальных и обычных условиях выполнения работы/операции.

Таким образом определяется ожидаемое время выполнения работы, которое заносится в данные при построении вероятностного сетевого графика.

Расчет ожидаемого времени выполнения проектной работы определяется по следующей формуле:

$$t_{ож} = \frac{t_{min} + 4t_{нв} + t_{max}}{6},$$

где:

- $t_{ож}$ – ожидаемое время выполнения работы/операции;
- t_{min} – наиболее благоприятная временная оценка;
- t_{max} – наихудшая временная оценка;
- $t_{нв}$ – наиболее вероятная временная оценка.

Содержание работы:

Задание 1. Имеются данные по продолжительности работ:

Работа	t_{min}	t_{max}	$t_{нв}$	$t_{ож}$
1-2	2	6	5	?

Зная три временные оценки становится возможным произвести расчет продолжительности работы «1-2». Применяя формулу, получаем:

$$t_{ож} = \frac{2 + 4 \times 5 + 6}{6} = \frac{2 + 20 + 6}{6} = \frac{28}{6} = 4,6$$

Таким образом, ожидаемое время выполнения работы «1-2» будет равно 4,6 дня. Заносим эти данные в таблицу. Они нам пригодятся при расчетах продолжительности путей, ранних и поздних сроков свершения событий сети и обнаружения резервов времени.

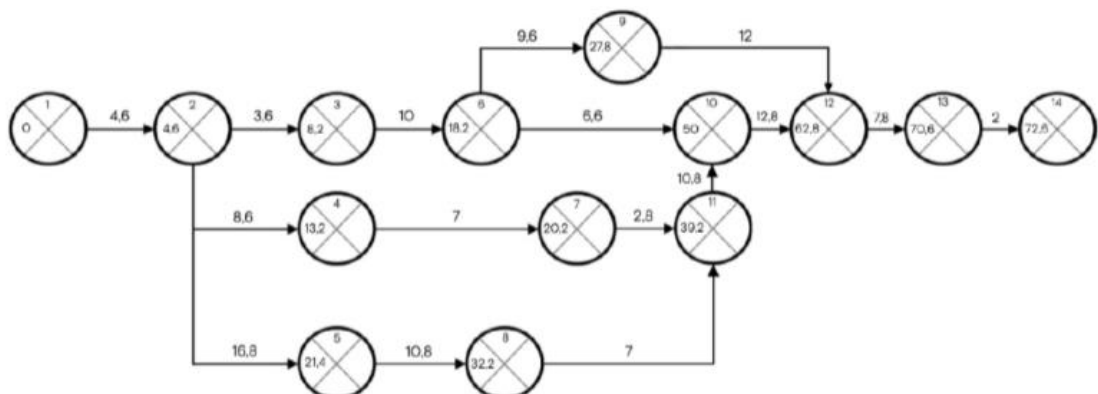
Аналогичным образом произведен расчеты для оставшихся работ проектного задания

Работа	t_{min}	t_{max}	$t_{нв}$	$t_{ож}$
1-2	2	6	5	4,6
2-3	1	5	4	3,6
2-4	5	11	9	8,6
2-5	12	21	17	16,8
3-6	8	12	10	10
4-7	5	9	7	7
5-8	7	14	11	10,8
6-9	6	12	10	9,6
6-10	4	8	7	6,6
7-11	1	4	3	2,8
8-11	5	9	7	7
9-12	10	14	12	12
10-12	11	14	13	12,8
11-10	8	13	11	10,8
12-13	5	10	8	7,8
13-14	1	3	2	2

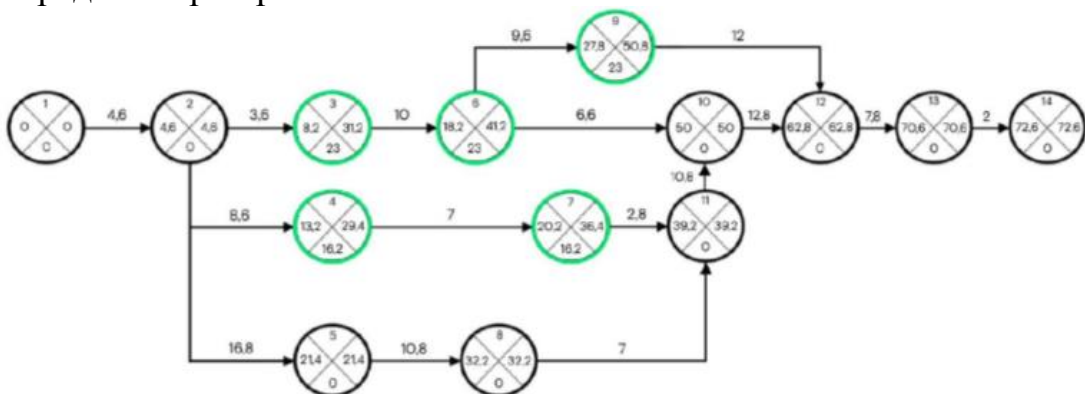
После окончания мероприятий, посвященных расчету продолжительности работ сетевого графика мы можем приступить к его созданию. Здесь необходимо технологически верно составить проектную сеть, учитывая правила построения сетевого планирования.

На рисунке представлен итоговый результат построения сети.

Шаг 1. Построить график и рассчитать ранние сроки свершения всех событий (№№1-14)



Шаг 2. Рассчитать поздние сроки свершения всех событий (№№1-14) и определить резервы.



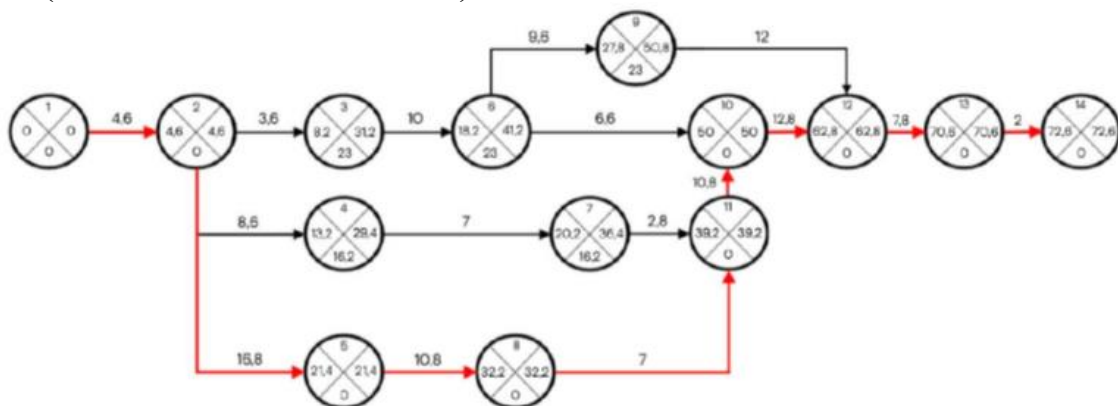
Так, были определены 5 событий, располагающих временными резервами. Это:

- событие №3 = 23 дня;
- событие №4 = 16,2 дня;
- событие №6 = 23 дня;
- событие №7 = 16,2 дня;
- событие №9 = 23 дня.

Шаг 3. Определить все возможные полные пути и их продолжительность, а также идентифицировать критический путь всего проектного задания.

Так, были определены 4 полных пути:

- $t(L1: 1;2;3;6;9;10;12;13;14) = 49,6$ дня;
- $t(L2: 1;2;3;6;10;12;13;14) = 47,4$ дня;
- $t(L3: 1;2;4;7;11;10;12;13;14) = 56,4$ дня;
- $t(L4: 1;2;5;8;11;10;12;13;14) = 72,6$ дня.



Критический путь определен. Его продолжительность составила 72,6 дня.

Задание 2. Дана информация о длине путей в сети: 18 недель, 20 недель и 14 недель. Нужно определить: длину критического пути; ожидаемую продолжительность проекта; количество резервного времени для каждого пути.

ПРАКТИЧЕСКАЯ РАБОТА № 7

Тема: Вычисление детерминированных и вероятностных показателей времени

Цель работы: научиться применять методы оценки времени выполнения задач проекта, используя как детерминированные показатели, так и вероятностные оценки для учета неопределенности, освоить расчет сроков выполнения проекта и анализ рисков задержек

Оборудование: ПК, интернет, программное обеспечение – операционная система MS Project, MS Word, инструкции по выполнению работы.

Справочный материал:

При планировании проектов важную роль играет оценка времени, необходимого для выполнения отдельных задач. В реальности существует неопределённость, связанная с различными рисками и условиями выполнения работ. Детерминированные показатели предполагают фиксированные значения времени выполнения задач, основанные на историческом опыте или экспертных оценках. Например: Задача занимает 5 дней

Вероятностные оценки учитывают неопределённость и выражаются в виде случайных величин с определённым распределением. Например: Время выполнения задачи может составлять 4, 5 или 6 дней с определённой вероятностью.

Методы оценки времени выполнения проекта

1. Метод оценки на основе экспертных заключений: использование средних значений.
2. Метод критического пути (CPM): расчет минимального времени выполнения проекта.
3. Метод оценки по вероятностным моделям: например, метод PERT (Program Evaluation and Review Technique).

Метод PERT

1. Использует три оценки времени для каждой задачи:
2. Оптимистическая (O): минимальное время.
3. Пессимистическая (P): максимальное время.
4. Наиболее вероятное (M): наиболее вероятное время.

Формула ожидаемого времени задачи:

$$TE = \frac{O + 4M + P}{6}$$

$$\text{Дисперсия оценки } \sigma^2 = \left(\frac{P-O}{6}\right)^2$$

Содержание работы:

Задание 1. Рассчитать общие сроки проекта при использовании только детерминированных данных.

Дана таблицы задач проекта с предполагаемым временем выполнения (в днях):

№	Название задачи	Время (дни)
1	Анализ требований	10

2	Проектирование системы	15
3	Разработка программного обеспечения	25
4	Тестирование	10
5	Внедрение и обучение	8

Задание 2. Расчет вероятностных показателей (метод PERT)

Для каждой задачи даны оценки:

№	Название задачи	О (оптимистическая)	М (наиболее вероятная)	Р (пессимистическая)
1	Анализ требований	8	10	12
2	Проектирование системы	12	15	20
3	Разработка ПО	20	25	35
4	Тестирование	8	10	14
5	Внедрение и обучение	6	8	10

Задача: вычислить ожидаемое время (TE) для каждой задачи; определить дисперсию каждого этапа; построить вероятностную оценку общего времени выполнения проекта, используя критический путь.

ПРАКТИЧЕСКАЯ РАБОТА № 8

Тема: Сокращение времени и затрат

Цель работы: изучить методы сокращения времени выполнения проекта и связанные с этим дополнительные расходы.

Оборудование: ПК, интернет, программное обеспечение – операционная система MS Project, MS Word, инструкции по выполнению работы.

Справочный материал:

В условиях ограниченных ресурсов и необходимости своевременного завершения проектов важно уметь снижать их временные рамки и издержки без потери качества. Для этого используют различные управленческие и технические методы.

Методы сокращения времени и затрат:

- *Параллельное выполнение работ* — выполнение отдельных этапов одновременно, что сокращает общий срок.
- *Автоматизация процессов* — внедрение технологий, снижающих трудозатраты и время выполнения задач.
- *Оптимизация процессов* — анализ текущих процедур и устранение неэффективных операций.
- *Пересмотр объема работ* — исключение избыточных задач, фокус на критичных.
- *Использование ресурсов более эффективно* — перераспределение ресурсов с целью повышения производительности.

Методы оценки эффективности

Коэффициент сокращения времени:

$$R_t = T_{\text{исх}} / T_{\text{сок}}$$

где $T_{\text{исх}}$ — исходное время, $T_{\text{сок}}$ — сокращенное время.

Коэффициент сокращения затрат:

$$R_z = Z_{\text{исх}} / Z_{\text{сок}}$$

где Z — затраты.

Анализ затрат и выгод — расчет ROI, чтобы определить целесообразность внедрения мер.

Содержание работ:

Задание 1. Анализ исходных данных. Определить общие сроки и затраты проекта.

Этап	Время (дни)	Затраты (\$)
Анализ требований	10	2000
Проектирование системы	15	3000
Разработка ПО	25	5000
Тестирование	10	2000
Внедрение	8	1500

Решение.

Общий срок – сумма всех этапов (при последовательной работе):

$$T_{\text{общ}} = 10 + 15 + 25 + 10 + 8 = 68 \text{ дней}$$

Общие затраты – сумма всех затрат:

$$Z_{\text{общ}} = 2000 + 3000 + 5000 + 2000 + 1500 = 13500 \$$$

Задание 2. Разработать меры по сокращению времени и затрат.

Допустим, внедрены следующие меры:

1. Параллельное выполнение этапов «Проектирование системы» и «Анализ требований» - сокращение времени на эти два этапа до минимального времени, равного максимальному из них (15 дней), т.к. оба могут выполняться одновременно.
2. Автоматизация разработки ПО – сокращение времени разработки на 20%, а затраты – на 10%.
3. Пересмотр требований и стандартизация тестирования – снижение затрат на тестирование на 15%.

Задача – рассчитать новые сроки и затраты.

Решение:

1. Параллельное выполнение «Анализ требований» и «Проектирование системы»:

- время каждого из этапов: Анализ требований = 10 дней, Проектирование = 15 дней.

- общий срок этих двух этапов после параллелизации: $T_H = \max(10, 15) = 15$ дней

2. Общий срок проекта после изменений:

$T_{\text{нов}} = T_{\text{этапы 1-2 (параллельно)}} + \text{Разработка} + \text{Тестирование} + \text{Внедрение}$

$T_{\text{разработка}} = 25 * (1 - 0,20) = 20$ дней

Заменяем последовательное выполнение: $T_{\text{нов}} = 15 + 20 + 10 + 8 = 53$ дней

3. Расчет новых затрат:

- затраты на разработку с учетом автоматизации:

$Z_{\text{разработка}} = 5000 * (1 - 0,10) = 4500$ \$

- затраты на тестирование с учетом пересмотра:

$Z_{\text{тестирование}} = 2000 * (1 - 0,15) = 1700$ \$

- затраты на остальные этапы остаются без изменений:

$Z_{\text{анализ}} = 2000$; $Z_{\text{проектирование}} = 3000$; $Z_{\text{внедрение}} = 1500$

Общие затраты после мер:

$Z_{\text{итог}} = 2000 + 3000 + 4500 + 1700 + 1500 = 12700$ \$

Задание 3. Произвести оценку эффективности мер.

1. Коэффициент сокращения времени: $R_t = T_{\text{исх}} / T_{\text{нов}} = 68 / 53 = 1,28$

Сокращение времени примерно на 28%

2. Коэффициент сокращения затрат: $R_z = 13500 / 12700 = 1,06$

Снижение затрат примерно на 6%

3. ROI (возврат инвестиций): затраты на внедрение мер 3000 \$.

Экономия затрат: $13500 - 12700 = 800$ \$

Время экономии – 15 дней (с 68 до 53)

В целом, инвестиции оправданы, т.к. сокращение времени и затрат способствует более быстрой реализации и экономии.

ПРАКТИЧЕСКАЯ РАБОТА № 9

Тема: Сокращение времени и затрат

Цель работы: изучить методы сокращения времени выполнения проекта и связанные с этим дополнительные расходы.

Оборудование: ПК, интернет, программное обеспечение – операционная система MS Project, MS Word, инструкции по выполнению работы.

Справочный материал:

Определение критического пути. Это путь, который определяет время завершения проекта.

Методы сокращения времени. К ним могут относиться снижение качества, заключение контракта с субподрядчиком, выделение дополнительных ресурсов и оборудования.

Построение графика стоимости времени выполнения проекта. Для этого нужно найти общие прямые и косвенные издержки для разных продолжительностей проекта, а затем суммировать эти издержки.

Определение операций для сокращения времени их выполнения. Обычно сокращают продолжительность критических операций, время выполнения которых можно уменьшить с наименьшим повышением стоимости на единицу времени.

Содержание работ:

Задание 1. Оптимизация производственного процесса.

Компания производит 1000 единиц продукции в месяц. Среднее время изготовления одной единицы – 2 часа, а затраты на один цикл – 5000р.

Задача – сократить время производства и затраты

Задание 2. Сокращение затрат в ИТ-проекте за счет аутсорсинга.

Компания разрабатывает программное обеспечение внутренними ресурсами. Общие затраты – 2000000р, сроки – 12 месяцев.

Задача – снизить затраты и сократить сроки.

Задание 3. Стандартизация и автоматизация документооборота

Бухгалтерия компании тратит около 40 часов в месяц на обработку договоров и отчетов, а затраты – 50000 р.

Задача – сократить время и затраты, повысить эффективность.

ПРАКТИЧЕСКАЯ РАБОТА № 10

Тема: Работа с Календарем проекта в программе MS Project

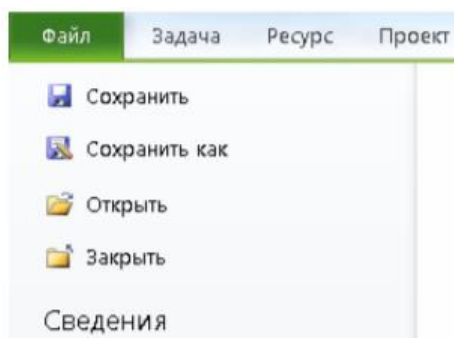
Цель работы: научить студентов создавать новые проекты среде MS Project; овладение навыками настройки календаря проекта.

Оборудование: ПК, интернет, программное обеспечение – операционная система MS Project, MS Word, инструкции по выполнению работы.

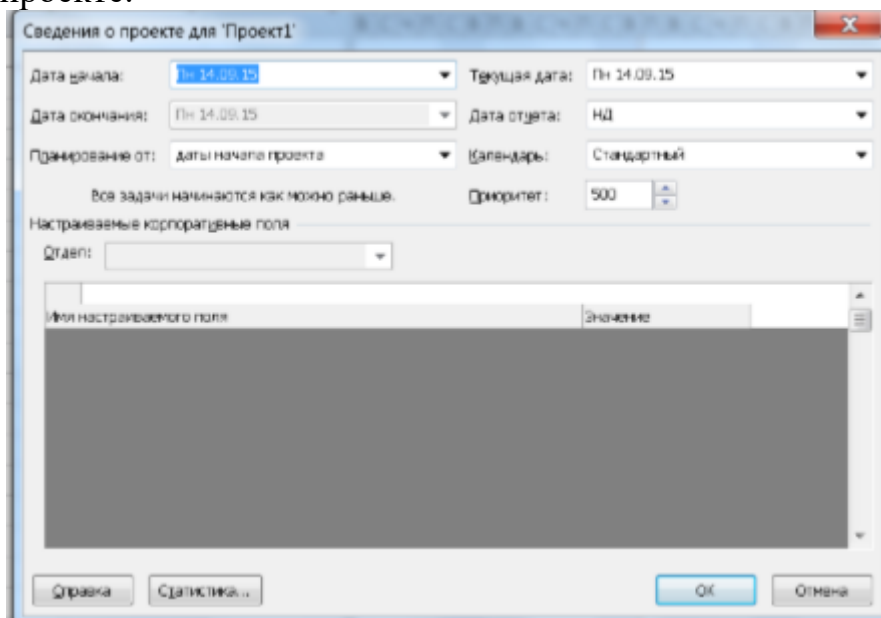
Содержание работы:

Задание 1. Настроить календарь проекта на текущий год

1. Запустить Microsoft Project. После запуска Microsoft Project открывается пустой проект, открытый на форме представления Диаграмма Гантта. Этот проект следует сохранить под нужным именем в нужную папку при помощи пункта Сохранить из меню Файл.



2. Следующим шагом необходимо установить базовые характеристики проекта, такие как Дата начала, (Дата окончания), Календарь и др. Для установки значений базовых характеристик проекта используется меню Проект, пункт Сведения о проекте.



3. Чтобы обозначить в проекте обычные рабочие дни и часы, а также стандартное нерабочее время используется Календарь проекта:

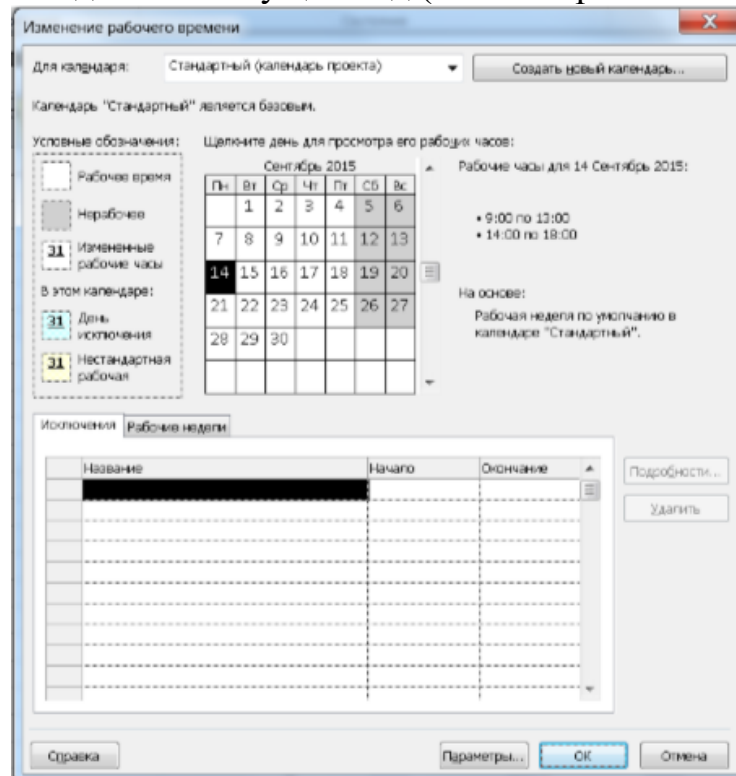
Календарь проекта - базовый календарь, используемый в проекте. Календарь, который используется для назначения календарного плана работ по умолчанию всем задачам проекта.

Нерабочее время часы или дни в календаре трудового ресурса или проекта в целом, на которые Project не может планировать выполнение задач, поскольку работа в это время не выполняется. В нерабочее время входят перерывы на обед, выходные и праздничные дни.

Рабочее время - часы, в которые может выполняться работа согласно календарю проекта или трудового ресурса.

Для настройки календаря проекта используется меню Проект, пункт Изменение рабочего времени.

Настройте календарь так, чтобы выходные дни соответствовали установленным дням отдыха на текущий год (включая рабочие воскресенья)



4. Сохранить результат.

Задание 2. Настройка календаря проекта по вариантам. Оформить отчет, включающий вид созданного календаря на текущий год (изображение вкладки Изменение рабочего времени - Исключения, Изменение рабочего времени - Рабочие недели и другие) и скриншоты выполнения задания

Примечание: 20XX – текущий год

Если начало проекта, дополнительный выходной в текущем году выпадает на выходной, то следует сместить дату назад (позже) на 1-2 дня

Вариант	Начало проекта	Часы работы	Дополнительный выходной
1.	27. 10.20XX	8:30-12:30- 13:30-17:30	08. 11.20XX
2.	28. 10.20XX	9:00-13:00- 14:00-18:00	12. 11.20XX
3.	29. 10.20XX	9:30-13:30- 14:30-18:30	13. 11.20XX
4.	30. 10.20XX	10:00-14:00- 15:00- 19:00	14. 11.20XX
5.	01.11.20XX	10:00-14:00- 15:00- 19:00	18. 11.20XX
6.	02. 11.20XX	8:00-12:00- 13:00-17:00	21. 11.20XX
7.	03. 11.20XX	8:30-12:30- 13:30-17:30	22. 11.20XX
8.	04. 11.20XX	9:00-13:00- 14:00-18:00	23. 11.20XX
9.	07. 11.20XX	9:30-13:30- 14:30-18:30	24. 11.20XX
10.	08. 11.20XX	10:00-14:00- 15:00- 19:00	25. 11.20XX

ПРАКТИЧЕСКАЯ РАБОТА № 11

Тема: Работа с Календарем проекта в программе MS Project

Цель работы: научить студентов создавать новые проекты среде MS Project; овладение навыками настройки календаря проекта.

Оборудование: ПК, интернет, программное обеспечение – операционная система MS Project, MS Word, инструкции по выполнению работы.

Содержание работы:

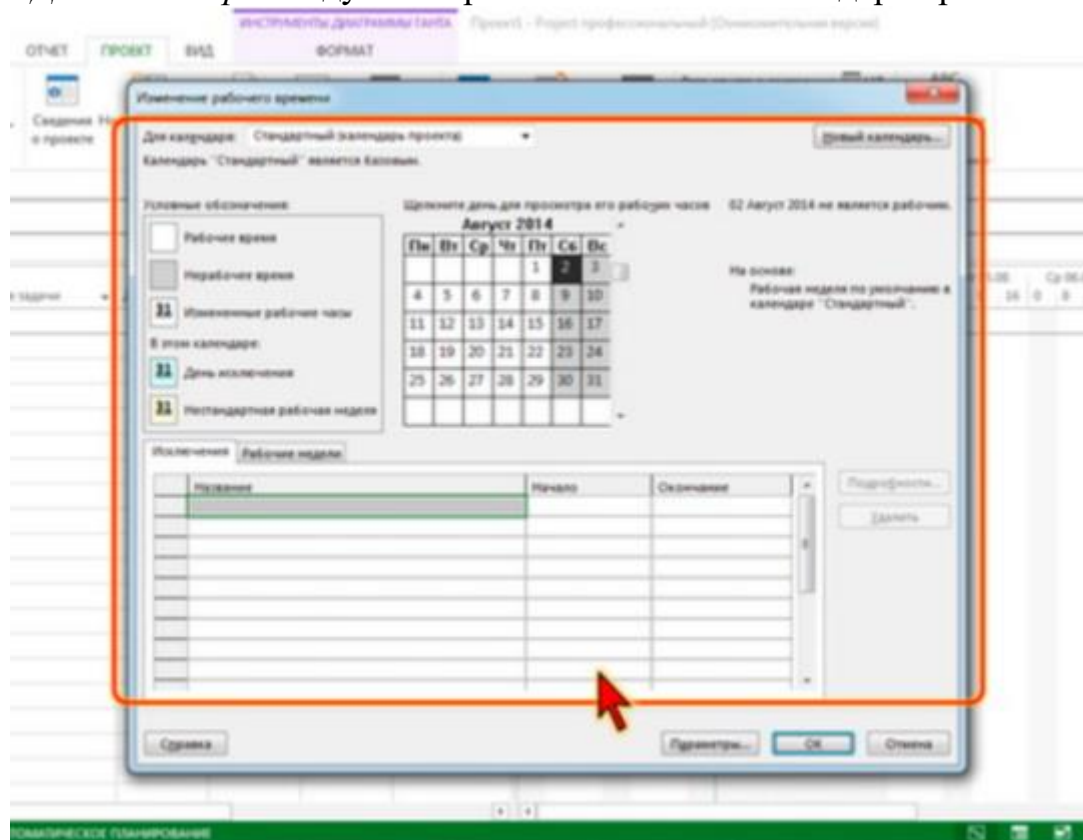
Задание 1. Изменение календаря.

1. Выбор изменяемого календаря.

Календарь Project не может учесть особенности национального календаря (праздники и предпраздничные дни) и, уж тем более, особенности рабочего графика конкретной организации (время начала и окончания рабочего дня, коллективный отпуск и т.д.). Поэтому практически всегда календарь проекта требует корректировки.

Для изменения календаря проекта необходимо на вкладке *Проект* в группе *Свойства* нажать кнопку *Изменить рабочее время*.

В диалоговом окне *Изменение рабочего времени* в раскрывающемся списке *Для календаря* следует выбрать изменяемый календарь проекта

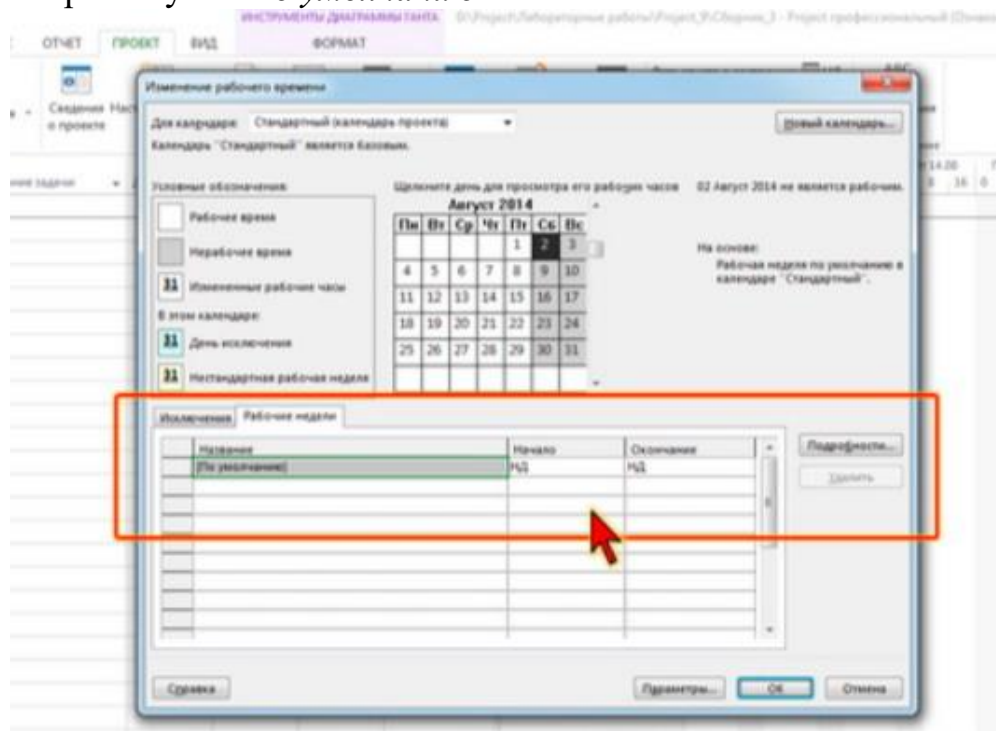


2. Изменение параметров рабочей недели

Можно изменить время начала и окончания рабочего дня, время и продолжительность обеденного перерыва. Эти параметры можно настраивать как для всех дней недели, так и для отдельных.

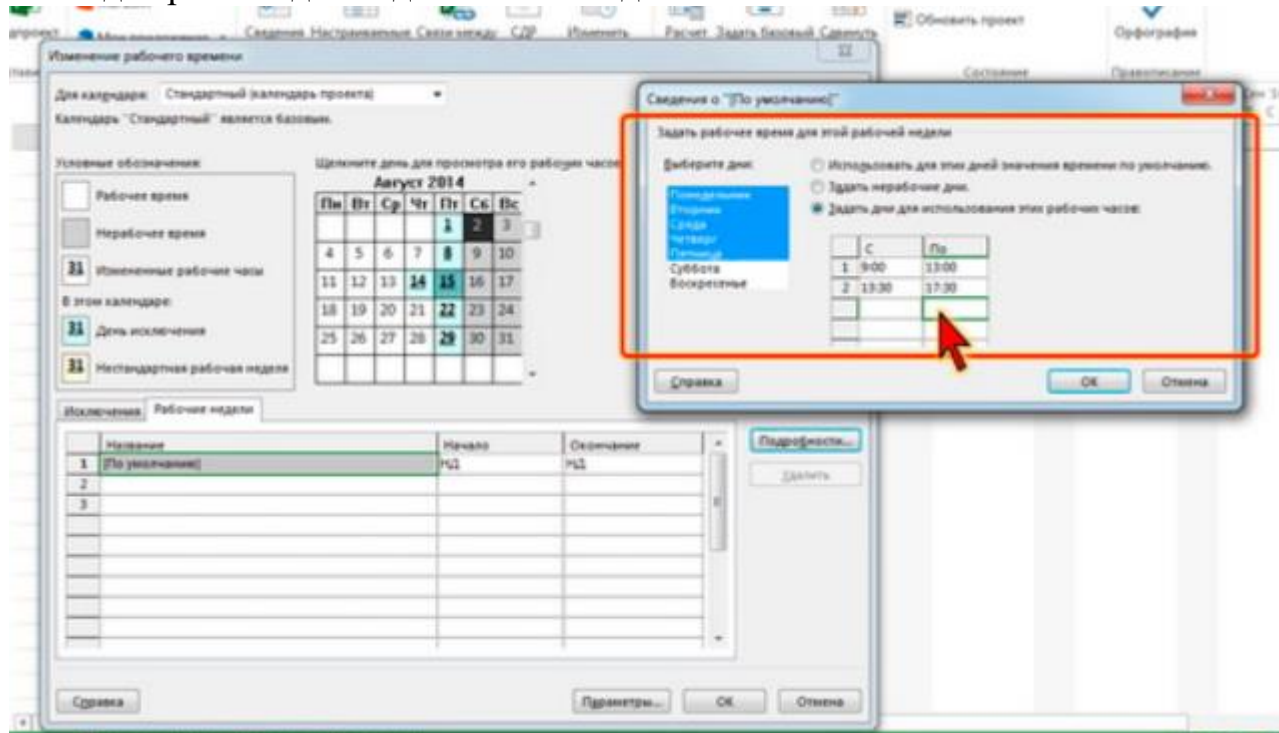
Например, требуется установить рабочее время с 9:00 до 17:30 с получасовым обеденным перерывом с 13:00 до 13:30.

1. В окне *Изменение рабочего времени* перейдите на вкладку *Рабочие недели* и выберите пункт *По умолчанию*



2. Нажмите кнопку *Подробности*, после чего отобразится диалоговое окно *Сведения о [По умолчанию]*.

3. В списке *Выберите дни* выделите день недели (или несколько, перетаскив указатель при нажатой левой кнопке мыши), установите переключатель *Задать дни для использования этих рабочих часов* и в таблице введите два интервала времени для работы до обеда и после обеда.



4. Нажмите *Ок*

5. Самостоятельно выполните следующее:

- изменить время работы в любой отдельный день или нерабочий день сделать рабочим.
- указать время окончания рабочего дня в пятницу не в 17:30, а в 16:30. Или сделать рабочим днем субботу или воскресенье.
- сделать какой-нибудь рабочий день недели сделать нерабочим, его следует выделить в списке *Выберите дни* и установить переключатель *Задать нерабочие дни*.
- вернуться к рабочему графику, принятому для данного календаря по умолчанию, достаточно в списке *Выберите дни* выделить день (или несколько дней) и установить переключатель *Использовать для этих дней значения рабочего времени по умолчанию*.

Примечание: Поскольку для пункта *[По умолчанию]* не выбраны значения начала и окончания периода, установленные параметры рабочей недели будут действовать все время выполнения проекта.

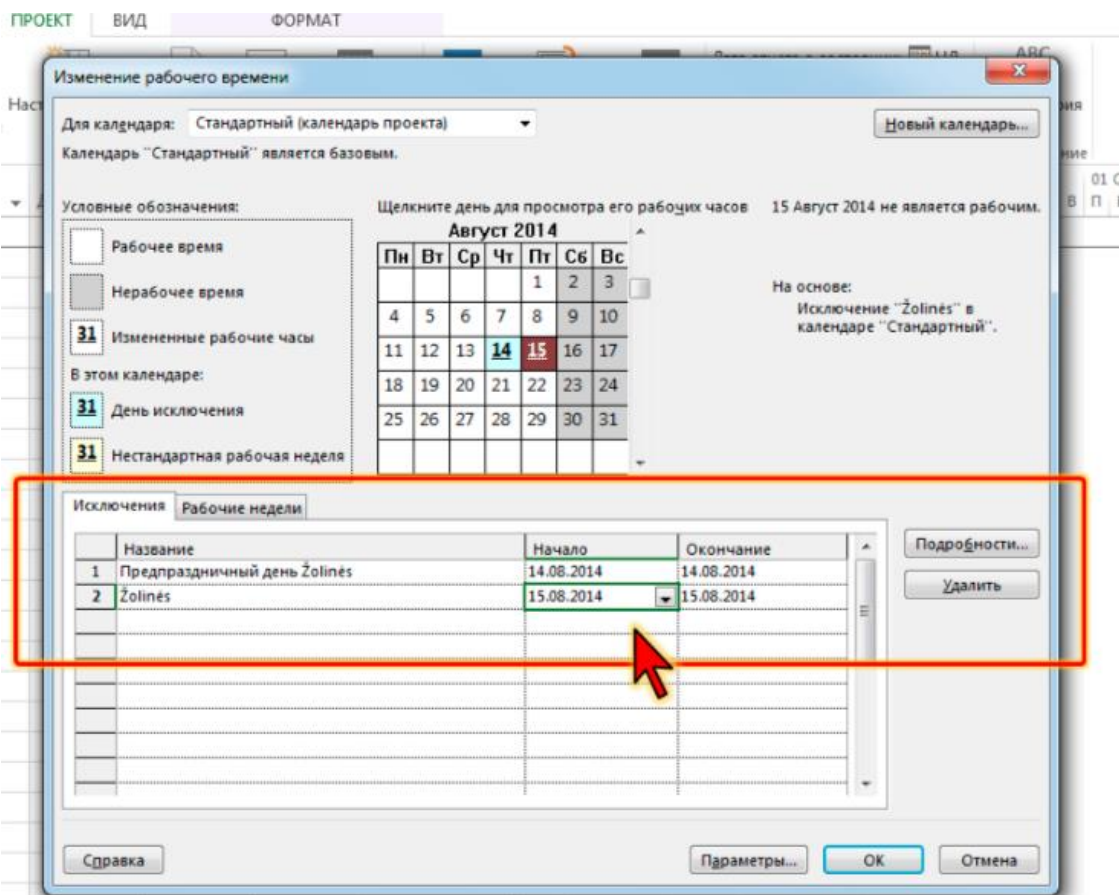
- задать параметры рабочей недели, действующие ограниченный период времени. Для этого надо ниже пункта *[По умолчанию]* ввести новое название, в столбцах *Начало* и *Окончание* указать продолжительность данного рабочего графика, а затем нажать кнопку *Подробности*, и аналогично предыдущему случаю задать точный интервал рабочего времени на каждый день.

Задание 2. Создать исключения

Исключения – это дни, рабочее время которых по каким-либо причинам отличается от рабочего времени установленного параметрами рабочей недели. Чаще всего это относится к праздникам и предпраздничным дням, но это могут быть, например, и дни, особенности работы в которые установлены в организации.

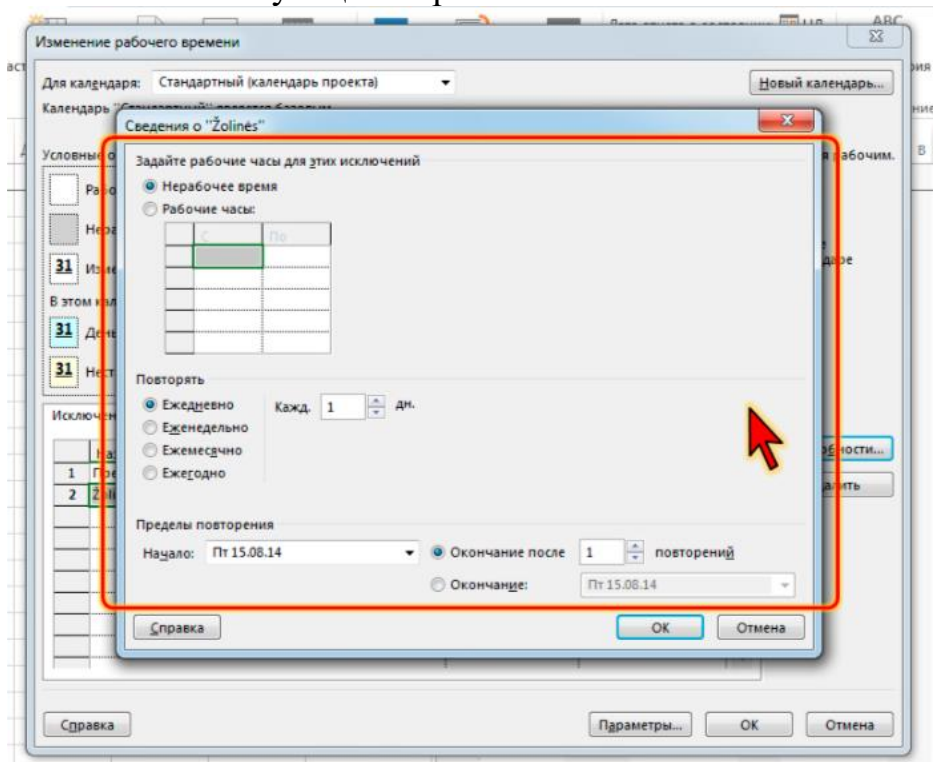
1. Для создания исключений в календаре используется вкладка *Исключения* диалогового окна *Изменение рабочего времени*. Например, в рамках создаваемого проекта 15 августа является праздничным днем, а, соответственно, 14 августа – предпраздничным (укороченным). Кроме того, в этом условном издательстве принято, что последняя суббота месяца является рабочей.

1.1. В столбце *Название* введите наименования периода, для которого изменяется календарь (в общем случае, период может быть назван любым именем), а затем в столбцах *Начало* и *Окончание* введите даты действия изменяемого периода. Если речь идет об одном дне, например, праздник, то достаточно ввести дату начала, а дата окончания автоматически будет установлена в тот же день. Даты в ячейках столбцов *Начало* и *Окончание* можно не вводить с клавиатуры, а выбрать из календаря, который открывается при щелчке по стрелке раскрывающего списка выделенной ячейки.



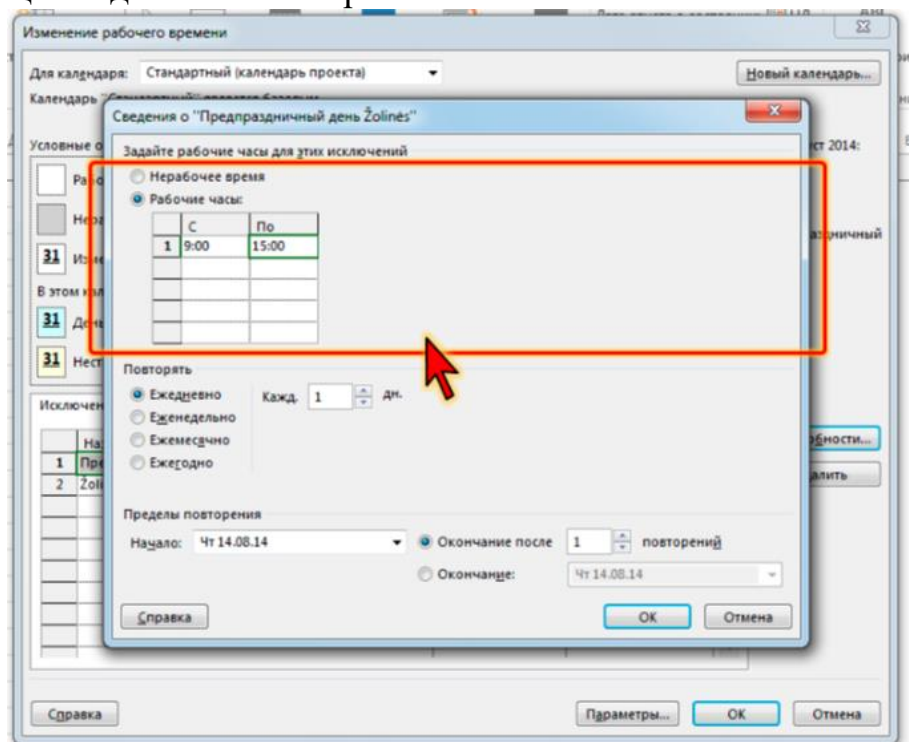
1.2. Для настройки параметров изменяемого периода нажмите кнопку *Подробнее*, после чего появится диалоговое окно *Сведения о <Название изменяемого периода>*.

1.3. Если настраиваемый период является нерабочим временем, в верхней части окна установите соответствующий переключатель.



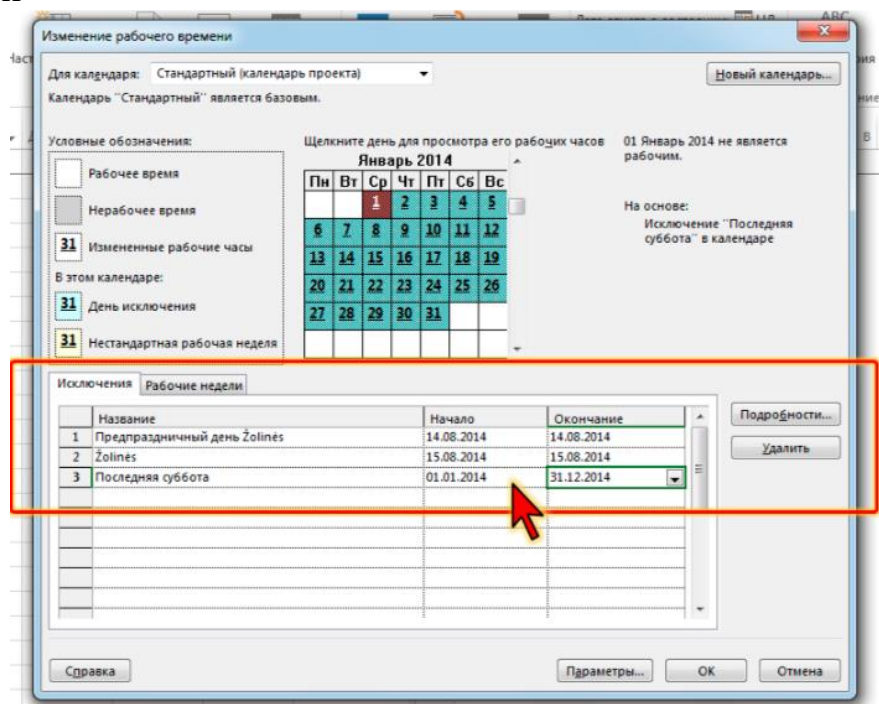
1.4. Если настраиваемый период является рабочим временем, в верхней части окна установите переключатель *Рабочие часы*.

1.5. В таблице введите значение рабочих часов.

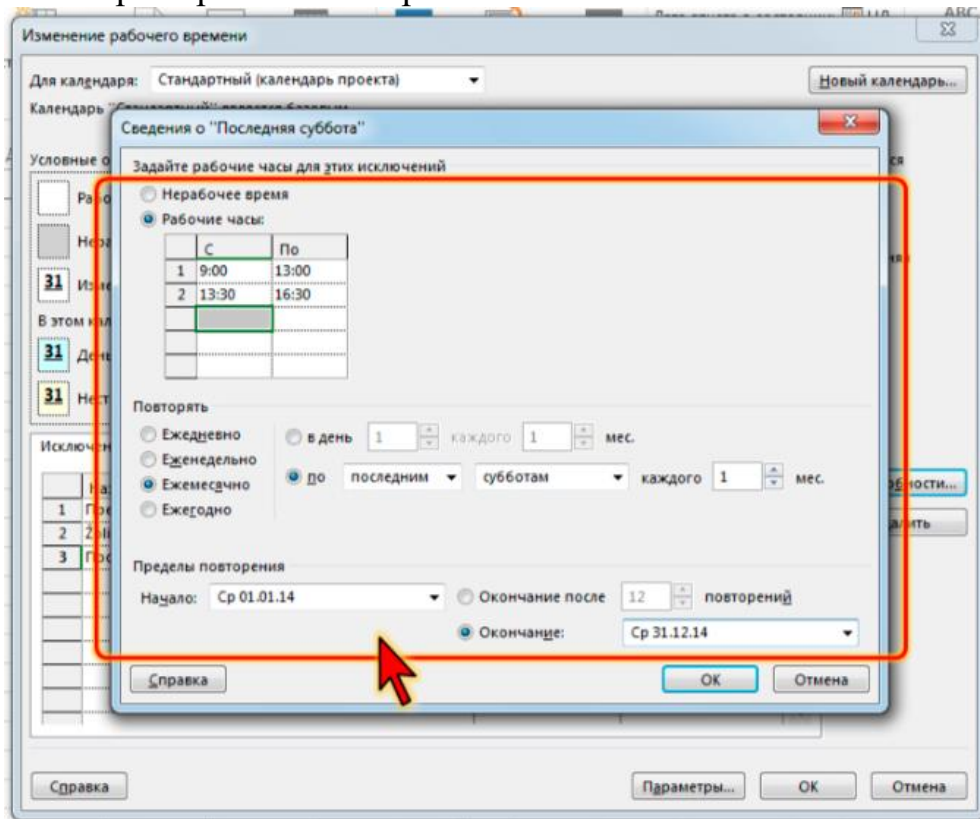


Как рабочие, так и нерабочие периоды могут повторяться. В этом случае в ячейках столбцов *Начала* и *Окончание* надо обязательно указать даты периода действия исключения. В рассматриваемом примере (последняя суббота каждого месяца является рабочим днем) в качестве даты начала можно использовать 1 января текущего года, а в качестве даты окончания – 31 декабря текущего года.

1.6. После ввода названия исключения и установки дат нажмите кнопку *Подробности*



1.7. В диалоговом окне Сведения о <Название исключения> в разделе Повторить выберите режим повторения



1.8. Повторение можно установить ежедневно, еженедельно, ежемесячно или ежегодно. В данном случае создается ежемесячное повторение. Для ежемесячного повторения можно выбрать число или день недели, когда создаваемое исключения будет повторяться. В рассматриваемом случае установите переключатель *по*, а рядом расположенных раскрывающихся списках выберите *последним* и *субботам*. В разделе *Пределы повторения* по умолчанию установлена дата начала периода времени. Выберите число повторений или дату окончания. Нажмите кнопку *OK*.

Задания для самостоятельного выполнения:

Задание 1.

1. Запустите MS Project.
2. Создайте новый проект.
3. Сохраните проект как файл под именем *Zadanie1.mpp*.
4. Выберите способ планирования проекта от даты начала.
5. Установите для проекта дату начала 22 сентября 2025 года.
6. Скопируйте в проект первые 5 задач из файла проекта *Zadanie1.mpp*. Убедитесь, что в качестве даты начала скопированных задач установлена дата начала проекта.
7. Сохраните файл *Lesson1*.
8. Закройте файлы *Zadanie1* и *Lesson1*. Возможные изменения в файлах не сохраняйте.

Задание 2.

1. Создайте новый проект.
2. Выберите способ планирования проекта от даты окончания.
3. Установите для проекта дату окончания 30 октября 2025 года.
4. Скопируйте в проект первые 5 задач из файла проекта Zadanie2.mpp. Убедитесь, что в качестве даты окончания скопированных задач установлена дата окончания проекта.
5. Сохраните проект как файл под именем Lesson2.

Задание 3.

1. Откройте файл Zadanie3.mpp.
2. Измените дату начала проекта на 27 марта 2015 года. Убедитесь, что автоматически перенесены даты начала всех задач проекта.
3. Сохраните проект под именем Lesson3. Закройте файлы Zadanie3.

Задание 4.

1. Откройте файл Zadanie4.mpp.
2. Измените календарь проекта *Стандартный*:
 - установите рабочее время с 10:00 до 18:00 с получасовым перерывом на обед с 13:30 до 14:00;
 - установите каждую первую субботу месяца рабочим днем;
 - установить сокращенный рабочий день (до 17:00) каждую пятницу, кроме пятницы перед рабочей субботой;
 - установите как нерабочие дни 10 марта (перенос дня отдыха с 8 марта), 1-2 мая, 11 мая (перенос дня отдыха с 9 мая).
3. Измените календарь проекта *24 часа*:
 - установите как нерабочие дни 10 марта (перенос дня отдыха с 8 марта), 1-2 мая, 11 мая (перенос дня отдыха с 9 мая);
 - удалите нерабочий день 24 июня.
4. Сохраните проект под именем Lesson4.
5. Закройте Project.

ПРАКТИЧЕСКАЯ РАБОТА № 12

Тема: Определение состава задач проекта, структурирование списка задач, СДР-коды, ввод задач в программе MS Project

Цель работы: научить студентов определять состав задач проекта в среде MS Project; овладение навыками структурирования списка задач проекта, использование СДР-кодов и ввода задач в проект.

Оборудование: ПК, интернет, программное обеспечение – операционная система MS Project, MS Word, инструкции по выполнению работы.

Справочный материал:

Часто, после того, как определен состав задач и ресурсов проекта, сделаны назначения, необходимо внести в проект дополнительную информацию, например, комментарии. Иногда требуется прокомментировать отдельное назначение или ресурс. Если задачи структурированы с помощью суммарных задач, то ресурсы в представлении лист ресурсов никак не структурированы. Для структурирования их, или задач другим способом, можно использовать коды структур. Например, присвоить ресурсам код отдела, к которому они относятся, и сгруппировать по этому коду. Можно также определить объем трудозатрат определенного отдела, спланировать его загрузку и т.д.

Дополнительную информацию можно вносить как в стандартные, так и в настраиваемые поля. Стандартные поля: поле кода структуры задач; поле приоритета; имена групп ресурсов, заметок и гиперссылок для задач, ресурсов и назначений. Нестандартные поля могут относиться и к задачам, и к ресурсам и бывают двух типов: собственно, настраиваемые поля, и настраиваемые поля кодов структур.

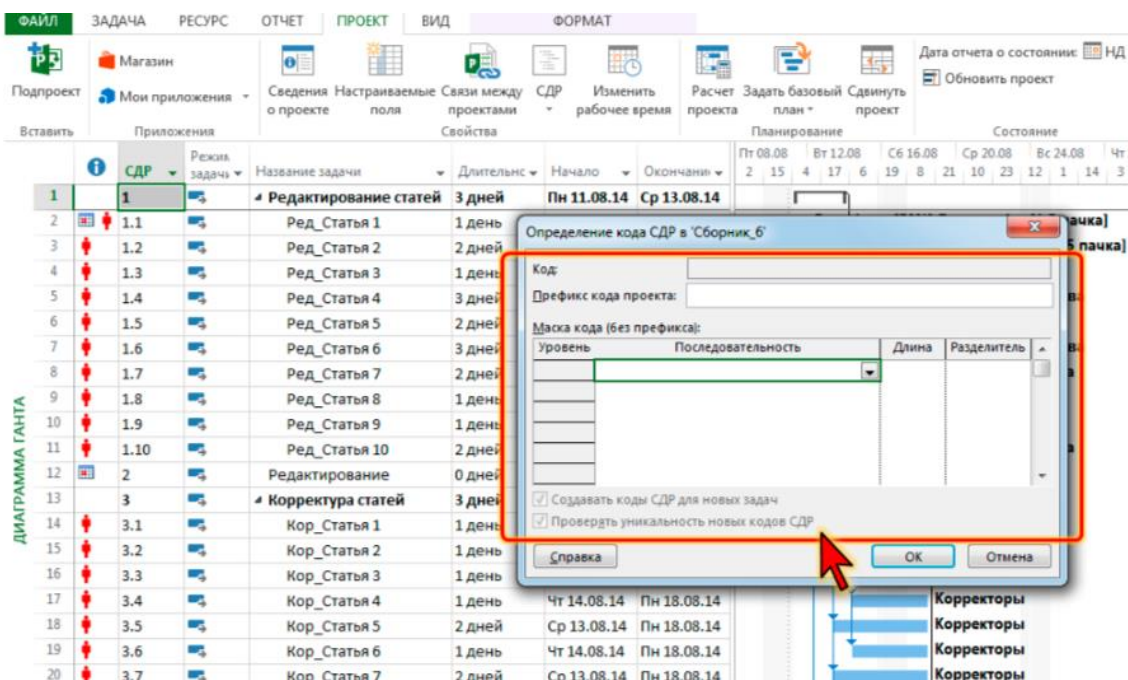
Project позволяет представить структурную декомпозицию работ с помощью идентификаторов задач или с помощью кодов СДР (структурная декомпозиция работ), назначаемых каждой задаче. Коды используются для идентификации задач, учета их исполнения, анализа трудозатрат и прочего. Например, собрав данные из однотипных проектов об одном и том же отделе можно сравнить производительность разных команд.

Project автоматически назначает базовые номера в структуре для каждой задачи, но в любое время к проекту можно применить собственные настроенные схемы структуры.

Содержание работы:

Задание 1. Работа с кодом структуры задач

Для работы с кодами в таблицу следует добавить столбец *СДР*. С кодами можно работать непосредственно в ячейках столбца, но удобнее для этой цели отобразить специальное диалоговое окно. Для это следует на вкладке *Проект* в группе *Свойства* нажать кнопку *СДР* и в появившемся меню выбрать команду *Определить код*.

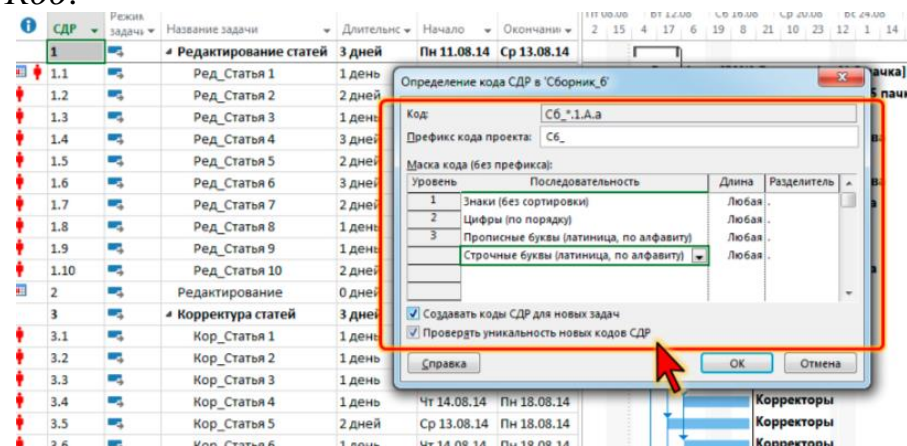


Структура кода задается в столбце *Последовательность*. В столбце *Длина* выбирается количество знаков для каждого уровня строки кода. Обычно выбирают параметр *Любая*, чтобы использовать на этом уровне кода любое число знаков. В столбце *Разделитель* выбирают знак, который будет разделять строки разных уровней кода.

Чтобы отметить все уровни структуры задачи, ее уникальный код строится из нескольких последовательностей символов, отделенных друг от друга разделителями. В каждом из уровней можно использовать свой тип символов.

Для того чтобы можно было использовать собственные коды в виде слов, в столбце *Последовательность* следует выбрать параметр *Знаки (без сортировки)*. Собственно, код будет задан в последующем.

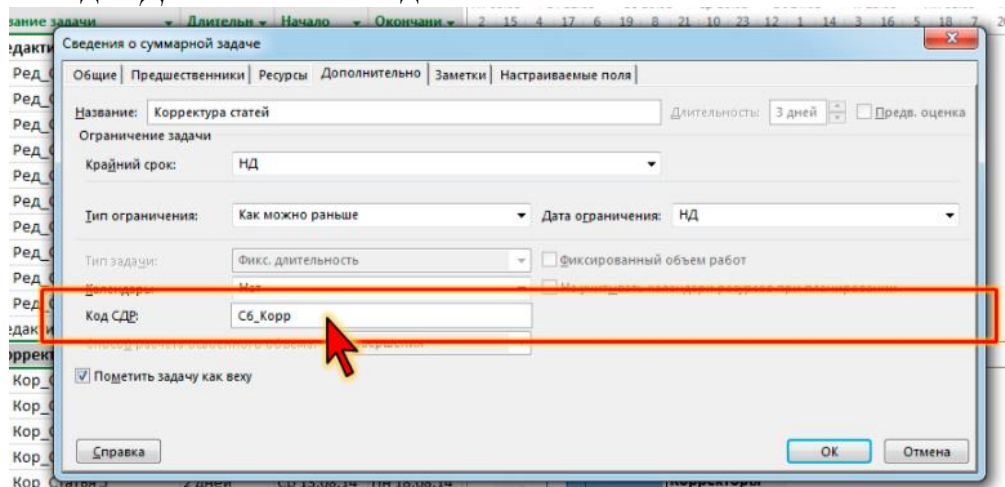
На показано формирование кода СДР таким образом, чтобы для задач первого уровня можно было задавать код произвольными символами, задачи второго уровня нумеровались арабскими цифрами, задачи третьего уровня – прописными латинскими буквами, а задачи четвертого уровня – строчными латинскими буквами. Образец кода отображается в верхней части окна в поле *Код*.



Следует обратить внимание, что после применения такого кода в ячейках столбца СДР для задач первого уровня вместо фрагмента кода, для которого выбран параметр *Знаки (без сортировки)*, отображаются звездочки. Это означает, что вместо звездочки можно ввести собственный фрагмент.

Для удобства последующей работы с кодами установить флажок *Создавать коды СДР для новых задач*, чтобы каждой новой задаче автоматически присваивался код, и флажка *Проверять уникальность новых кодов СДР*, который обеспечит проверку уникальности кодов, введенных вручную.

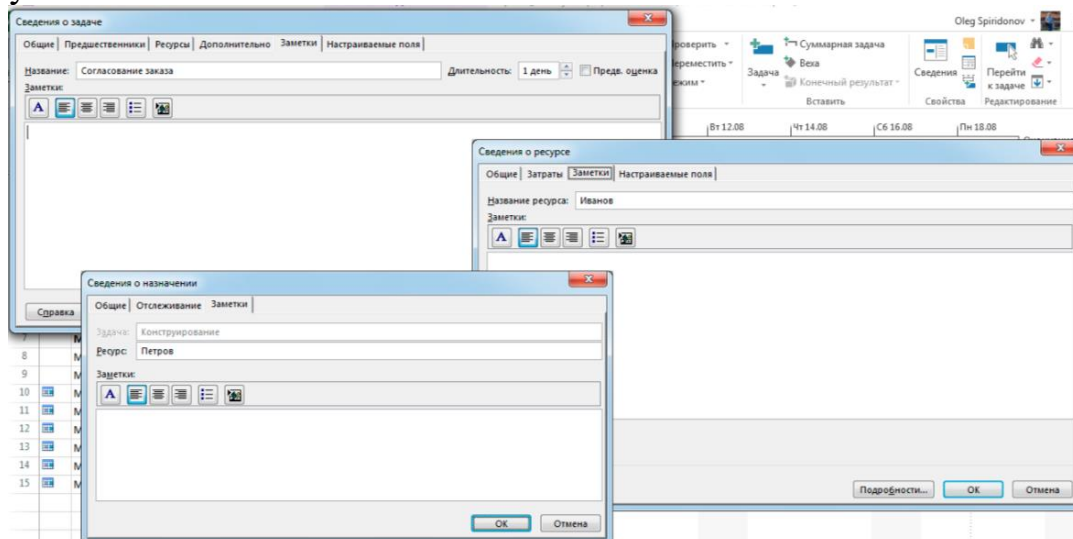
Изменение кода можно производить непосредственно в ячейках столбца СДР, но несколько удобнее это сделать на вкладке *Дополнительно* диалогового окна *Сведения о задаче*.



Следует обратить внимание, что при изменении кода более высокого уровня, все вложенные коды изменяются автоматически.

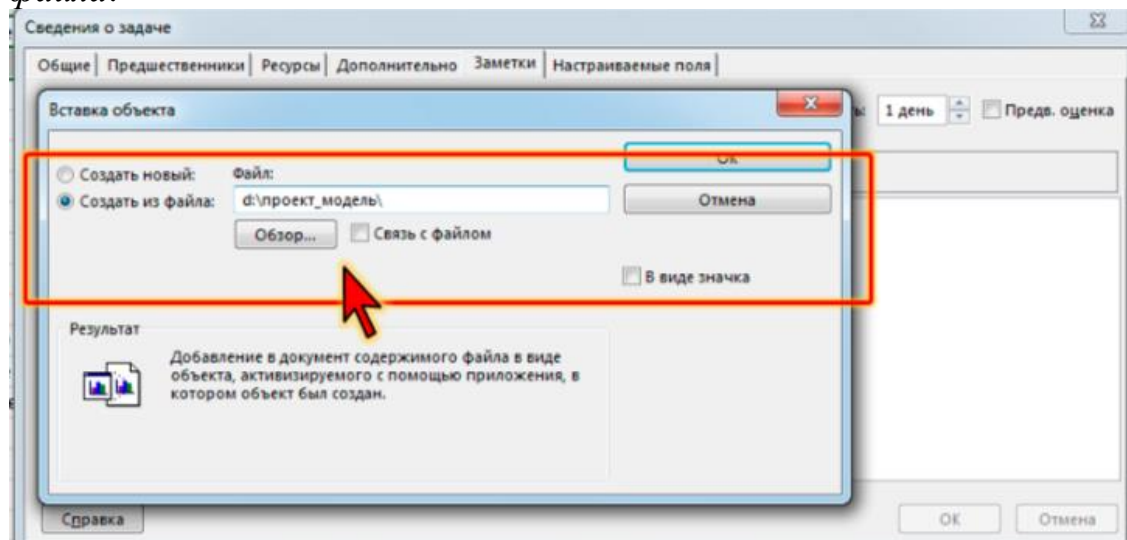
Задание 2. Добавление заметок и документов

Чтобы сопроводить какую-либо задачу, ресурс или назначение заметкой или документом нужно использовать вкладку *Заметки* диалогового окна сведений о задаче или ресурсе. Для быстрого отображения нужной вкладки диалогового окна можно на вкладке *Задача (Ресурс)* в группе *Свойства* нажать кнопку *Заметки*.



Заметка может содержать текст, в том числе и форматированный средствами вкладки (размер шрифта, выравнивание, маркированный список). В заметку можно вложить файл.

1. На вкладке *Заметки* нажмите кнопку *Вставить объект*. В появившемся диалоговом окне *Вставка объекта* установите переключатель *Создать из файла*.



2. Для перехода в папку, содержащую добавляемый файл, нажмите кнопку *Обзор*, после чего появится окно *Обзор*. Внешний вид окна определяется типом установленной на компьютере операционной системы. Перейдите в нужную папку и дважды щелкните мышью по значку добавляемого файла.

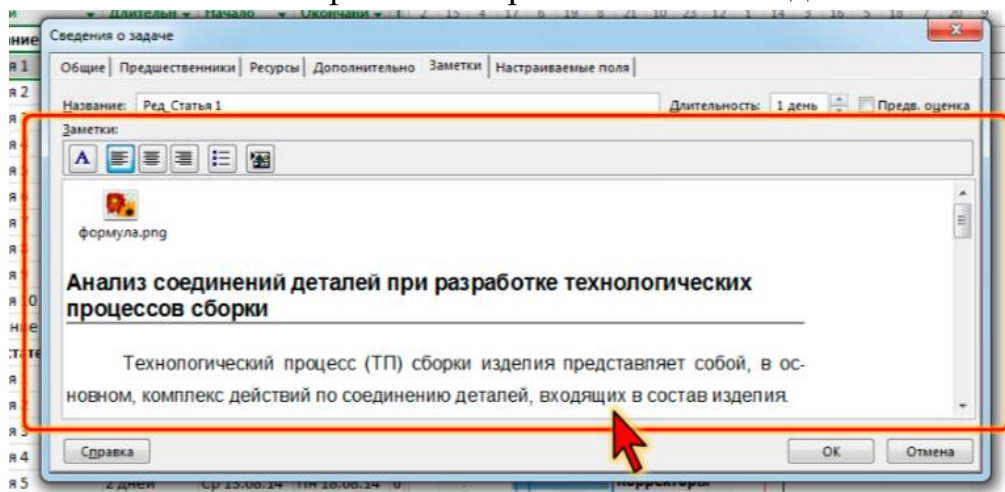
3. В окне *Обзор* будет сформирован путь к добавляемому файлу.

4. Если требуется, чтобы вложенный файл автоматически обновлялся при изменении исходного файла, установите флажок *Связь с файлом*.

5. Если требуется, чтобы в заметке отображался только значок файла, установите флажок *В виде значка*. Следует обратить внимание, что файлы некоторых типов в любом случае будут вставлены только в виде значка.

6. Нажмите кнопку *ОК*.

Вставленные файлы отображаются на вкладке *Заметки*



В таблицах задачи, ресурсы и назначения, к которым добавлены заметки, выделены специальным значком в столбце *Индикаторы*.

Следует иметь в виду, что добавление файлов в заметки увеличивает размер файла проекта.

Для удаления заметки нужно выделить любую ячейку задачи (ресурса, назначения), на вкладке *Задача* в группе *Редактирование* нажать кнопку *Очистить* и в появившемся меню выбрать команду *Заметки*.

Задание 3.

1. Запустите Project.
2. Откройте файл *Zadanie12.3. mpp*.
3. Создайте маску кода СДР, позволяющую для задач по редактированию создать код *P-1*, *P-2* и т.д., а для задач по корректуре – коды *K-1*, *K-2* и т.д.
4. Отобразите столбец *СДР*.
5. Для подзадач по редактированию создайте коды *P-1*, *P-2* и т.д., а для подзадач по корректуре – коды *K-1*, *K-2* и т.д.
6. Удалите коды СДР всех остальных задач.
7. Сохраните проект как файл под именем *Lesson_12_03*.

ПРАКТИЧЕСКАЯ РАБОТА № 13

Тема: Определение состава задач проекта, структурирование списка задач, СДР-коды, ввод задач в программе MS Project

Цель работы: научить студентов определять состав задач проекта в среде MS Project; овладение навыками структурирования списка задач проекта, использование СДР-кодов и ввода задач в проект.

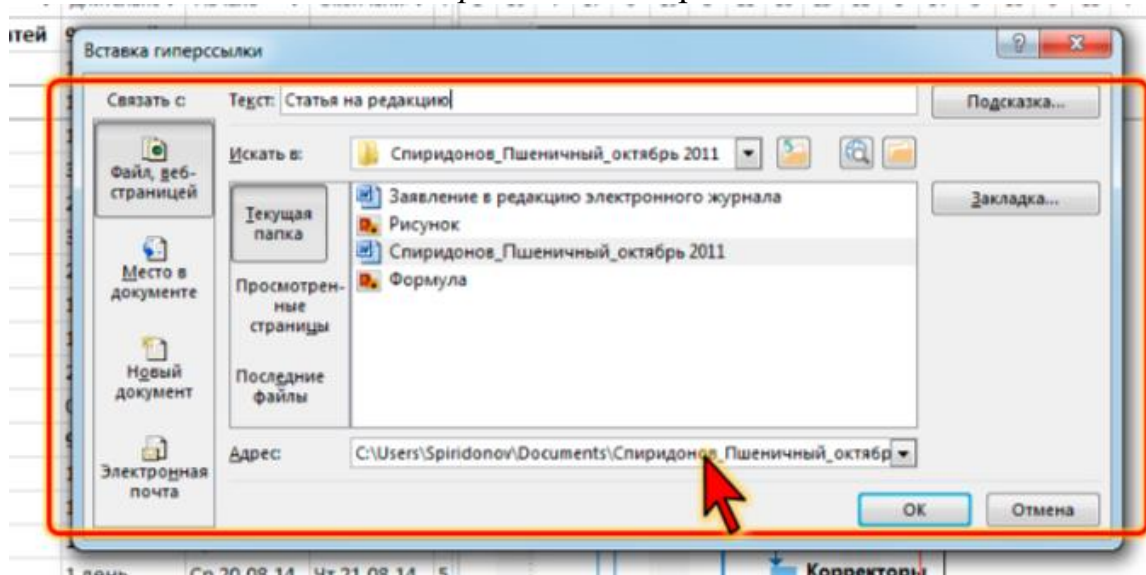
Оборудование: ПК, интернет, программное обеспечение – операционная система MS Project, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Создание гиперссылок

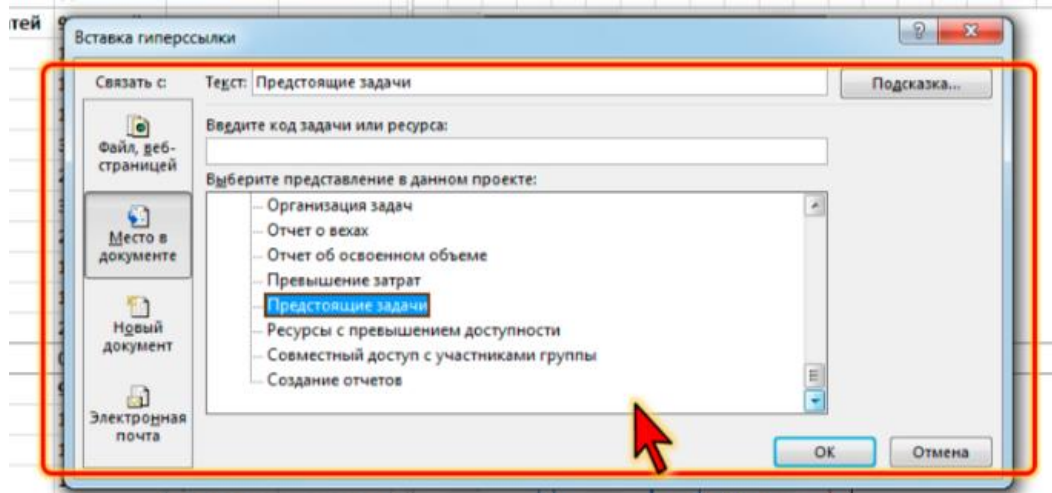
Гиперссылки можно создавать для задач, ресурсов и назначений. Особенно удобно это делать, когда документ, сопровождающий проект, слишком объемен, чтобы сохранять его в файле проекта. Гиперссылки можно создавать и на другие объекты: страницы Интернет, адреса электронной почты, различные элементы текущего документа.

1. По любой ячейке задачи (ресурса, назначения) щелкните правой кнопкой мыши и в контекстном меню выберите команду *Гиперссылка*.
2. В диалоговом окне *Гиперссылка* выберите тип ссылки *Файл, веб-страница*



3. При создании ссылки на файл с помощью элементов окна перейдите в папку, содержащую нужный файл, и выделите его. Путь к файлу будет отображен в поле *Адрес*.
4. В поле *Текст* можно ввести информацию, которая будет отображаться в виде всплывающей подсказки к гиперссылке.
5. Можно нажать кнопку *Подсказка* и в появившемся окне ввести дополнительную информацию, которая будет отображаться в виде всплывающей подсказки к гиперссылке.
6. Нажмите кнопку *OK*.
7. Интересна также возможность создания гиперссылки на место в текущем документе.
8. По любой ячейке задачи (ресурса, назначения) щелкните правой кнопкой мыши и в контекстном меню выберите команду *Гиперссылка*.

9. В диалоговом окне *Гиперссылка* выберите тип ссылки *Место в документе*



10. Выберите представление или отчет, на который создается гиперссылка.

11. Используя поле *Текст* и кнопку *Подсказка*, как и в предыдущем случае, можно ввести информацию, которая будет отображаться в виде всплывающей подсказки к гиперссылке.

12. Нажмите кнопку *OK*.

В таблицах задачи, ресурсы и назначения, к которым добавлены гиперссылки, выделены специальным значком в столбце *Индикаторы*. При щелчке мышью по значку гиперссылки на файл, он будет открыт. При щелчке мышью по значку гиперссылки на представление или отчет текущего проекта, произойдет переход к представлению или отчету.

Для удаления гиперссылки нужно выделить любую ячейку задачи (ресурса, назначения), на вкладке *Задача* в группе *Редактирование* нажать кнопку *Очистить* и в появившемся меню выбрать команду *Очистить гиперссылки*.

Задание 2. Создание настраиваемого поля

Не очень объемную информацию о задачах и ресурсах можно хранить в поле заметок. Это фактически настраиваемые текстовые поля, в них можно помещать фрагменты сопроводительного текста. Их главное преимущество состоит в том, что можно прикрепить к задаче внешние файлы или использовать команды форматирования текста. Однако структурировать, например, отфильтровать таблицу по информации, содержащейся в заметках, невозможно.

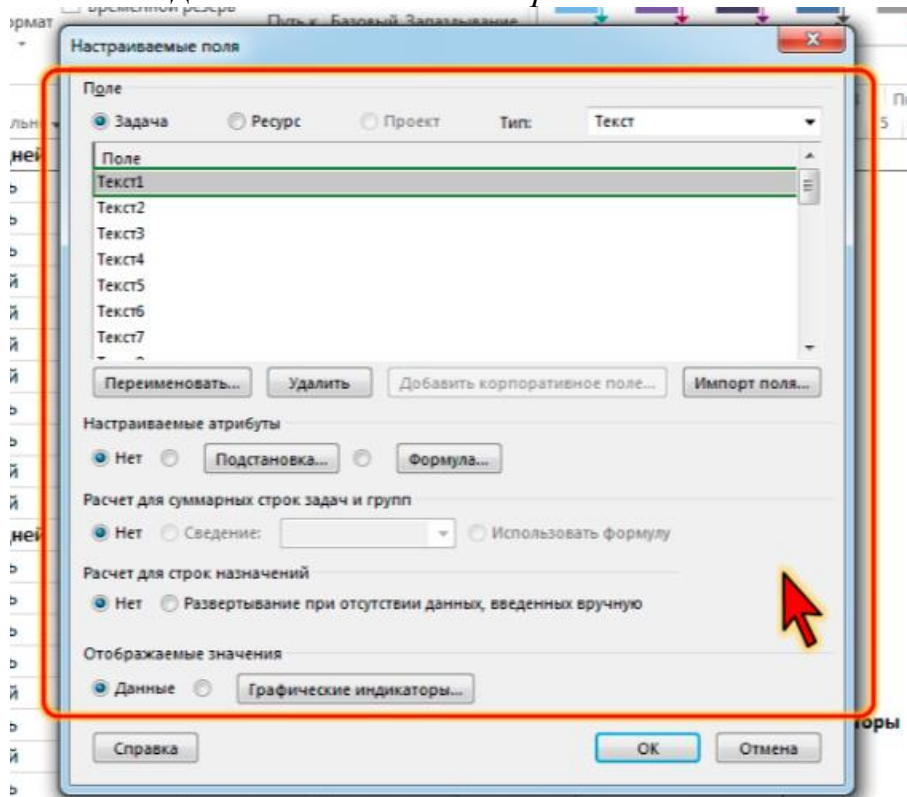
Использование настраиваемых полей (столбцов) позволяет легко структурировать информацию о задачах и ресурсах.

Например, необходимо знать, что конкретно делается в ходе исполнения каждой из задач, и какие результаты достигаются после ее завершения. Использовать заметки для хранения двух типов информации неудобно. Но можно создать два настраиваемых поля. Тогда по этим полям можно будет структурировать данные. И использовать их отдельно в дальнейшем.

Для создания нового поля проще всего на вкладке *Проект* в группе *Свойства* нажать кнопку *Настраиваемые поля* или нажать эту же

кнопку на вкладке *Формат* в группе *Столбцы*. Можно также в любом представлении, в котором отображается таблица задач или ресурсов, щелкнуть правой кнопкой мыши по названию столбца и в контекстном меню выбрать команду *Настраиваемые поля*.

Появится диалоговое окно *Настраиваемые поля*



Поля можно создавать как для таблиц задач, так и для таблиц ресурсов. В верхней части окна следует установить соответствующий переключатель.

Можно создавать поля различных типов. Тип создаваемого поля выбирается в соответствующем раскрывающемся списке в верхней части окна. Всего доступно девять типов полей.

Тип	Макс. число полей	Содержимое поля
Затраты	10	Данные о стоимости задачи
Дата	10	Даты
Длительность	10	Длительность, трудозатраты
Окончание	10	Даты
Флаг	20	Логические <i>Да</i> или <i>Нет</i>
Число	20	Числа
Начало	10	Даты
Текст	30	Текстовые данные
Кодировка справочника	10	Структурированные значения

Создаваемое поле должно иметь содержательной наименование. Для создания имени поля следует нажать кнопку *Переименовать* и в появившемся диалоговом окне ввести нужно имя.

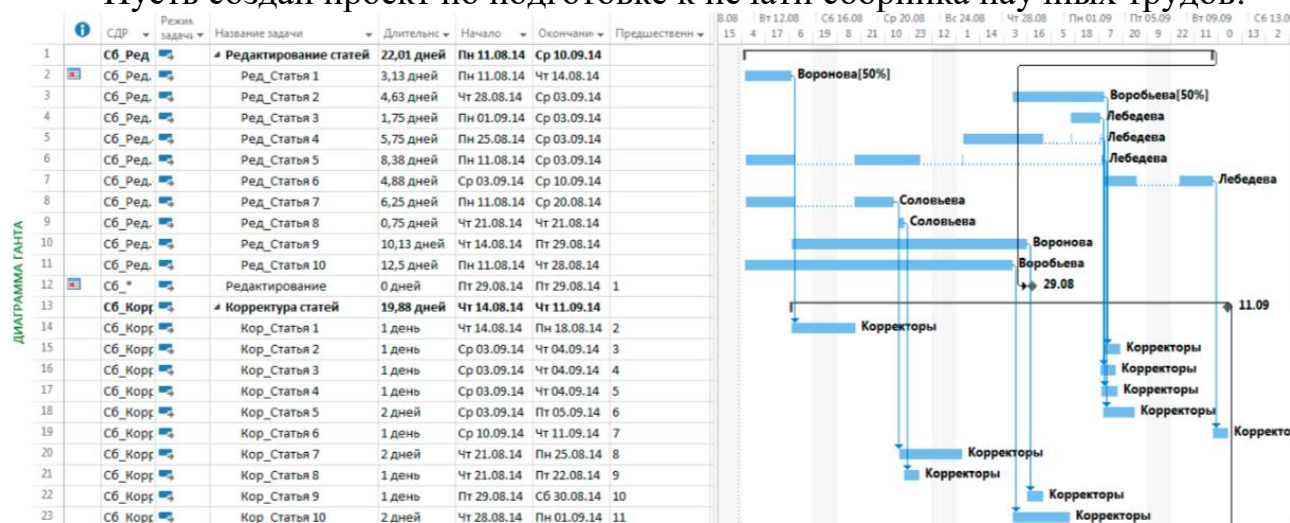
После этого в нижней части окна можно настроить отдельные атрибуты создаваемого поля. Следует иметь в виду, что для различных типов полей возможности настройки различаются.

После установки всех необходимых параметров, в окне *Настраиваемые поля* следует нажать кнопку *ОК*.

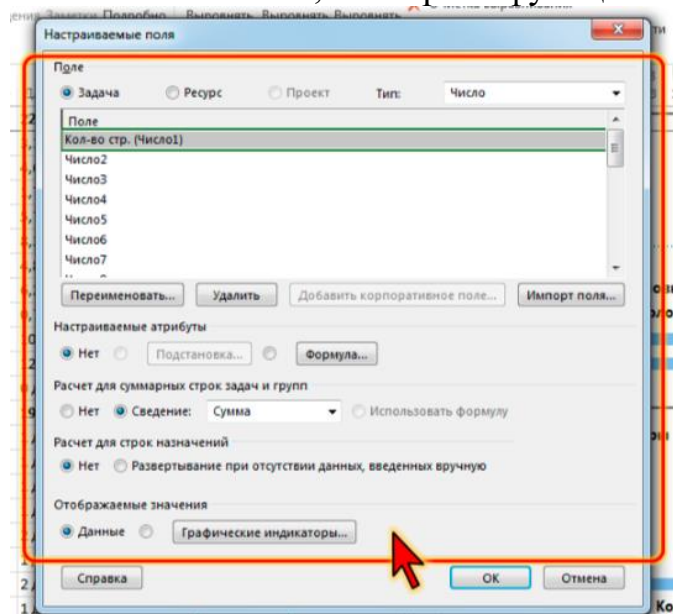
Следует обратить внимание, что созданное поле не будет автоматически отображено в таблице. Для его следует щелкнуть правой кнопкой мыши по заголовку столбца, перед которым добавляется столбец со значениями приоритета, в контекстном меню выбрать команду *Вставить столбец* и в появившемся списке выбрать имя созданного поля.

Задание 3. Использование формул

Пусть создан проект по подготовке к печати сборника научных трудов.



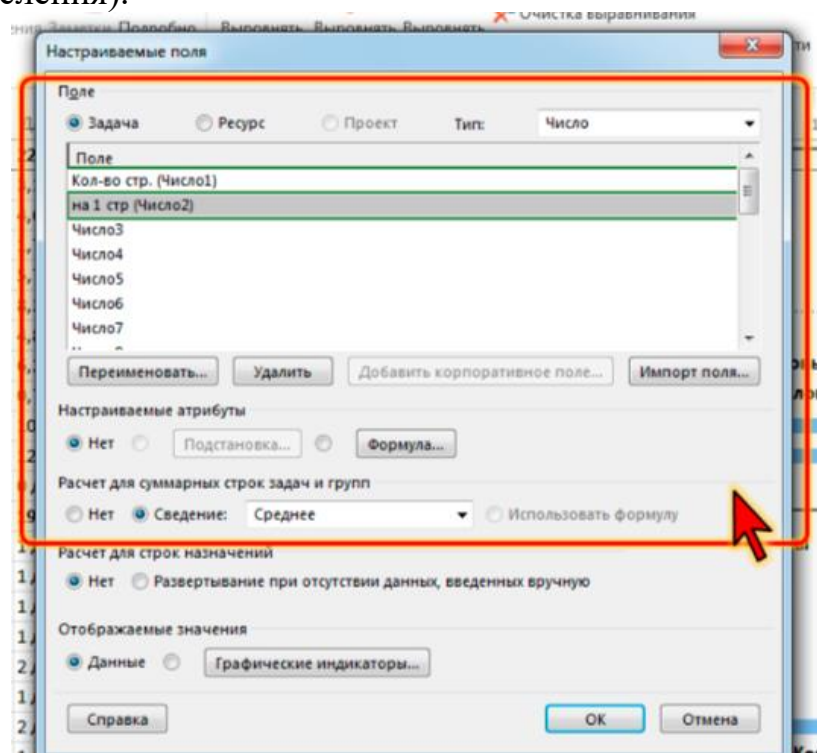
С помощью настраиваемых полей в таблицу задач был добавлен столбец *Кол-во стр.* Для обеспечения возможности вычислений с использованием значений в ячейках столбца, был выбран тип поля *Число*. Кроме того, для получения суммы трудозатрат для суммарной задачи, в разделе *Расчет для суммарных строк задач и групп* был установлен переключатель *Сведение*, и выбрана функция сведения *Сумма*.



В ячейки столбца *Кол-во стр* были введены необходимые данные.

В проекте требуется получить данные о длительности редактирования одной страницы каждой статьи. Эти значения могут быть рассчитаны как частное от деления длительности задачи на число страниц. Таким образом, с использованием настраиваемых полей необходимо создать новый столбец, ячейки которого будут содержать формулы.

1. В диалоговом окне *Настраиваемые поля* в раскрывающем списке *Тип* выберите параметр *Число* (поскольку будут производиться вычисления).

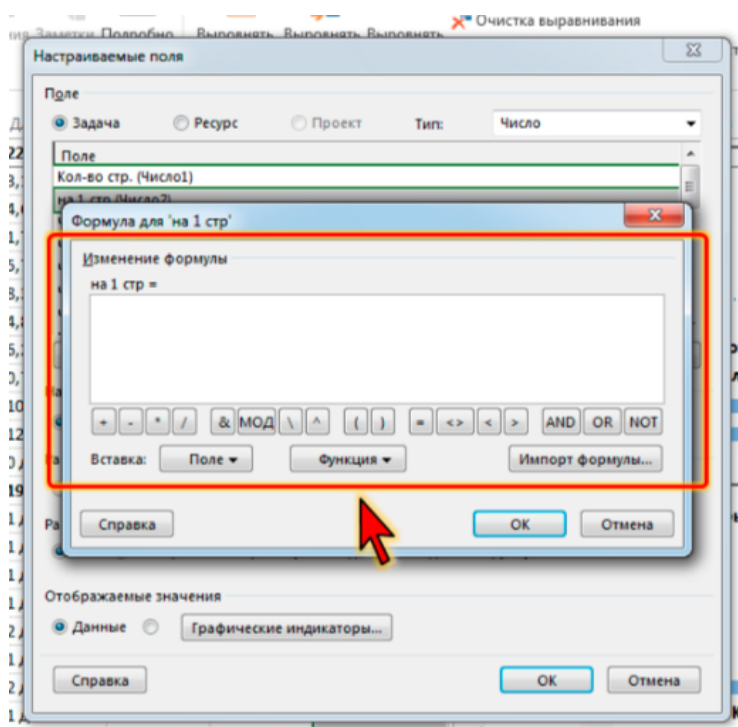


2. В списке *Поля* выберите *Число 2*, так как поле *Число 1* уже использовано для столбца *Число страниц*.

3. Нажмите кнопку *Переименовать* и в появившемся диалоговом окне введите имя столбца.

4. Для того чтобы в ячейке суммарной задачи можно было получить среднее значение трудозатрат на редактирование статей, в разделе *Расчет для суммарных строк задач и групп* установите переключатель *Сведение*, и выберите функцию сведения *Среднее*.

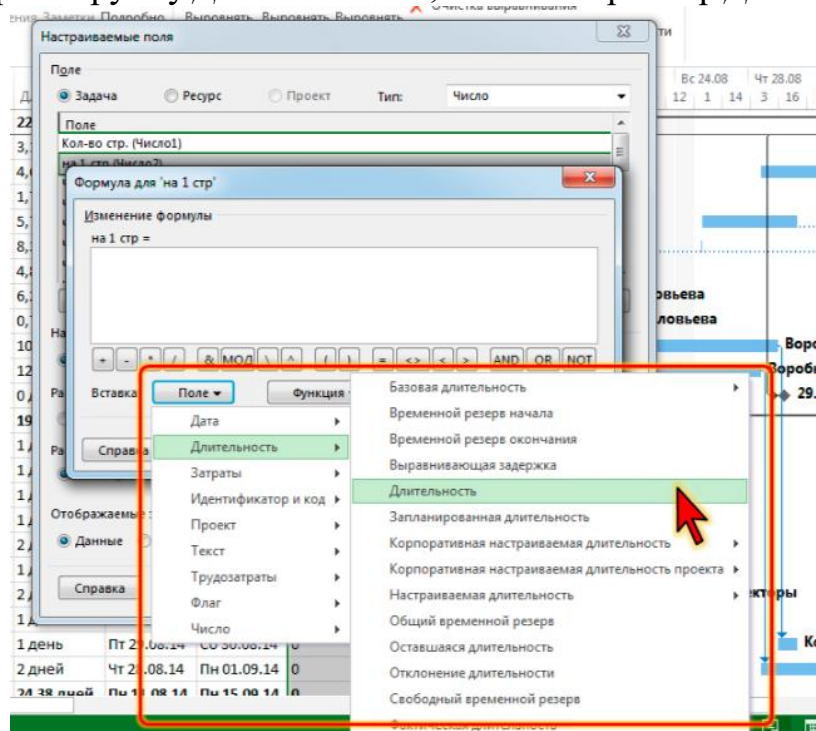
5. В разделе *Настраиваемые атрибуты* установите переключатель *Формула*. В появившемся предупреждении об удалении всех данных нажмите кнопку *OK*, поскольку никаких данных еще в этом столбце нет. Нажмите кнопку *Формула*, после чего появится диалоговое окно *Формула для <Имя поля>*. Кнопку *Формула* можно нажать, не устанавливая переключатель; тогда окно предупреждения появляться не будет.



С использованием элементов окна в формулу можно добавлять существующие поля задачи (раскрывающийся список *Поля*), простейшие арифметические и логические знаки (набор кнопок ниже поля формулы) и функции (раскрывающийся список *Функции*). Кроме того, с использованием кнопки *Импорт формулы* можно импортировать формулы из других проектов.

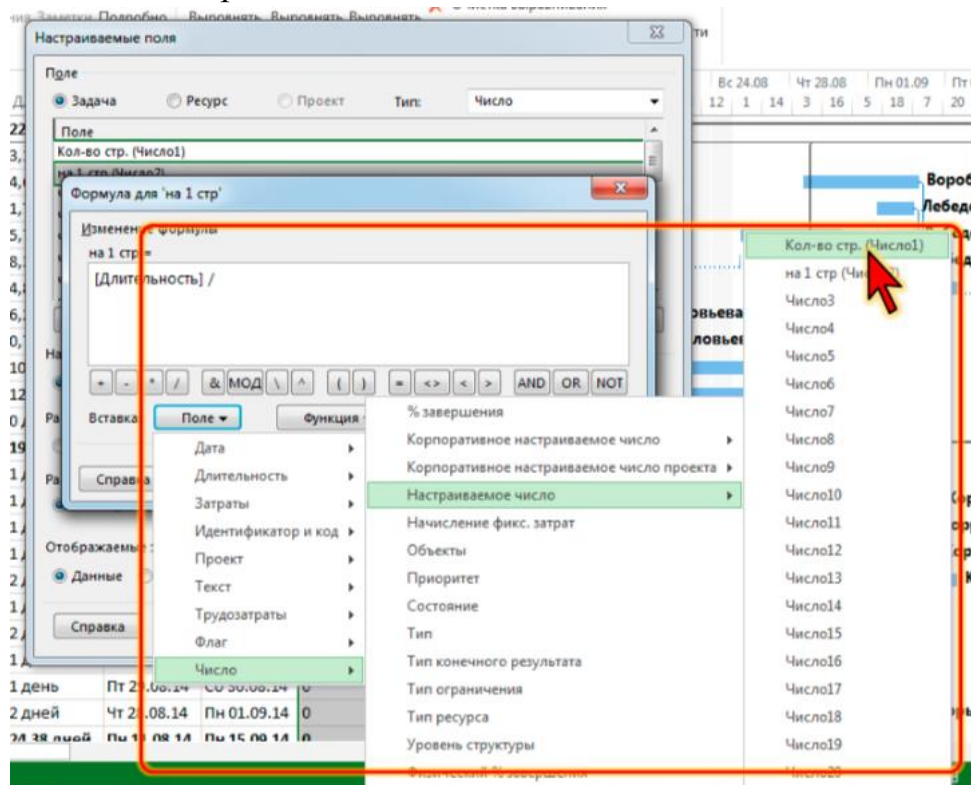
В рассматриваемой задаче необходимо поле *Длительность* разделить на поле *Кол-во стр.*

Для ввода делимого снова нажмите кнопку *Поля*, в появившемся меню выберите группу *Длительность*, а затем параметр *Длительность*.

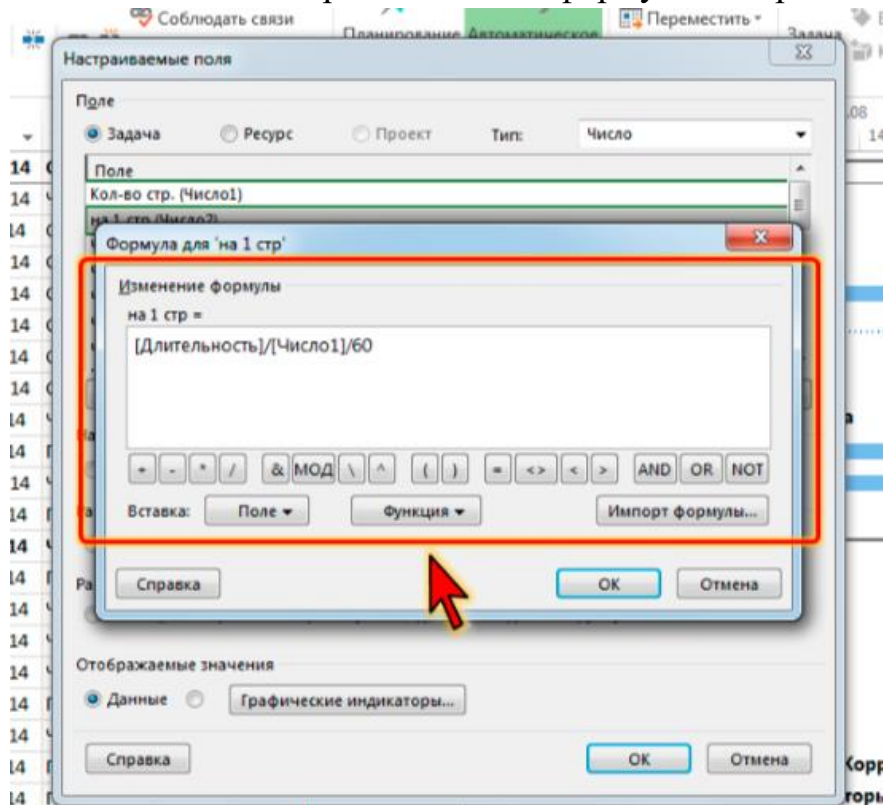


Для ввода знака деления щелкните по кнопке деления (дробь) или ведите символ с клавиатуры.

Для ввода делителя щелкните мышью по кнопке *Поле*, в появившемся списке выберите группу *Число*, затем группу *Настраиваемое число* и, наконец, *Кол-во стр*



Поскольку Project длительность рассчитывает в минутах, а результат расчета удобнее отображать в часах, после ввода поля *Кол-во стр* следует ввести деление на 60. В процессе ввода формула отображается в окне.

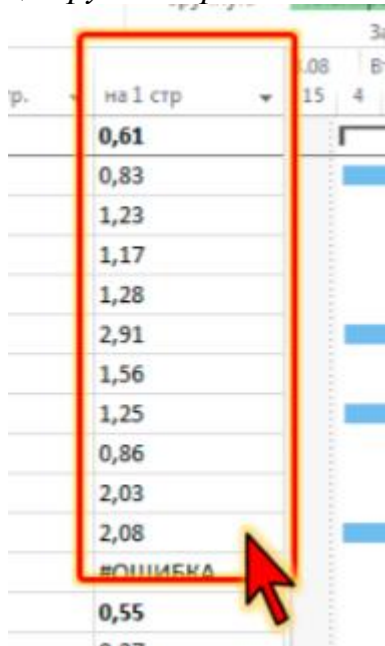


После завершения создания формулы в окне *Формула* для *<Имя поля>* нажмите кнопку *ОК*.

После возврата в окно *Настраиваемые поля* нажмите в нем кнопку *ОК*.

Созданное поле отображается как столбец таблицы стандартными средствами.

Таким образом, в таблице появился столбец *на 1 стр*, значения которого автоматически вычисляются на основе значений столбца *Трудозатраты* и *Число страниц*.



на 1 стр
0,61
0,83
1,23
1,17
1,28
2,91
1,56
1,25
0,86
2,03
2,08
#ОШИБКА
0,55

При изменении значений в указанных столбцах, автоматически будут изменяться и значения нового столбца.

Следует обратить внимание на следующие возможности и особенности создания вычисляемых настраиваемых полей.

При создании формулы в окне *Формула* для *<Имя поля>* для ввода аргументов и операторов не обязательно пользоваться элементами окна. Все это можно делать и с использованием клавиатуры.

При использовании функций следует строго соблюдать их синтаксис. Всего доступно несколько десятков функций, объединенных в пять групп: Функции преобразования, Функции даты и времени, Общие функции, Математические функции и Текстовые функции.

Задание 4

1. Откройте файл *Zadanie 13_04.mpp*.
2. Просмотрите заметку к задаче *Ред_Раздел 1*.
3. К задаче *Корр_Раздел 1* добавьте заметку с текстом *Получить статью от редактора*.
4. К задаче *Ред_Раздел 3* в качестве заметки добавьте документ *Spravka2.docx*.
5. В заметке к задаче *Ред_Раздел 1* удалите вложенный документ и вместо него добавьте документ *Spravka5.docx*.
6. Удалите заметку к задаче *Корр_Раздел 2*.

7. Просмотрите гиперссылку к задаче *Обложка*.
8. При наличии подключения к Интернет просмотрите гиперссылку к задаче *Заседание редсовета*.
9. К задаче *Корр_Раздел 3* добавьте гиперссылку на документ Spravka1.docx.
10. При наличии подключения к Интернет к задаче *Корректурa* добавьте гиперссылку сайт Википедии.
11. Удалите гиперссылку к задаче *Ред_Раздел 2*.
12. Сохраните проект как файл под именем Lesson_13_04.

Задание 5

1. Откройте файл Zadanie_13_05.mpp.
2. В таблице *Запись* представления *Диаграмма Ганта* с помощью настраиваемых полей добавьте столбец для ввода количества страниц каждого раздела, а для суммарной задачи *Редактирование* – среднее значение количества страниц. Поместите столбец справа от столбца *Длительность*.
3. Введите количество страниц для каждого раздела.
4. В таблице *Запись* представления *Лист ресурсов* с помощью настраиваемых полей добавьте столбец для указания должности каждого сотрудника. Поместите столбец справа от столбца *Название ресурса*.
5. Для первых четырех сотрудников введите должность *Редактор*.
6. Сохраните проект как файл под именем Lesson_13_05.

Задание 6

1. Откройте файл Zadanie_13_06.mpp.
2. В таблице *Запись* представления *Диаграмма Ганта* с помощью настраиваемых полей добавьте столбец, в котором будут автоматически рассчитаны затраты на одну страницу при редактировании и корректуре. Поместите столбец справа от столбца *Кол-во страниц*.
3. Сохраните проект как файл под именем Lesson_13_06.

ПРАКТИЧЕСКАЯ РАБОТА № 14

Тема: Определение состава задач проекта, структурирование списка задач, СДР-коды, ввод задач в программе MS Project

Цель работы: научить студентов определять состав задач проекта в среде MS Project; овладение навыками структурирования списка задач проекта, использование СДР-кодов и ввода задач в проект.

Оборудование: ПК, интернет, программное обеспечение – операционная система MS Project, MS Word, инструкции по выполнению работы.

Содержание работы:

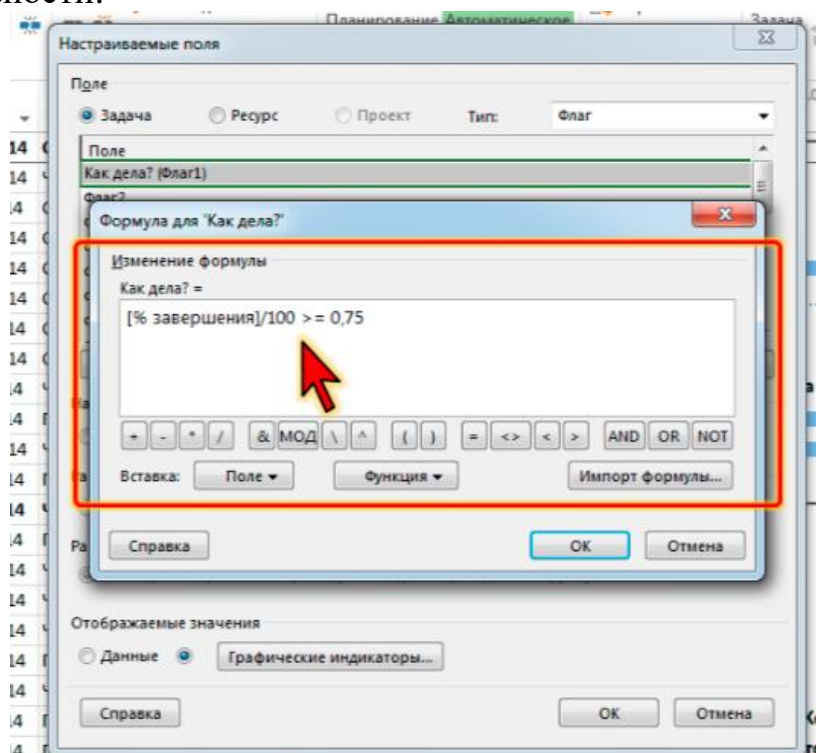
Задание 1. Использование индикатора

Настраиваемые поля можно использовать для наглядного графического представления данных путем отображения различных индикаторов вместо значений. Вид индикатора при этом зависит от выполнения или невыполнения заданных логических условий.

Например, в таблице задач требуется добавить столбец, в котором будет отображаться состояние готовности статьи. Например, при готовности более 75 % автоматически должен отображаться один значок, а при меньшей готовности - другой.

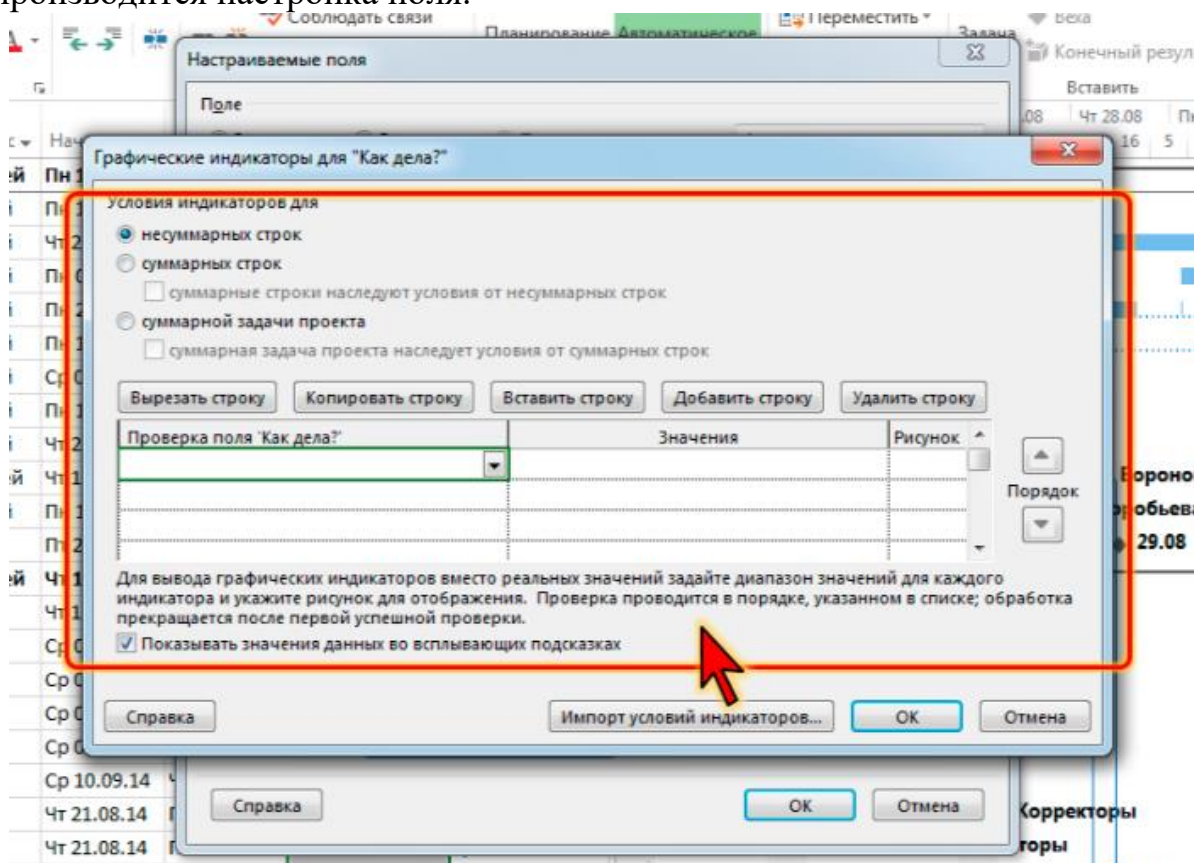
При создании такого поля следует в раскрывающемся списке *Тип* диалогового окна *Настраиваемые поля* выбрать параметр *Флаг*, в списке *Поля* выбрать свободное поле *Флаг*, нажать кнопку *Переименовать* и в появившемся диалоговом окне ввести имя поля.

Затем необходимо создать формулу, обеспечивающую проверку готовности.



После создания формулы в нижней части окна *Настраиваемые поля* необходимо нажать кнопку *Цветовые индикаторы* после чего появится

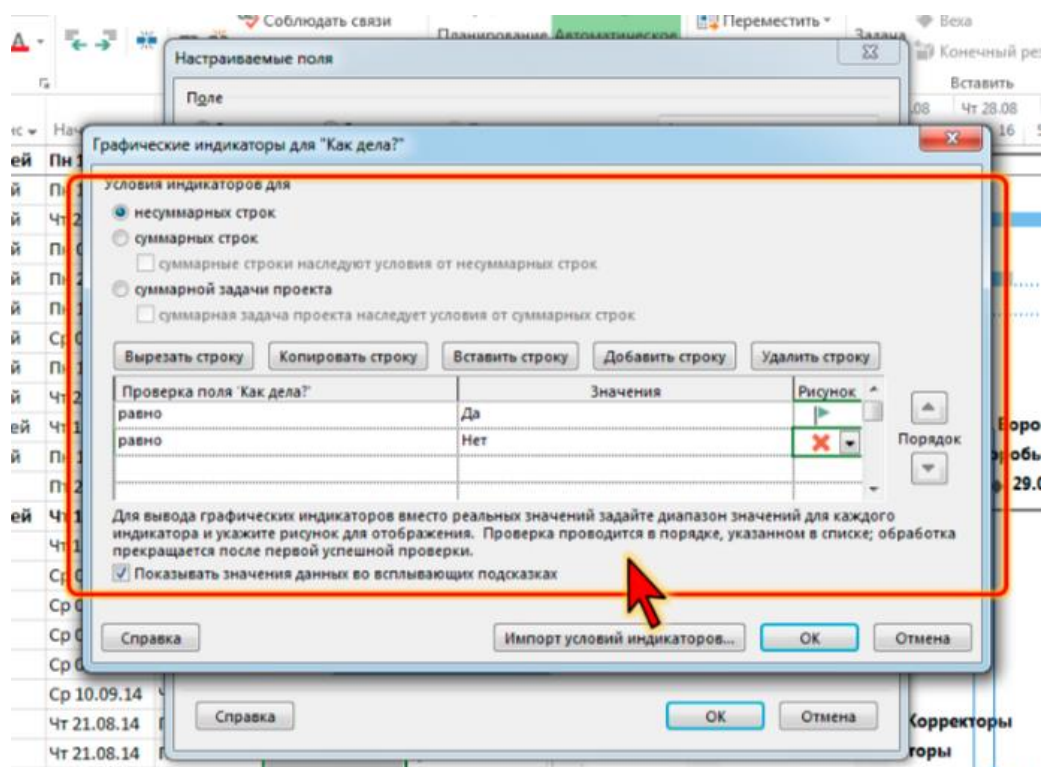
диалоговое окно *Графические индикаторы для <Имя поля>*, в котором и производится настройка поля.



В верхней части окна есть переключатель для типа задач (суммарные или не суммарные). Для каждого из них можно установить свои критерии. Если установлены оба флажка в этом переключателе, то тем самым, настраивая параметры обычных задач, автоматически будут заданы параметры для всех остальных задач проекта.

Если нужно увидеть значение поля, скрытого индикатором, следует установить флажок Показывать значения данных во всплывающих подсказках. Данные будут отображаться при наведении указателя мыши на индикатор.

Для настройки отображения графических индикаторов в столбце *Проверка поля* следует выбрать условие. В данном случае выбирается *равно*, поскольку надо проверить, выполняется ли введенная формула. Затем в столбце *Значение* следует выбрать проверяемое значение. В данном случае проверяется выполнение созданной формулы, то есть, если формула выполняется (готовность более 75 %) выбирают значению *Да*, а если нет – значение *Нет*. Наконец, в столбце *Рисунок* выбирают вид индикатора



После создания нового настраиваемого поля его следует в обычном порядке вставить в таблицу как столбец. Таким образом, в столбце *Как дела?* будет отображаться индикатор, внешний вид которого показывает, достигнута ли 75-процентная готовность статьи

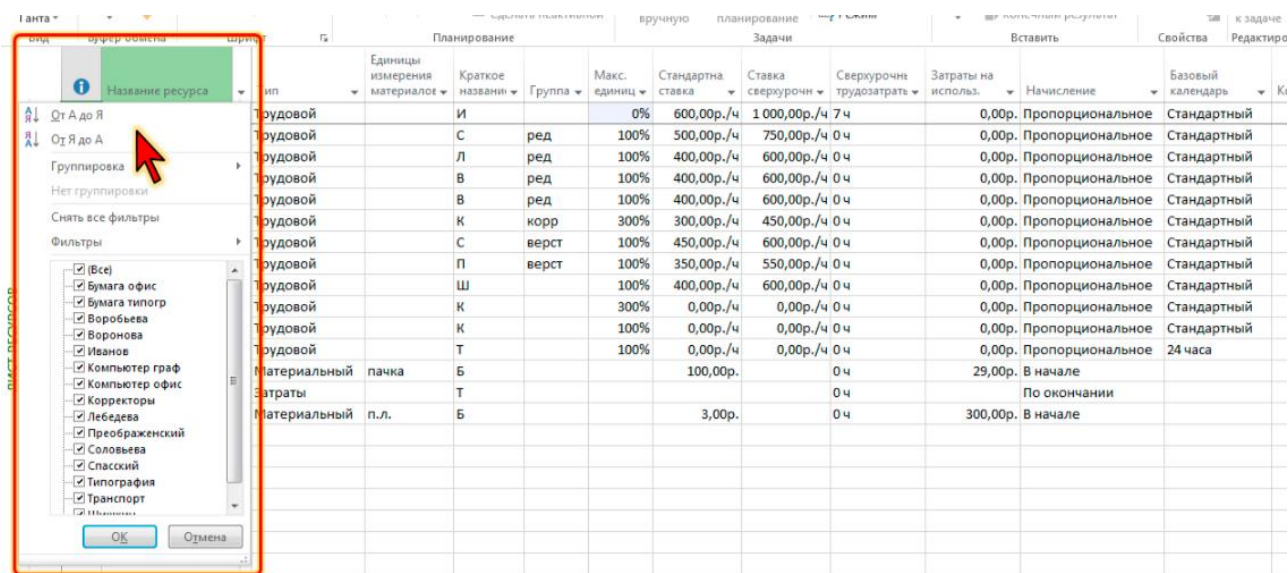
Окончание	% завершения	Как дела?
1.08.14 Ср 10.09.14	51%	
1.08.14 Чт 14.08.14	85%	▶
3.08.14 Ср 03.09.14	5%	✗
1.09.14 Ср 03.09.14	0%	✗
5.08.14 Ср 03.09.14	10%	✗
1.08.14 Ср 03.09.14	50%	✗
3.09.14 Ср 10.09.14	50%	✗
1.08.14 Ср 20.08.14	33%	✗
1.08.14 Чт 21.08.14	0%	✗
4.08.14 Пт 29.08.14	80%	▶
1.08.14 Чт 28.08.14	75%	▶
9.08.14 Пт 29.08.14	0%	✗
4.08.14 Чт 11.09.14	0%	✗
4.08.14 Пн 18.08.14	0%	✗
3.09.14 Чт 04.09.14	0%	✗

Задание 2. Сортировка, группировка и отбор данных

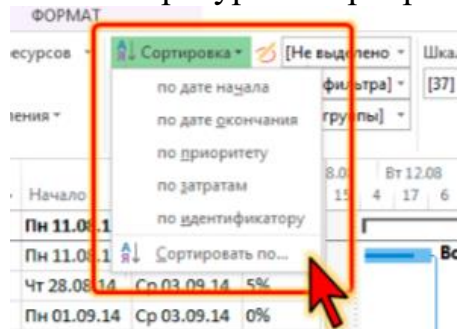
1. Сортировка данных

В простейшем случае таблицу можно сортировать по данным одного столбца.

- Щелкните по стрелке, отображающейся в правой части ячейки заголовка столбца, по данным которого производится сортировка.
- В появившемся меню выберите направление сортировки.



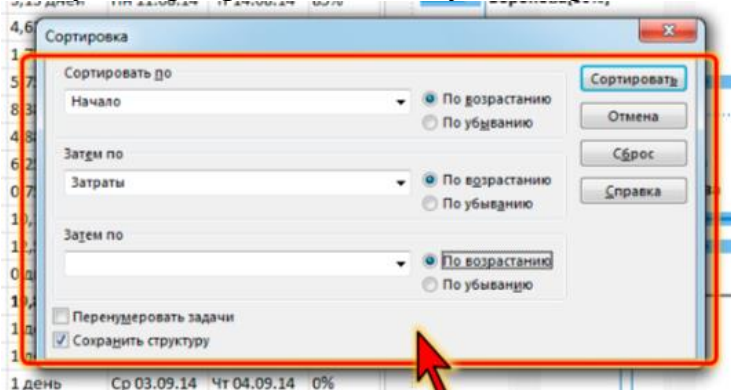
Для сортировки таблиц можно использовать кнопку *Сортировка* в группе *Данные* вкладки *Вид*. При щелчке по этой кнопке отображается меню с названиями столбцов, сортировка по которым представляется разумной в данном представлении. Например, таблицу со списком задач предлагается сортировать по данным одного из пяти столбцов. А, например, в таблицах представлений ресурсов сортировка возможна по трем столбцам.



Сортировку можно осуществлять по нескольким столбцам для группировки данных с одинаковыми значениями в одном столбце и последующего осуществления сортировки другого столбца в этих группах с одинаковыми значениями. Например, таблицу задач можно отсортировать сначала по данным столбца *Начало*, а потом *Затраты*.

1.Нажмите кнопку *Сортировка* в группе *Данные* вкладки *Вид* и выберите команду *Сортировать по*.

2.В раскрывающихся списках окна *Сортировка* выберите столбцы, по которым следует сортировать таблицу, и направления сортировки



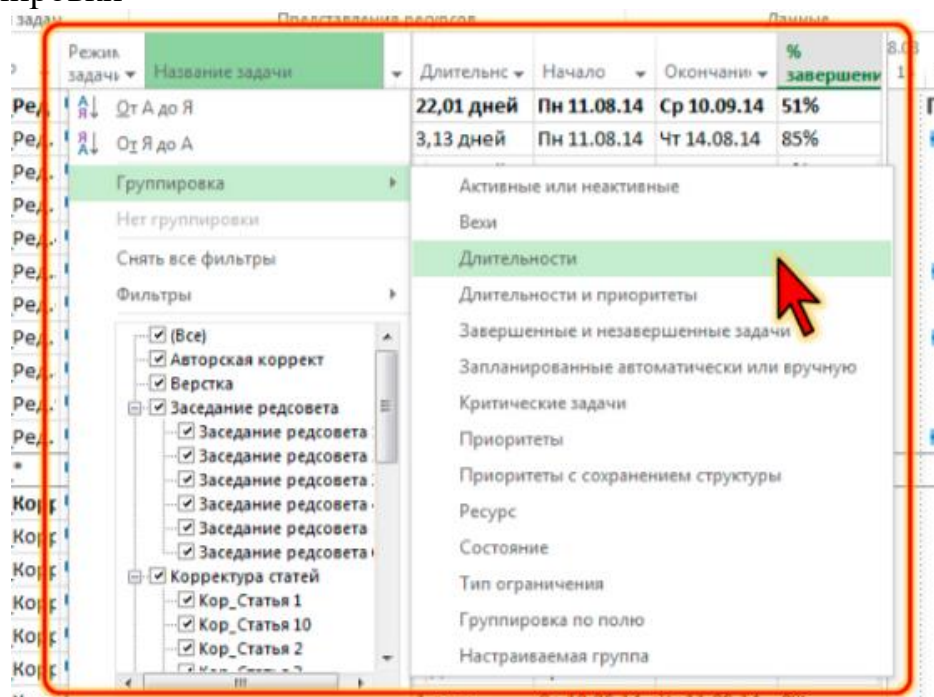
Невозможно сортировать выделенный диапазон таблицы.

Для восстановления исходного вида таблицы, измененного сортировкой, следует нажать кнопку *Сортировка* в группе *Данные* вкладки *Вид*, а затем в появившемся меню выбрать команду *По идентификатору*.

2. Группировка данных

Таблицы представлений Project можно структурировать, используя различные критерии группировки строк. Для этого используют разные способы.

Можно, например, щелкнуть по стрелке, отображающейся в правой части ячейки заголовка столбца, по данным которого производится группировка, в появившемся списке выбрать параметр *Группировка*, а затем способ группировки

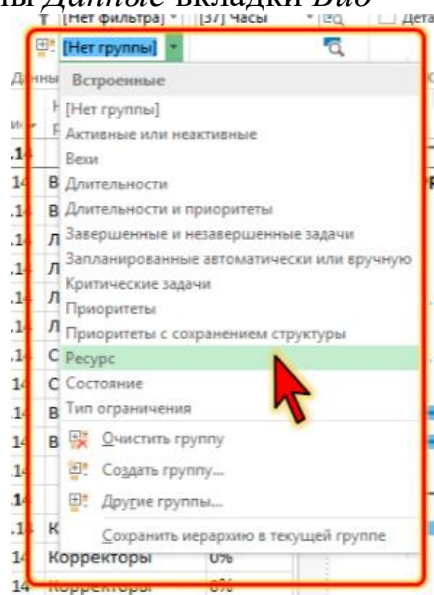


После группировки в таблице появятся новые строки групп

		Длительность: 0 дней	0д	Пт 29.08.14	Пт 29.08.14	0%
12	Сб_*	Редактирование	0 дней	Пт 29.08.14	Пт 29.08.14	0%
		Длительность: 0,25 дней	24,38д	Пн 11.08.14	Пн 15.09.14	0%
25	Сб_*.1	Заседание редсовета 1	2 ч	Пн 11.08.14	Пн 11.08.14	0%
26	Сб_*.2	Заседание редсовета 2	2 ч	Пн 18.08.14	Пн 18.08.14	0%
27	Сб_*.3	Заседание редсовета 3	2 ч	Пн 25.08.14	Пн 25.08.14	0%
28	Сб_*.4	Заседание редсовета 4	2 ч	Пн 01.09.14	Пн 01.09.14	0%
29	Сб_*.5	Заседание редсовета 5	2 ч	Пн 08.09.14	Пн 08.09.14	0%
30	Сб_*.6	Заседание редсовета 6	2 ч	Пн 15.09.14	Пн 15.09.14	0%
		Длительность: 0,75 дней	0,75д	Чт 21.08.14	Чт 21.08.14	0%
9	Сб_Ред.	Ред_Статья 8	0,75 дней	Чт 21.08.14	Чт 21.08.14	0%
		Длительность: 1 день	19,88д	Чт 14.08.14	Чт 11.09.14	0%
14	Сб_Корр	Кор_Статья 1	1 день	Чт 14.08.14	Пн 18.08.14	0%
15	Сб_Корр	Кор_Статья 2	1 день	Ср 03.09.14	Чт 04.09.14	0%
16	Сб_Корр	Кор_Статья 3	1 день	Ср 03.09.14	Чт 04.09.14	0%
17	Сб_Корр	Кор_Статья 4	1 день	Ср 03.09.14	Чт 04.09.14	0%
19	Сб_Корр	Кор_Статья 6	1 день	Ср 10.09.14	Чт 11.09.14	0%
21	Сб_Корр	Кор_Статья 8	1 день	Чт 21.08.14	Пт 22.08.14	0%
22	Сб_Корр	Кор_Статья 9	1 день	Пт 29.08.14	Сб 30.08.14	0%
		Длительность: 1,75 дней	1,75д	Пн 01.09.14	Ср 03.09.14	0%
4	Сб_Ред.	Ред_Статья 3	1,75 дней	Пн 01.09.14	Ср 03.09.14	0%
		Длительность: 2 дней	12,13д	Чт 21.08.14	Пт 05.09.14	0%
18	Сб_Корр	Кор_Статья 5	2 дней	Ср 03.09.14	Пт 05.09.14	0%

По разным столбцам группировка производится по разным критериям. Не по всем столбцам возможна группировка.

Для группировки по наиболее часто используемым критериям можно использовать соответствующий раскрывающийся список группы *Данные* вкладки *Вид*



С помощью команды *Создать группу* можно создавать и собственные критерии группировки.

Независимо от способа группировки, для разгруппировки таблицы можно воспользоваться параметром *Нет группировки* в раскрывающемся списке столбца, по которому произведена группировка, или параметром *[Нет группы]* раскрывающегося списка *Группировка*.

3.Отбор данных

Отбор данных, соответствующие какому-либо критерию, в таблицах производится с использованием фильтров и выделения. Следует иметь в виду, что невозможно одновременно использовать фильтр и выделение.

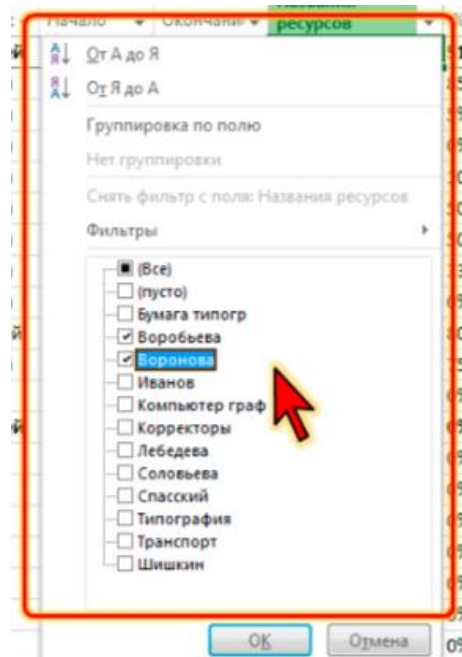
При фильтрации в таблице будут отображены только данные, соответствующие выбранному критерию, а остальные данные будут скрыты. При выделении нужные данные выделяются среди прочих.

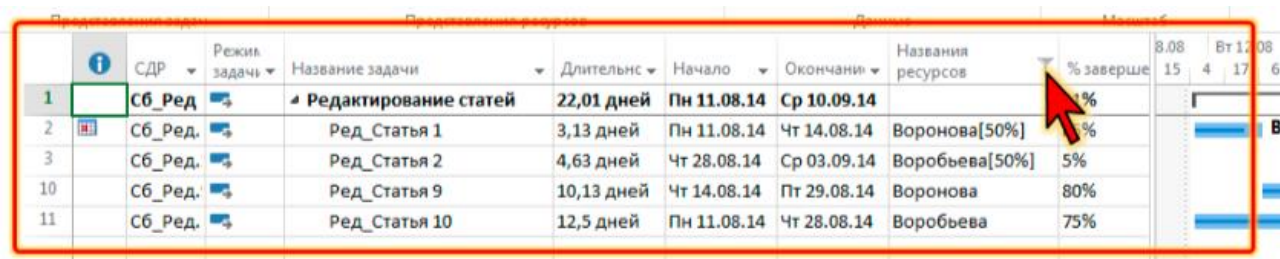
В простейшем случае фильтрацию произвести можно по данным одного столбца.

1.Щелкните по стрелке, отображающейся в правой части ячейки заголовка столбца, по данным которого производится отбор.

2.Установите флажки для данных, которые будут отображаться в таблице

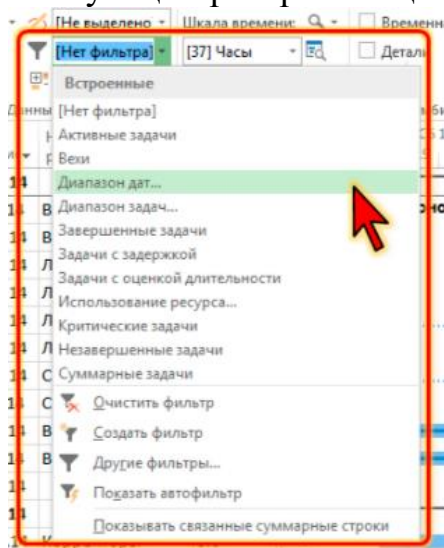
В таблице будут показаны только строки, соответствующие установленному фильтру. В правой части ячейки заголовка столбца, по которому проведена фильтрация, вместо стрелки отображается значок фильтра





	SDR	Режим задачи	Название задачи	Длительность	Начало	Окончание	Названия ресурсов	% заверше
1	СБ_Ред.		Редактирование статей	22,01 дней	Пн 11.08.14	Ср 10.09.14		15%
2	СБ_Ред.		Ред_Статья 1	3,13 дней	Пн 11.08.14	Чт 14.08.14	Воронова[50%]	15%
3	СБ_Ред.		Ред_Статья 2	4,63 дней	Чт 28.08.14	Ср 03.09.14	Воробьева[50%]	5%
10	СБ_Ред.		Ред_Статья 9	10,13 дней	Чт 14.08.14	Пт 29.08.14	Воронова	80%
11	СБ_Ред.		Ред_Статья 10	12,5 дней	Пн 11.08.14	Чт 28.08.14	Воробьева	75%

Фильтры можно использовать в нескольких столбцах одновременно. Для отбора по наиболее часто используемым критериям можно использовать соответствующий раскрывающийся список группы *Данные* вкладки *Вид*



Задание 3

1. Откройте файл *Zadanie_14_03.mpp*.
2. В таблице *Затраты* представления *Лист задач* с помощью настраиваемых полей добавьте столбец, в котором будут отображаться графические индикаторы затрат на 1 страницу: если затраты ниже 60 рублей – зеленый кружок, а если выше 60 рублей – красный. Поместите столбец справа от столбца *Затраты на 1 стр.*
3. Сохраните проект как файл под именем *Lesson_14_03*.

Задание 4

1. Откройте файл *Zadanie_14_04.mpp*.
2. Таблицу *Запись* представления *Диаграмма Ганта* отсортируйте по длительности задач.
3. К таблице *Запись* представления *Лист ресурсов* примените многоуровневую сортировку: сначала по типу ресурсов, затем по должности, а в пределах должности – по алфавиту.
4. Сохраните проект как файл под именем *Lesson_14_04*.

Задание 5

1. Откройте файл *Zadanie_14_05.mpp*.
2. К таблице *Запись* представления *Лист ресурсов* сгруппируйте ресурсы по должности.

3. Таблицу *Запись* представления *Диаграмма Ганта* сгруппируйте задачи по длительности.
4. Сохраните проект как файл под именем *Lesson_14_05*.

Задание 6

1. Откройте файл *Zadanie_14_06.mpp*.
2. Разгруппируйте записи таблицы *Запись* представления *Диаграмма Ганта*.
3. Разгруппируйте записи таблицы *Запись* представления *Лист ресурсов*.
4. Сохраните проект как файл под именем *Lesson_14_06*.

Задание 7

1. Откройте файл *Zadanie_14_07.mpp*.
2. В таблице *Запись* представления *Диаграмма Ганта* с использованием фильтра отобразите задачи, выполняемые *Петровым*.
3. В таблице *Запись* представления *Лист ресурсов* с использованием фильтра отобразите только корректоров.
4. Сохраните проект как файл под именем *Lesson_14_07*.

Задание 8

1. Откройте файл *Zadanie_14_08.mpp*.
2. В таблице *Запись* представления *Диаграмма Ганта* отобразите все задачи.
3. В таблице *Запись* представления *Лист ресурсов* отобразите все ресурсы.
4. Сохраните проект как файл под именем *Lesson_14_08*.

Задание 9

1. Откройте файл *Zadanie_14_09.mpp*.
2. В таблице *Запись* представления *Диаграмма Ганта* выделите задачи, выполняемые *Петровым*.
3. В таблице *Запись* представления *Лист ресурсов* снимите выделение отдельных ресурсов.
4. Сохраните проект как файл под именем *Lesson_14_09*.

Задание 10

1. Откройте файл *Zadanie_14_10.mpp*.
2. В таблице *Запись* представления *Диаграмма Ганта* настройте отображение только задач первого уровня без отображения подзадач.
3. Сохраните проект как файл под именем *Lesson_14_10*.

Задание 11

1. Откройте файл *Zadanie_14_11.mpp*.
2. В таблице *Запись* представления *Диаграмма Ганта* отобразите все задачи и подзадачи.
3. Сохраните проект как файл под именем *Lesson_14_11*.
4. Закройте Project.

ПРАКТИЧЕСКАЯ РАБОТА № 15

Тема: Выполнение ресурсного и бюджетного планирования в программе MS Project

Цель работы: научить студентов использовать MS Project для составления бюджета проекта, определять ресурсы проекта.

Оборудование: ПК, интернет, программное обеспечение – операционная система MS Project, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Разработать бюджет проекта

1. Назначить ресурсы проекта. В качестве ресурсов в Microsoft Project обычно используется рабочее время создателей проекта. Кроме рабочего времени к ресурсам относят механизмы, материалы, оплату различного вида работ, выполняемых сторонними организациями.

Для ввода данных о ресурсах в меню Вид необходимо выбрать форму представления Лист ресурсов:

6	Название ресурса	Тип	Единицы измерения материалов	Краткое название	Группа	Макс. единиц	Стандартная ставка	Ставка сверхурочных	Затраты на испол.	Начисление	Базовый календарь	Код

Ввод ресурсов можно произвести копированием из таблицы

№	Название ресурса	Тип	Группа	Единицы измерения материалов	Макс. единиц	Стандартная ставка, р./ч	Ставка сверхурочных, р./ч	Затраты на использование, Р-	Начисление	Календарь
1	Писатель	Трудовой	Люди		100%	по вариантам (таблица Д5.)			По оконч.	20XX
2	Редактор	Трудовой	Люди		100%				Пропорц.	20XX
3	Художник	Трудовой	Люди		100%				Пропорц.	20XX
4	Верстальщик	Трудовой	Люди		100%				Пропорц.	20XX
5	Корректор	Трудовой	Люди		100%				Пропорц.	20XX
6	Бумага	Материальный	Материалы	пачка	-				Пропорц	-

Стоимость использования ресурсов можно взять из таблицы

№	Название ресурса	Стандартная ставка, р./ч			Ставка сверхурочных в 2 раза выше стандартной ставки	Ставка сверхурочных, р./ч		Затраты на использование, р.	
		Вар. 0-5	Вар. 6-10	Вар. 11-15		Вар. 0-5		Вар. 6-10	Вар. 11-15
1	Писатель	-	-	-		25000		30000	20000
2	Редактор	200	170	220		-		-	-
3	Художник	250	270	300		-		-	-
4	Верстальщик	220	200	350		-		-	-
5	Корректор	220	200	350		-		-	-
6	Бумага	200	220	310	-	-		-	-

Для создания нового ресурса можно два раза щелкнуть по строке таблицы ресурсов левой кнопкой мыши и заполнить диалоговое окно *Сведения о ресурсе/Общее*.

В этом диалоговом окне указываются: название ресурса; тип ресурса – трудовой или материальный; доступность ресурса – временной период и количество ресурса, выделенного для проекта.

Назначение ресурсов задачам производим в представлении Диаграмма Ганта меню Ресурс нажатием кнопки Назначить ресурс, после которого появляется диалоговое окно. В нем и производим назначение ресурсов.

Использование ресурсов в проекте можно увидеть в представлении Использование ресурсов.

Далее назначаем базовый план. Задание базового плана производим в меню Проект кнопкой Задать базовый план.

Задание 2. Планирование работ проекта

1. Запустите Microsoft Project. Поместите в рабочем окне системы панель представлений – пункт меню Вид – Панель представлений. Задайте ключевые параметры проекта в окне сведений о проекте. Номер варианта проекта указывает преподаватель. Дата начала проекта – текущая дата.
2. Настройте календарь проекта. Откройте окно изменения рабочего времени – Сервис / Изменить рабочее время и установите праздничные нерабочие дни для календаря Стандартный.
3. Введите перечень задач проекта согласно таблице соответствующего варианта. Расположите задачи таким образом, чтобы их порядок соответствовал последовательности выполнения, а после каждой фазы были бы перечислены входящие в нее вехи и задачи. Выберите на панели представлений Диаграмму Ганта. В столбец Название задачи последовательно введите названия задач из таблицы соответствующего варианта. Фазы в таблицах выделены полужирным курсивом, а вехи имеют нулевую длину.
4. Преобразуйте задачи в фазу. Для этого все подзадачи этой фазы должны следовать в таблице непосредственно после нее. Удерживая нажатой левую кнопку мыши в области номеров задач, выделите строки задач, которые входят

в фазу. Нажмите кнопку На уровень ниже на панели инструментов Форматирование.

5. Создайте связи между задачами при помощи столбца Предшественник в таблице ввода данных в соответствии с таблицей варианта.

6. Назначьте стили текста, выделив разными цветами фазы, вехи, критические и некритические задачи. Для заголовков строк и столбцов установите жирный шрифт коричневого цвета, для суммарных задач – малиновый цвет, для нижнего уровня шкалы времени – темно-синий.

Вариант 1. Создать проект «Строительство дома», предназначенный для управления строительством частного одноэтажного жилого дома площадью 200м². Перечень задач проекта, их связи и длительности приведены в таблице.

Название задачи	Длительность (дней)	Предшественники
1	2	3
1. Начало проекта	0	
2. Утверждение проектов		
3. Начало утверждения проектов	1	1
4. Утверждение проекта на строительство	90	3
5. Утверждение проекта на газ	60	3
6. Утверждение проекта на водопровод и канализацию	30	3
7. Утверждение проекта на отопление	45	3
8. Проекты утверждены	0	4; 5; 6; 7
9. Строительство фундамента		
10. Начало закладки фундамента	2	8
11. Рытье траншей	10	10
12. Заливка фундамента	5	11
13. Фундамент завершен	0	12
14. Каркас и крыша		
15. Начало каркаса	1	13
16. Кладка стен	60	15
17. Перекрытие стен	15	16
18. Установка крыши	30	17
19. Установка наружных дверей и окон	7	17
20. Установка полов	5	17
21. Каркас готов	0	18; 19; 20
22. Коммуникации		
23. Начало установки коммуникаций	1	21
24. Проведение и подключение водопровода и канализации	10	23

25. Установка и подключение электропроводки	5	23
26. Установка и подключение газовых коммуникаций	5	23
27. Коммуникации готовы	0	24; 25; 26
28. Внутренняя отделка		
29. Начало отделки	0	27
30. Внутренние двери	10	29
31. Навесные потолки	5	30
32. Отделка стен	3	30
33. Монтаж отопления	10	30
34. Установка оборудования, приборов, сантехники	5	31; 33
35. Настил полов	15	32; 34
36. Конец отделки	0	35
37. Конец проекта	0	36

Между работами 12 и 13 установить задержку в 30 дней, необходимую для выдержки фундамента. Для задачи 32 установить ограничение Как можно позже.

Вариант 2. Создать проект «Внедрение бухгалтерской системы», предназначенный для автоматизации бухгалтерии небольшого предприятия, состоящей из 10 человек. Перечень задач проекта, их связи и длительности приведены в таблице.

Название задачи	Длительность (дней)	Предшественники
1	2	3
1. Начало проекта	0	
2. Выбор системы		
3. Изучение рынка бухгалтерских систем	7	1
4. Составление требований к бухгалтерским системам	7	1
5. Консультации с фирмами-разработчиками	7	3; 4
6. Принятие окончательного решения	2	5
7. Выбор завершен	0	6
8. Приобретение программного обеспечения		
9. Заключение договоров	6	2
10. Оплата за ПО	2	9
11. Оформление ПО на баланс	3	10
12. Приобретение ПО завершено	0	11
13. Приобретение компьютеров и сетевого оборудования		
14. Сбор информации о поставщиках и предложениях	7	7

15. Анализ и выбор поставщика	5	14
16. Заключение договоров	5	15
17. Оплата за оборудование	2	16
18. Оформление оборудования на баланс	3	17
19. Приобретение оборудования завершено	0	18
20. Монтаж локальной сети		
21. Установка компьютеров на рабочих местах	3	19
22. Монтаж кабеля	10	19
23. Монтаж сетевых устройств	10	19
24. Подключение кабеля к компьютерам и сетевым устройствам	5	21; 22; 23
25. Монтаж завершен	0	24
26. Установка ПО на компьютеры		
27. Установка сервера	5	25
28. Создание доменов и пользователей	7	27
29. Проверка и настройка работы сети	5	28
30. Настройка сети завершена	0	29
31. Обучение персонала		
32. Принципы работы системы	3	30
33. Изучение интерфейса	5	32
34. Изучение справочников	20	33
35. Изучение документов и журналов	30	34
36. Обучение завершено	0	35
37. Передача в эксплуатацию		
38. Формирование тестовой отчетности	5	36
39. Акт ввода в эксплуатацию	3	38
40. Передача в эксплуатацию завершена	0	39
41. Конец проекта	0	40

Между задачами 10 и 11 установить задержку в 5 дней, необходимую для прохождения безналичной оплаты. Между задачами 17 и 18 установить задержку в 7 дней, необходимую для прохождения безналичной оплаты и доставки оборудования.

Вариант 3. Создать проект «Ремонт квартиры», предназначенный для проведения ремонта в двухкомнатной квартире. Дата начала проекта – 1 февраля 2021 г. Перечень задач проекта, их связи и длительности приведены в таблице.

Название задачи	Длительность (дней)	Предшественники
1	2	3
1. Начало проекта	0	
2. Выравнивание стен		
3. Стены в спальне	4	1
4. Стены в гостиной	4	3
5. Стены в кухне	3	4
6. Стены в прихожей	4	5
7. Выравнивание стен завершено	0	6
8. Санузел		
9. Снятие штукатурки в санузле	3	1
10. Отделка стен санузла	4	9
11. Отделка потолка санузла	2	10
12. Отделка пола санузла	2	11
13. Установка сантехнического оборудования	1	12
14. Ремонт санузла завершен	0	13
15. Ванная		
16. Снятие штукатурки в ванной	3	1
17. Отделка стен ванной	5	16
18. Отделка потолка ванной	2	17
19. Отделка пола ванной	2	18
20. Установка сантехники	1	19
21. Ремонт ванной завершен	0	20
22. Отделка стен		
23. Отделка стен в спальне	5	7; 14; 21
24. Отделка стен в гостиной	7	23
25. Отделка стен в кухне	5	24
26. Отделка стен в прихожей		25
27. Отделка стен завершена	0	26
28. Потолки		
29. Замер	2	7
30. Заказ и оплата потолков	2	29
31. Навесной потолок в спальне	2	23; 30
32. Навесной потолок в гостиной	2	24; 30
33. Панельный потолок в кухне	2	25; 30
34. Навесной потолок в прихожей	2	26; 30
35. Монтаж потолков завершен	0	31; 32; 33; 34

36. Полы		
37. Отделка полов в спальне	6	31
38. Отделка полов в гостиной	6	32
39. Отделка полов в кухне	3	33
40. Отделка полов в прихожей	5	34
41. Отделка полов завершена	0	37; 38; 39; 40
42. Конец проекта	0	41

Установить задержки между задачами в соответствии с таблицей

Предшественник	Последователь	Величина задержки
30	31	10
30	32	10
30	34	10

ПРАКТИЧЕСКАЯ РАБОТА № 16

Тема: Выполнение ресурсного и бюджетного планирования в программе MS Project

Цель работы: научить студентов использовать MS Project для составления бюджета проекта, определять ресурсы проекта.

Оборудование: ПК, интернет, программное обеспечение – операционная система MS Project, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Планирование ресурсов и создание назначений. Выравнивание ресурсов

1. Создайте список ресурсов, задействованных при выполнении проекта на основе соответствующей варианту таблицы.
2. Выполните назначения ресурсов в соответствии с таблицей варианта.

Вариант 1. Для проекта «Строительство дома» создать список ресурсов в соответствии с параметрами, перечисленными в таблице

Название	Тип ресурса	Таблица норм	Стандартная ставка	Ставка сверх-урочных	Затраты на использование, руб.
1	2	3	4	5	6
Архитектор	Т	А	–	–	550 000
МУП «Горгаз»	Т	А	–	–	700 000
МУП «Водоканал»	Т	А	–	–	500 000
АО «Водолей2	Т	А	–	–	500 000
Рабочий1	Т	А	10 000 руб./дней	–	–
Рабочий2	Т	А	10 000 руб./дней	–	–
Рабочий3	Т	А	10 000 руб./дней	–	–
Подсобник1	Т	А	4 000 руб./дней	–	–
Подсобник2	Т	А	4 000 руб./дней	–	–
Трактор	Т	А	–	–	70 000
Плотник1	Т	А	150 000 руб./дней	2 000 руб./ч	–
		В	–		75 000
Плотник2	Т	А	1 5000 руб./дней	2 000 руб./ч	–
		В	–		75 000
АО «Неопласт»	Т	А	–	–	1 200 000
Водопроводчик1	Т	А	8 000 руб./дней	–	–
Водопроводчик2	Т	А	8 000 руб./дней	–	–
Электрик	Т	А	10 000 руб./дней	–	–
АО «Газовик»	Т	А	–	–	250 000
ООО «Потолки»	Т	А	–	–	1 500 000
Песок	М	А	5 000 руб./т	–	–
Щебень	М	А	6 000 руб./т	–	–
Цемент	М	А	780 000 руб./т	–	250 000
Кирпич	М	А	70 руб./шт.	–	–
Брус	М	А	–	–	250 000

Доска обрезная	М	А	70 000 руб./м ³	–	–
Доска необрезная	М	А	50 000 руб./м ³	–	–
Шифер	М	А	–	–	400 000
Электропровод	М	А	–	–	150 000
Электросчетчик	М	А	–	–	50 000
Труба водопроводная	М	А	–	–	350 000
Труба канализаци- онная	М	А	–	–	300 000
Штукатурка	М	А	–	–	1 500 000
Потолок	М	А	1 500 руб./м ²	–	–
Окно	М	А	100 000	–	–
Дверь наружная	М	А	–	–	200 000
Труба отопительная	М	А	–	–	200 000
Котел	М	А	–	–	400 000
Печь газовая	М	А	–	–	200 000
Ванна	М	А	450 000 руб.	–	–
Унитаз компакт	М	А	200 000 руб.	–	–
Раковина	М	А	160 000 руб.	–	–
Кран	М	А	70 000 руб.	–	–
Паркет	М	А	5 500 руб./м ²	–	–
Труба газовая	М	А	–	–	500 000
Дверь внутренняя	М	А	90 000 руб.	–	–
Доставка	З		–	–	500 000

Создать назначения ресурсов для данного проекта в соответствии с таблицей

Задача	Ресурс	Единицы (затраты)	Таблица норм затрат
1	2	3	4
Утверждение проекта на строи- тельство	Архитектор	100 %	А
Утверждение проекта на газ	МУП «Горгаз»	100%	А
Рытье траншей	Рабочий1	100%	А
	Рабочий2	100%	А
	Рабочий3	100%	А
	Подсобник1	100%	А
	Подсобник2	100%	А
	Трактор	100%	А
Утверждение проекта на водо- провод и канализацию	МУП «Водоканал»	100%	А
Утверждение проекта на ото- пление	АО «Водолей»	100%	А
Заливка фундамента	Рабочий1	100%	А
	Рабочий2	100%	А
	Рабочий3	100%	А
	Подсобник1	100%	А
	Подсобник2	100%	А
	Песок	10 т	А
	Щебень	10 т	А
	Цемент	2 500 кг	А
	Доска необрезная	3 м ³	А
	Доставка	250 000 руб.	

Кладка стен	Рабочий1	100%	А
	Рабочий2	100%	А
	Рабочий3	100%	А
	Подсобник1	100%	А
	Подсобник2	100%	А
	Кирпич	70 000 шт.	А
	Песок	6 т	А
	Цемент	2000 кг	А
	Доставка	250 000 руб.	
Перекрытие стен	Рабочий1	100%	А
	Рабочий2	100%	А
	Рабочий3	100%	А
	Подсобник1	100%	А
	Подсобник2	100%	А
	Брус	15 шт.	А
	Доска обрезная	7 м ³	А
	Доставка	150 000 руб.	
Установка крыши	Плотник1	100	А
	Плотник2	100	А
	Доска необрезная	10 м ³	А
	Шифер	20 лист	А
	Доставка	120 000 руб.	
Установка наружных дверей и окон	ООО «Неопласт»	100 %	А
	Окно	9 шт.	А
	Дверь наружная	1 шт.	А
Установка полов	Плотник1	100%	А
	Плотник2	100%	А
	Доска обрезная	10 м ³	А
	Доставка	70 000 руб.	
Проведение и подключение водопровода и канализации	Водопроводчик1	100%	А
	Водопроводчик2	100%	А
	Труба водопроводная	1 шт.	А
	Труба канализация	1 шт.	А
Установка и подключение электропроводки	Электрик	100%	А
	Электросчетчик	1 шт.	А
	Электропровод	100 м	А
Установка и подключение газовых коммуникаций	АО «Газовик»	100%	А
	Труба газовая	1 шт.	А
Отделка стен	Рабочий1	100%	А
	Рабочий2	100%	А
	Рабочий3	100%	А
	Подсобник1	100%	А
	Подсобник2	100%	А
	Штукатурка	100 кг	А
Навесные потолки	ООО «Потолки»	100%	А
	Потолок	190 м ²	А
Внутренние двери	Плотник1	100%	В
	Плотник2	100%	В
	Дверь внутренняя	10 шт.	А
	Доставка	100 000 руб.	
Монтаж отопления	Водопроводчик1	100%	А
	Водопроводчик2	100%	А
	Труба отопительная	1 шт.	А

Установка оборудования, приборов и сантехники	Водопроводчик1	100%	А
	Водопроводчик2	100%	А
	Котел	1 шт.	А
	Печь газовая	1 шт.	А
	Ванна	1 шт.	А
	Унитаз компактный	2 шт.	А
	Раковина	3 шт.	А
	Кран	4 шт.	А
Настил полов	Рабочий1	100%	А
	Рабочий2	100%	А
	Рабочий3	100%	А
	Подсобник1	100%	А
	Подсобник2	100%	А
	Паркет	190 м ²	А

Установить профили загрузки ресурсов: МУП «Горгаз» – затраты в конце, МУП «Водоканал» – поздний пик, АО «Водолей» – колокол.

Вариант 2. Для проекта «Внедрение бухгалтерской системы» создать список ресурсов в соответствии с параметрами, перечисленными в таблице.

Название	Тип ресурса	Таблица норм	Стандартная ставка	Ставка сверхурочная, руб./ч	Затраты на использование, руб.
1	2	3	4	5	6
Главбух	Т	АВ	900 000 руб./мес	5 000	300 000
Администратор	Т	АВ	700 000 руб./мес	4 500	400 000
Программист	Т	АВ	600 000 руб./мес	4 000	500 000
Техник	Т	А	400 000 руб./мес	2 500	–
Бухгалтер мат. учета1	Т	А	400 000 руб./мес	2 500	–
Бухгалтер мат. учета2	Т	А	400 000 руб./мес	2 500	–
Бухгалтер учета ОС и НМА	Т	А	400 000 руб./мес	2 500	–
Бухгалтер учета ОС	Т	А	400 000 руб./мес	2 500	–
Бухгалтер учета реализации	Т	А	400 000 руб./мес	2 500	–
Бухгалтер производственного учета	Т	А	400 000 руб./мес	2 500	–
Компьютер	М	А	1500 000 руб./шт.	–	–
Сервер	М	А	1500 000 руб./шт.	–	–
Принтер	М	А	150 000 руб./шт.	–	–
МФУ	М	А	170 000 руб./шт.	–	–
Сетевой кабель	М	А	–	–	150 000
Сетевой концентратор	М	А	30 000 руб./шт.	–	–
Панель	М	А	–	–	100 000
Разъемы и розетки	М	А	–	–	150 000
Бухгалтерская система	М	А	–	–	1 000 000
Офисный пакет	М	А	–	–	700 000
ОС рабочей станции	М	А	–	–	600 000
Серверная ОС	М	А	–	–	300 000
Интернет	З	–	2000 руб./ч	–	–
Междугородние переговоры	З	–	5000 руб./ч	–	100 000

Создать назначения ресурсов в соответствии с таблицей.

Задача	Ресурс	Единицы (затраты)	Таблица норм затрат
1	2	3	4
Изучение рынка бухгалтерских систем	Администратор Интернет	100%	А 15 000 руб.
Составление требований к бухгалтерским системам	Администратор Главбух	100% 20%	А А
Консультации с фирмами-разработчиками	Администратор Междугородние переговоры Интернет	100%	А 20 000 руб. 10 000 руб.
Принятие окончательного решения	Администратор Главбух	100% 100%	А А
Заключение договоров	Администратор Программист Главбух	100% 100% 100%	А А А
Оплата за ПО	Главбух Бухгалтерская система Офисный пакет ОС рабочей станции Серверная ОС	10% 1% 1% 1% 1%	А А А А А
Оформление ПО на баланс	Бухгалтер учета ОС и НМА	30%	А
Сбор информации о поставщиках и предложениях	Администратор Интернет Междугородние переговоры	50%	А 10 000 руб. 15 000 руб.
Анализ и выбор поставщика	Администратор Главбух Интернет	50% 20%	А А 10 000 руб.
Заключение договоров	Администратор Главбух	100% 50%	А А
Оплата за оборудование	Главбух Компьютер Сервер Принтер МФУ Сетевой кабель Сетевой концентратор Панель Разъемы и розетки	30% 12 шт. 1 шт. 2 шт. 2 шт. 20 м 1 шт. 5 шт. 10 шт.	А А А А А А А А А
Оформление оборудования на баланс	Бухгалтер учета ОС	70%	А
Установка компьютеров на рабочих местах	Техник	100%	А
Монтаж кабеля	Техник	100%	А
Монтаж сетевых устройств	Техник	100%	А

Подключение кабеля к компьютерам и сетевым устройствам	Техник	100%	A
Установка сервера	Администратор	100%	A
Создание доменов и пользователей	Администратор	100%	A
Проверка и настройка работы сети	Администратор	100%	A
	Программист	100%	A
Принципы работы системы	Администратор	50%	A
	Главбух	50%	A
	Бухгалтер мат. учета1	50%	A
	Бухгалтер мат. учета2	50%	A
	Бухгалтер учета ОС и НМА	50%	A
	Бухгалтер учета ОС	50%	A
	Бухгалтер учета реализации	50%	A
	Бухгалтер производств. учета	50%	A
Изучение интерфейса	Программист	50%	A
	Главбух	50%	A
	Бухгалтер мат. учета1	50%	A
	Бухгалтер мат. учета2	50%	A
	Бухгалтер учета ОС и НМА	50%	A
	Бухгалтер учета ОС	50%	A
	Бухгалтер учета реализации	50%	A
	Бухгалтер производств. учета	50%	A
Изучение справочников	Программист	50%	A
	Главбух	50%	A
	Бухгалтер мат. учета1	50%	A
	Бухгалтер мат. учета2	50%	A
	Бухгалтер учета ОС и НМА	50%	A
	Бухгалтер учета ОС	50%	A
	Бухгалтер учета реализации	50%	A
	Бухгалтер производств. учета	50%	A
Изучение документов и журналов	Программист	50%	A
	Главбух	50%	A
	Бухгалтер мат. учета1	50%	A
	Бухгалтер мат. учета2	50%	A
	Бухгалтер учета ОС и НМА	50%	A
	Бухгалтер учета ОС	50%	A
	Бухгалтер учета реализации	50%	A
	Бухгалтер производств. учета	50%	A
Формирование тестовой отчетности	Администратор	100%	A
	Программист	100%	A
	Главбух	100%	A
Акт ввода в эксплуатацию	Администратор	50%	A
	Главбух	50%	A

Установить различные профили загрузки для ресурса Техник.

Вариант 3. Для проекта «Ремонт квартиры» создать список ресурсов в соответствии с параметрами, перечисленными в таблице.

Название	Тип ресурса	Таблица норм	Стандартная ставка	Ставка сверхурочных, руб./ч	Затраты на использование, руб.
Слесарь-водопроводчик	Т	А В	10 000 руб./дней	1 500	20 000
Штукатур	Т	А	8 000 руб./дней	1000	–
Подсобник	Т	А	4 000 руб./дней	500	–
Плиточник	Т	А	15 000 руб./дней	2 000	–
Плотник	Т	А	15 000 руб./дней	2 000	–
АО «Светлый потолок»	Т	А	10 000 руб./дней	1 500	–
Унитаз-компакт	М	А	150 000 руб./шт.	–	–
Ванна	М	А	350 000 руб./шт.	–	–
Раковина	М	А	250 000 руб./шт.	–	–
Смеситель с душем	М	А	100 000 руб./шт.	–	–
Плитка	М	А	10 000 руб./ м ²	–	–
Панель	М	А	5 000 руб./шт.	–	–
Обои	М	А	15 000 руб./рул.	–	–
Навесной потолок	М	А	–	–	700 000
Паркет	М	А	15 000 руб./ м ²	–	–
Доставка	З		–	–	500 000

Создать назначения ресурсов в соответствии с таблицей.

Задача	Ресурс	Единицы (затраты)	Таблица норм затрат
1	2	3	4
Стены:			
в спальне	Штукатур	100%	А
гостиной	Штукатур	100%	А
кухне	Штукатур	100%	А
прихожей	Штукатур	100%	А
Снятие штукатурки в санузле	Подсобник	100%	А
Отделка санузла:			
стен	Плиточник	100%	А
	Плитка	10 м ²	А
потолка	Плиточник	100%	А
	Панель	5 шт.	А
пола	Плиточник	100%	А
	Плитка	5 м ²	А
Установка сантехнического оборудования	Слесарь-водопроводчик	100%	В
	Унитаз-компакт	1 шт.	А
Снятие штукатурки в ванной	Подсобник	100%	А
Отделка ванной:			
стен	Плиточник	100%	А
	Плитка	10 м ²	А
потолка	Плиточник	100%	А
	Панель	6 шт.	А
пола	Плиточник	100%	А
	Плитка	6 м ²	А

Установка сантехники	Слесарь-водопроводчик	100%	В
	Ванна	1 шт.	А
	Раковина	1 шт.	А
	Смеситель с душем	1 шт.	А
Отделка стен: в спальне	Штукатур	100%	А
	Обои	8 рул.	А
гостиной	Штукатур	100%	А
	Обои	8 рул.	А
кухне	Штукатур	100%	А
	Плиточник	100%	А
	Плитка	5 м ²	А
	Панель	10 шт.	А
прихожей	Штукатур	100%	А
	Плиточник	100%	А
	Панель	15 шт.	А
Замер	АО «Светлый потолок»	100%	А
Заказ и оплата потолков	Навесной потолок	1 шт.	А
Навесной потолок: в спальне	«Светлый потолок»	100%	А
гостиной	«Светлый потолок»	100%	А
кухне	Плиточник	100%	А
	Панель	6 шт.	А
прихожей	«Светлый потолок»	100%	А
Отделка полов в: спальне	Плотник	100%	А
	Паркет	20 м ²	А
гостиной	Плотник	100%	А
	Паркет	20 м ²	А
кухне	Плотник	100%	А
	Паркет	10 м ²	А
прихожей	Плотник	100%	А
	Паркет	15 м ²	А

Установить различные профили загрузки для ресурса Подсобник. Для созданных проектов определить ресурсы, которые перегружены, установить причины перегрузки и выполнить выравнивание загрузки ресурсов следующими способами.

3. Автоматическое выравнивание (результат сохранить в файл Автоматическое_выравнивание.mpr).

4. Замена одного ресурса другим (результат сохранить в файл Замена_ресурсов.mpr).

5. Редактирование распределения трудозатрат вручную (результат сохранить в файл Ручное_выравнивание.mpr).

6. Перенос части трудозатрат в сверхурочные (результат сохранить в файл Перенос_сверхурочные.mpr). При выравнивании можно комбинировать эти способы.

ПРАКТИЧЕСКАЯ РАБОТА № 17

Тема: Выполнение ресурсного и бюджетного планирования в программе MS Project

Цель работы: научить студентов использовать MS Project для составления бюджета проекта, определять ресурсы проекта.

Оборудование: ПК, интернет, программное обеспечение – операционная система MS Project, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Анализ проектов

Для созданных в Практических работа № 15 и 16 проектов выполнить следующие виды анализа

1. Параметрический анализ длительностей задач. Выполните параметрический анализ длительностей задач. Назначить как параметрическую следующую задачу:

- для варианта 1. Задача «Отделка стен». Норма длительности обработки одной стены – 0,4 дня, количество стен – 6;
- для варианта 2. Задача «Установка компьютера на рабочих местах». Норма длительности – 2 ч, количество компьютеров 40;
- для варианта 3. Задача «Отделка полов». Норма длительности обработки 1 м² – 0,3 дня

Создать настраиваемые поля для параметра, нормы и оценки длительности; ввести значения параметра и нормы длительности и вычислить оценку длительности. Результаты анализа сохранить в файле ПараметрическийАнализ.mpp.

2. Выполните PERT-анализ длительностей задач. Задайте в таблице ввода PERT значения оптимистической, ожидаемой и пессимистической длительностей согласно таблицам по вариантам. Выполнить расчет длительностей по методу PERT. Результаты сохраните в файл PERTАнализ.mpp.

Таблица оценок длительности для варианта 1

Задача	Оптимистическая длительность	Ожидаемая длительность	Пессимистическая длительность
Утверждение проектов	85	90	100
Строительство фунда-мента	14	15	18
Каркас и крыша	100	105	110
Коммуникации	9	10	15
Внутренняя отделка	40	43	45

Таблица оценок длительности для варианта 2

Задача	Оптимистическая длительность	Ожидаемая длительность	Пессимистическая длительность
Выбор системы	8	9	10
Оплата за ПО	10	11	12
Монтаж локальной сети	14	15	17
Обучение персонала	55	58	60
Передача в эксплуатацию	7	8	10

Таблица оценок длительности для варианта 3

Задача	Оптимистическая длительность	Ожидаемая длительность	Пессимистическая длительность
Выравнивание стен	14	15	17
Санузел	10	12	15
Ванная	12	13	15
Отделка стен	21	22	25
Потолки	11	12	15
Полы	18	20	22

3. Выполните анализ критического пути. Результат сохраните в файл АнализКрит.mpp.
4. Выполните анализ стоимости проекта. Результат сохранить в файл АнализСтоим.mpp.
5. Выполните анализ стоимости ресурсов разного вида. Результат сохранить в файл РесурсыРазногоВида.mpp.
6. Выполните анализ сверхурочных затрат. Результат сохранить в файл Сверхур.mpp.
7. Выполните анализ рисков задач со слишком короткой длительностью. Результат сохранить в файл Короткие.mpp.
8. Выполните анализ рисков слишком длинных задач с большим количеством ресурсов. Результат сохранить в файл Длинные.mpp.

ПРАКТИЧЕСКАЯ РАБОТА № 18

Тема: Анализ и планирование рисков

Цель работы: изучение базовых принципов построения и приобретение навыков использования модели угроз и уязвимостей ИС при анализе рисков информационной безопасности в корпоративных информационных системах

Оборудование: ПК, интернет, программное обеспечение – MS Project, MS Word, инструкции по выполнению работы.

Справочный материал:

Обеспечение безопасности информации носит комплексный характер и основано, среди прочего, на анализе возможных негативных последствий развертывания и эксплуатации ИС.

Модель угроз и уязвимостей предназначена для оценки защищенности ИС, при которой моделирование нарушений безопасности осуществляется на базе анализа цепочки "уязвимость угроза-ресурс". При этом необходимо убедиться, что все ресурсы и уязвимости идентифицированы и сопоставлены с угрозами. Важно иметь возможность при необходимости, не меняя самого методического инструментария, вводить новые виды угроз и уязвимостей, которые станут известны вследствие развития знаний в этой области. В результате анализа по модели угроз и уязвимостей вычисляется уровень риска нарушения безопасности ИС и определяются его возможные причины.

Понятийный базис модели:

- угроза – действие, которое потенциально может привести к нарушению безопасности. Свойством угрозы является перечень уязвимостей, при помощи которых она может быть реализована;

- уязвимость - "слабое" место в ИС, которое может привести к нарушению безопасности путем реализации некоторой угрозы. Свойствами уязвимости являются вероятность (простота) и критичность реализации угрозы через данную уязвимость;

- вероятность реализации угрозы через данную уязвимость в течение года – степень возможности реализации угрозы через данную уязвимость в тех или иных условиях. Измеряется в процентах;

- критичность реализации угрозы – степень влияния реализации угрозы на ресурс, т. е. как сильно реализация угрозы повлияет на работу ресурса. Задается в процентах. Допустимо рассматривать критичность реализации угрозы и по составляющим безопасности (конфиденциальности, целостности и доступности), и в целом (с учетом всех трех рисков);

- ресурс – часть ИС, предназначенная для хранения информации, подверженной угрозам ИБ (сервер, рабочая станция, мобильный компьютер,

- устройство хранения резервных копий и т. д.). Свойствами ресурса в данной модели являются перечень угроз, воздействующих на него, и критичность ресурса;

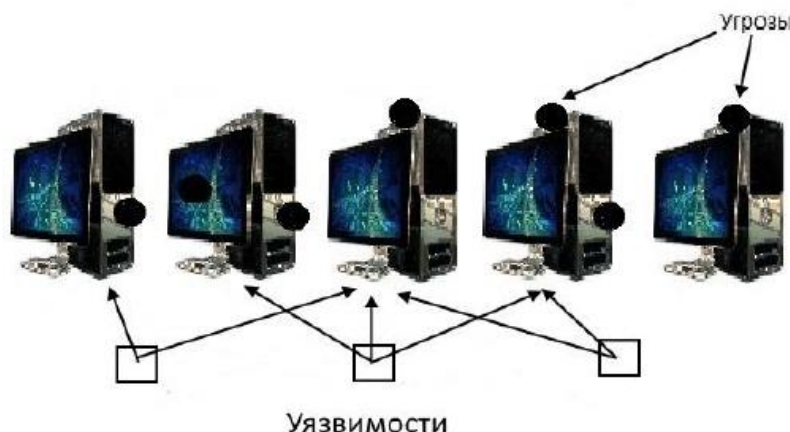
- критичность ресурса – степень значимости ресурса для ИС, т. е. насколько сильно реализация угроз безопасности на ресурс повлияет на работу всей ИС. Критичность ресурса оценивают в денежных единицах,

относительных уровнях, условных единицах. Чаще всего она оценивается в процентах, т. е. в уровнях от 1 до 100 по отношению к критичности других ресурсов.

– максимальное критичное время простоя – значение времени простоя, которое является критичным для компании. Ущерб, нанесенный компании при простом ресурсе в течение этого времени, достигает максимального значения. При дальнейшем простое ущерб, наносимый компании, уже не увеличивается. Эта характеристика используется для оценки критичности ресурса по угрозе доступности.

Критичность ресурса оценивается, как правило, из ущерба, который понесет компания при осуществлении угроз нарушения конфиденциальности, доступности или целостности. При рассмотрении этой характеристики необходимо определить шкалу, используемую для классификации критичности.

Для каждого из ресурсов определяются список угроз, действующих на него, и список уязвимостей. Для нарушения безопасности по одной угрозе нарушитель может воспользоваться разными уязвимостями. Одна и та же уязвимость может привести к нарушению безопасности по нескольким угрозам.



Содержание работы:

1. Получить описание ИС.
2. Для данной ИС построить модель угроз и уязвимостей:
 - выделить угрозы, применимые к рассматриваемой ИС;
 - выделить уязвимости, через которые могут быть реализованы угрозы;
 - определить угрозы, которые могут воздействовать на каждый из ресурсов в рамках ИС, и обосновать причины наличия этих угроз;
 - определить уязвимости, через которые могут быть реализованы указанные угрозы.
3. Определить вероятности и критичности реализации угроз через уязвимости для каждой пары "угроза-уязвимость".
4. Определить функции для расчета рисков.
5. Рассчитать риски для всех ресурсов в рассматриваемой модели ИС.
6. Провести анализ полученных результатов. Выделить наиболее опасные уязвимости и предложить способы снижения вероятности и критичности.

Предложить дальнейший план развития политики информационной безопасности для рассматриваемой ИС.

Описание информационной системы. Тестовая информационная система ЗАО "ТестИС-Строй".

Основной вид деятельности ЗАО "ТестИС-Строй" – продажа строительных товаров на рынке "BusinessstoClient". Поставщиками являются частные лица и организации среднего и малого бизнеса. ЗАО "ТестИС-Строй" имеет четыре точки продаж, расположенные в пределах города. Каждая из этих точек – магазин площадью от 300 до 2000 м². В каждом магазине работает до 100 сотрудников.

ЗАО "ТестИС-Строй" имеет центральный офис в центре города, где располагается дата-центр, включающий центральную базу данных товаров и серверы баз данных бухгалтерии, отдела кадров и т. д. В центральном офисе и на каждой из точек продаж развернуты локальные вычислительные сети (ЛВС). Каждая из ЛВС точек продаж связана с центральным офисом посредством сети Интернет. В точках продаж функционируют 1-2 сервера, обеспечивающих синхронизацию с центральной базой данных, и до 20 рабочих станций: компьютеры директора магазина, секретаря, терминалы в торговых залах.

В дата-центре установлены Web-сайт электронного магазина и почтовый сервер. К терминалам торговых залов исключена возможность подключения внешних носителей. В дата-центре все серверы размещены в несгораемых сейфах, доступ в помещение контролируется физически (охраняемое помещение). В торговых точках все серверы находятся в кабинетах, закрываемых на ключ. На всех компьютерах, кроме терминалов в торговых залах, установлено антивирусное ПО.

На серверах дата-центра установлен межсетевой экран. На сервере базы данных бухгалтерии дополнительно установлена система обнаружения вторжений. Для подключения к дата-центру используется защищенное VPN-соединение. Для подключения к центральной базе товаров предусмотрен резервный канал. Загрузка терминалов торговых залов обеспечивается только после введения пароля в BIOS.

ПРАКТИЧЕСКАЯ РАБОТА № 19

Тема: Анализ и планирование рисков

Цель работы: изучение базовых принципов построения и приобретение навыков использования модели угроз и уязвимостей ИС при анализе рисков информационной безопасности в корпоративных информационных системах

Оборудование: ПК, интернет, программное обеспечение – MS Project, MS Word, инструкции по выполнению работы.

Содержание работы:

1. Получить описание ИС.
2. Для данной ИС построить модель угроз и уязвимостей:
 - выделить угрозы, применимые к рассматриваемой ИС;
 - выделить уязвимости, через которые могут быть реализованы угрозы;
 - определить угрозы, которые могут воздействовать на каждый из ресурсов в рамках ИС, и обосновать причины наличия этих угроз;
 - определить уязвимости, через которые могут быть реализованы указанные угрозы.
3. Определить вероятности и критичности реализации угроз через уязвимости для каждой пары "угроза-уязвимость".
4. Определить функции для расчета рисков.
5. Рассчитать риски для всех ресурсов в рассматриваемой модели ИС.
6. Провести анализ полученных результатов. Выделить наиболее опасные уязвимости и предложить способы снижения вероятности и критичности. Предложить дальнейший план развития политики информационной безопасности для рассматриваемой ИС.

Описание информационной системы. Тестовая информационная система издательства газеты "ТестИС-Пресс". Редакция газеты "ТестИС-Пресс" занимается публикацией новостей из мира информационных технологий.

Читатели и конкуренты не должны иметь возможности узнать о публикуемых новостях ранее выпуска номера (факторы актуальности и эксклюзивности).

Издательство "ТестИС-Пресс" включает подразделения: руководство (директор и заместитель, главный редактор), IT-отдел (администраторы), бухгалтерия (главный бухгалтер и бухгалтеры), журналисты, редакторы, наборщики, верстальщики. Типография также входит в состав издательства (сотрудниками являются инженеры по печати и переплету). Всего в редакции работают 60 сотрудников.

В бухгалтерии используются два сервера: для хранения бухгалтерской базы данных и ее резервных копий. Каждый редактор, журналист, наборщик и верстальщик работает на своей рабочей станции. В издательстве используется несколько серверов: для хранения материалов готовящегося к выходу номера, хранения архивов номеров. Почта и web-сайт издательства функционируют на двух выделенных серверах. Доступ в Интернет осуществляется через Провайдера. Издательство готово к сотрудничеству с внешними аудиторами.

В издательстве используется система криптозащиты электронной почты. На рабочих станциях редакторов и журналистов настроена система

автоматической блокировки станции при отсутствии сотрудника на рабочем месте. В бухгалтерии установлена система видеонаблюдения. Доступ в серверную комнату обеспечивается только по пропускам. Предусмотрено резервное копирование бухгалтерской базы данных.

ПРАКТИЧЕСКАЯ РАБОТА № 20

Тема: Анализ и планирование рисков

Цель работы: изучение базовых принципов построения и приобретение навыков использования модели угроз и уязвимостей ИС при анализе рисков информационной безопасности в корпоративных информационных системах

Оборудование: ПК, интернет, программное обеспечение – MS Project, MS Word, инструкции по выполнению работы.

Содержание работы:

1. Получить описание ИС.
2. Для данной ИС построить модель угроз и уязвимостей:
 - выделить угрозы, применимые к рассматриваемой ИС;
 - выделить уязвимости, через которые могут быть реализованы угрозы;
 - определить угрозы, которые могут воздействовать на каждый из ресурсов в рамках ИС, и обосновать причины наличия этих угроз;
 - определить уязвимости, через которые могут быть реализованы указанные угрозы.
3. Определить вероятности и критичности реализации угроз через уязвимости для каждой пары "угроза-уязвимость".
4. Определить функции для расчета рисков.
5. Рассчитать риски для всех ресурсов в рассматриваемой модели ИС.
6. Провести анализ полученных результатов. Выделить наиболее опасные уязвимости и предложить способы снижения вероятности и критичности. Предложить дальнейший план развития политики информационной безопасности для рассматриваемой ИС.

Описание информационной системы. Тестовая информационная система компании по обслуживанию средств электронной коммерции "ТестИС-Е".

Компания "ТестИС-Е" осуществляет деятельность в рамках продаж и обслуживания оборудования электронной коммерции. В компании существуют три отдела: бухгалтерия (главный бухгалтер, бухгалтеры), отдел технического анализа (руководитель отдела, инженеры), отдел продаж (руководитель отдела, заместитель, менеджеры). Для всех сотрудников предоставлена рабочая станция с выходом в Интернет. В конференц-зале установлена рабочая станция (ноутбук), которую используют только члены совета директоров (финансовый директор, генеральный директор и его заместитель).

В компании определена должность администратора, имеющего доступ ко всем ресурсам системы за исключением сервера коммерческих данных.

В отделе технического анализа расположен сервер-хранилище информации. Доступ к нему имеют только сотрудники этого отдела. Почтовым сервером, расположенным в серверном помещении, пользуются сотрудники отдела продаж, дирекция и бухгалтерия. К серверу коммерческих данных, расположенному в серверном помещении, имеют доступ только члены совета директоров и главный бухгалтер (администратор не имеет доступа к этому серверу). Резервное копирование этого сервера выполняется на CD-R-носители, хранимые в сейфе генерального директора.

Для доступа в Интернет используется шлюз в виде отдельного сервера, размещенного в серверном помещении. На нем установлена служба VPN-доступа, Web-прокси, служба фильтрации запросов, система учета трафика и система обнаружения вторжений. В компании не используется антивирусное программное обеспечение.

Внутренние документы и настройки безопасности рабочих станций запрещают запускать любые программы, кроме, установленных на компьютерах. Запрещена инициация соединений с рабочими станциями пользователей внешней сети. На почтовом сервере используется система антивирусной защиты электронной почты и защиты от спама.

ПРАКТИЧЕСКАЯ РАБОТА № 21

Тема: Анализ и планирование рисков

Цель работы: изучение базовых принципов построения и приобретение навыков использования модели угроз и уязвимостей ИС при анализе рисков информационной безопасности в корпоративных информационных системах

Оборудование: ПК, интернет, программное обеспечение – MS Project, MS Word, инструкции по выполнению работы.

Содержание работы:

1. Получить описание ИС.
2. Для данной ИС построить модель угроз и уязвимостей:
 - выделить угрозы, применимые к рассматриваемой ИС;
 - выделить уязвимости, через которые могут быть реализованы угрозы;
 - определить угрозы, которые могут воздействовать на каждый из ресурсов в рамках ИС, и обосновать причины наличия этих угроз;
 - определить уязвимости, через которые могут быть реализованы указанные угрозы.
3. Определить вероятности и критичности реализации угроз через уязвимости для каждой пары "угроза-уязвимость".
4. Определить функции для расчета рисков.
5. Рассчитать риски для всех ресурсов в рассматриваемой модели ИС.
6. Провести анализ полученных результатов. Выделить наиболее опасные уязвимости и предложить способы снижения вероятности и критичности. Предложить дальнейший план развития политики информационной безопасности для рассматриваемой ИС.

Описание информационной системы. Тестовая информационная система компании по разработке программного обеспечения "ТестИС-Солюшн".

Компания "ТестИС-Солюшн" занимается разработкой программного обеспечения, используемого в банках. Руководство компании "ТестИС-Солюшн" представлено генеральным директором, его заместителем и техническим директором. Штатный состав компании составляет 70 человек.

Разработки компании имеют закрытый тип. Для их контроля и контроля используемых ресурсов в штатном расписании предусмотрены следующие должности: главный администратор, администратор файловых серверов, сетевой администратор. Остальные сотрудники – инженеры по разработке ПО, тестировщики, разработчики и руководители проектов. В бухгалтерии работает один человек – главный бухгалтер. База данных бухгалтерии находится на его рабочей станции.

Информация, обрабатываемая в системе – результаты и данные разработки программных продуктов: исходные коды, документация, результаты тестирования.

В компании используется антивирусная защита, все серверы расположены в серверном помещении, закрываемом на ключ. На серверах установлены системы контроля версий; почтовый и файловый серверы защищены межсетевым экраном. Сервер системы контроля версий доступен тестировщикам из Интернет по VPN-соединению. Компьютер главного

бухгалтера не имеет дисководов и устройств подключения USB, кроме того, он не включен в основную сеть. Резервное копирование не производится.

Дирекция и руководители проектов имеют доступ в Интернет с рабочих мест. Только разработчики могут вносить изменения в систему контроля версий.

ПРАКТИЧЕСКАЯ РАБОТА № 22

Тема: Выявление первичных и вторичных ошибок

Цель работы: изучить взаимосвязь между причинами и последствиями ошибок, а также разработать подходы для минимизации проблем, изучить первичные и вторичные ошибки, научиться применять методы и способы выявления ошибок, оформлять отчет о дефектах программного продукта.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для тестирования и отладки, MS Word, инструкции по выполнению работы.

Справочный материал:

Под *ошибкой* в широком смысле слова понимается неправильность, погрешность или неумышленное, невольное искажение объекта или процесса. При этом подразумевается, что известно правильное или неискаженное эталонное состояние объекта, к которому относится ошибка. Считается, что если программа не выполняет того, что пользователь от нее ожидает, то в ней имеется ошибка.

Важной особенностью процесса выявления ошибок в сложных программах является отсутствие полностью определенной правильной программы-эталона, которому должен соответствовать проверяемый текст. Поэтому нельзя гарантированно утверждать, что возможно написать программу без ошибок.



Искажения в тексте программ (первичные ошибки) являются элементами, подлежащими корректировке. Однако непосредственно наличие ошибки обнаруживаются по ее вторичным проявлениям.

Искажение выходных результатов исполнения программ (вторичная ошибка) вызывает необходимость выполнения ряда операций по локализации устранению первичной ошибки (отладка программ).

На этапе отладки программ выявляются и исправляются много ошибок, но не все.

После отладки в течение некоторого времени интенсивность обнаружения ошибок при самом активном тестировании снижается настолько, что разработчики попадают в зону нечувствительности к программным ошибкам и отказам.

При такой интенсивности отказов программ трудно прогнозировать затраты времени, необходимые для обнаружения очередной ошибки.

Создается представление о полном отсутствии ошибок в программе, о невозможности и бесцельности их поиска. Поэтому усилия на отладку сокращаются, и интенсивность обнаружения ошибок еще больше снижается. Этой предельно низкой интенсивности обнаружения отказов соответствует наработка на обнаруженную ошибку, при которой прекращается улучшение характеристик программного обеспечения на этапах его отладки и испытаний у заказчика.

Ошибку можно отнести к одному из ниже перечисленных классов:

- системные ошибки;
- ошибки в выборе алгоритма;
- алгоритмические ошибки;
- технологические ошибки;
- программные ошибки.

Системные ошибки в большом (сложном) программном обеспечении определяются, прежде всего неполной информацией о реальных процессах, происходящих в источниках и потребителях информации. На начальных стадиях проектирования программного обеспечения не всегда удается точно сформулировать целевую задачу всей системы и требования к ней. В процессе проектирования целевая функция системы уточняется и выявляются отклонения от уточненных требований, которые могут квалифицироваться как системные ошибки.

Некачественное определение требований к программе приводит к созданию программы, которая будет правильно решать неверно сформулированную задачу. В таких случаях, как правило, требуется полное перепрограммирование. Признаком того, что создаваемая для заказчика программа может оказаться не соответствующей его истинным потребностям, служит ощущение неясности задачи. Письменная регистрация требований к программе заставляет заказчика собраться с мыслями и дать достаточно точное определение требований. Всякие устные указания являются заведомо ненадежными и часто приводят к взаимному недопониманию.

При автономной и в начале комплексной отладки программного обеспечения доля найденных системных ошибок в нем невелика (примерно 10%), но она существенно возрастает (до 35—40%) на завершающих этапах комплексной отладки. В процессе эксплуатации преобладающими являются системные ошибки (примерно 80% всех ошибок).

Ошибки в выборе алгоритма. Часто плохой выбор алгоритма становится очевидным лишь после его опробования. Поэтому все же следует уделять внимание и время выбору алгоритма, с тем, чтобы впоследствии не приходилось переделывать каждую программу. Во избежание выбора некорректных алгоритмов, необходимо хорошо ознакомиться с литературой по своей специальности.

К алгоритмическим ошибкам следует отнести, прежде всего, ошибки, обусловленные некорректной постановкой функциональных задач, когда в спецификациях не полностью оговорены все условия, необходимые для

получения правильного результата. Эти условия формируются и уточняются в значительной части в процессе тестирования и выявления ошибок в результатах функционирования программ.

Также следует отнести ошибки связей модулей и функциональных групп программ. Их можно квалифицировать как ошибки некорректной постановки задачи.

Алгоритмические ошибки проявляются в неполном учете диапазонов изменения переменных, в неправильной оценке точности используемых и получаемых величин, в неправильном учете связи между различными переменными, в неадекватном представлении формализованных условий решения задачи в спецификациях или схемах, подлежащих программированию и т.д.

Эти обстоятельства являются причиной того, что для исправления каждой алгоритмической ошибки приходится изменять иногда целые ветви программного обеспечения, т.е. пока еще существенно больше операторов, чем при исправлении программных ошибок.

Алгоритмические ошибки значительно труднее поддаются обнаружению методами формализованного автоматического контроля. Вот почему необходимо тщательным образом продумывать алгоритм прежде, чем транслировать его в программу.

Некоторые программисты проверяют алгоритм следующим образом. Через несколько дней после составления алгоритма они повторно обращаются к описанию задачи и составляют алгоритм заново. Затем сличают оба варианта. Такой шаг на первый взгляд может показаться пустой тратой времени, однако всякая ошибка на уровне алгоритма может в дальнейшем обернуться катастрофой и повлечь основательный пересмотр программы.

Технологические ошибки – это ошибки документации и фиксирования программ в памяти ЭВМ. Они составляют 5—10 % от общего числа ошибок, обнаруживаемых при отладке. Большинство технологических ошибок выявляются автоматически формализованными методами (например, транслятором).

Программные ошибки. Языки программирования – это искусственные языки, созданные человеком для описания алгоритмов. Все предложения таких языков строятся по строгим синтаксическим правилам, обеспечивающим однозначное их понимание, что позволяет поручать расшифровку алгоритма ЭВМ, построенного по правилам семантики.

Синтаксис – это набор правил построения из символов алфавита специальных конструкций, с помощью которых можно составлять различные алгоритмы (программы). Эти правила требуют их неукоснительного соблюдения. В противном случае будет нарушен основной принцип - четкая и строгая однозначность в понимании алгоритма.

Семантика языка — это система правил истолкования построений конструкций. Правила семантики конструкций обычно вполне естественны и

понятны, но в некоторых случаях их надо специально оговаривать, комментировать.

Таким образом, программы, позволяющие однозначно производить процесс переработки данных, составляются с помощью соединения символов из алфавита в предложения в соответствии с синтаксическими правилами, определяющими язык, с учетом правил семантики.

Выделяют синтаксические и семантические ошибки.

Под синтаксическими ошибками понимается нарушение правил записи программ на данном языке программирования. Они выявляются самой машиной, точнее транслятором, вовремя перевода записи алгоритма на язык машины. Исправление их осуществляется просто — достаточно сравнить формат исправляемой конструкции с синтаксисом в справочнике и исправить его.

Семантические (смысловые) ошибки — это применение операторов, которые не дают нужного эффекта (например, $a \leftarrow$ вместо $a + v$), ошибка в структуре алгоритма, в логической взаимосвязи его частей, в применении алгоритма к тем данным, к которым он неприменим и т.д. Правила семантики не формализуемы. Поэтому поиск и устранение семантической ошибки и составляет основу отладки.

Каждая программная ошибка влечет за собой необходимость изменения команд существенно меньше, чем при алгоритмических и системных ошибках.

На этапах комплексной отладки программного обеспечения и эксплуатации удельный вес программных ошибок падает и составляет примерно 15 и 30 % соответственно от общего количества ошибок, выявляемых в единицу времени.

Порядок выполнения работы:

1. Протестируйте сайт Регистрация пользователя.
2. Заполните таблицу 1

Таблица 1. Категории тяжести ошибки в программном обеспечении

Номер категории ошибки	Наименование категории тяжести ошибки	Описание последствий проявления ошибки

2. Разделите ошибки на первичные и вторичные
3. Сделайте вывод по проделанной работе

Контрольные вопросы

1. Приведите классификацию ошибок программного обеспечения
2. Какие ошибки могут возникнуть при тестировании программного обеспечения?
3. Перечислите основные пути борьбы с ошибками.

ПРАКТИЧЕСКАЯ РАБОТА № 23

Тема: Выявление первичных и вторичных ошибок

Цель работы: изучить первичные и вторичные ошибки, научиться применять методы и способы выявления ошибок, оформлять отчет о дефектах программного продукта.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для тестирования и отладки, MS Word, инструкции по выполнению работы.

Справочный материал:

При отладке и тестировании сначала обнаруживаются вторичные ошибки, т.е. результаты проявления исходных дефектов, которые являются первичными ошибками или причинами обнаруженных аномалий. Проявления дефектов и ошибок в разной степени влияют на работоспособность программы.

По величине ущерба проявления дефектов и вторичных ошибок их делят на:

1. сбои, которые не отражаются существенно на работоспособности программы и ущербом, от которых можно пренебречь.
2. ординарные отказы, ущерб от которых находится в допустимых пределах;
3. катастрофические отказы, ущерб от которых так велик, что определяет безопасность применения данного комплекса программ.

Методы обнаружения ошибок:

1. Инспекции исходного текста
2. Сквозные просмотры
3. Проверка за столом

Содержание работы:

Задание 1. Изучить код программы. Написать отчет по тестированию. Произвести отладку программного продукта на основании отчета по тестированию работоспособности. Разделите все выявленные ошибки на первичные и вторичные. Определите уровень критичности выявленных ошибок.

Условие задачи: Программа на C# ,которая создаёт два объекта, после нажатия на кнопку Запуск 1 объекта, начинает двигаться 1 объект по вытянутой параболе. Затем нажимаем на Запуск 2 объекта, он должен догнать первый.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace pr_36{
public partial class Form1 : Form {
bool l = false;
MyLabel1 D;
```

```

MyLabel T;
static public arch d = new arch();
public class MyLabel1 {
public System.Windows.Forms.Label m3;
public System.Windows.Forms.Timer MyTimer1;
double y0;
double x0;
double M;
public MyLabel1(Control Sender){
m3 = new Label();
m3.Parent = Sender;
m3.Size = new System.Drawing.Size(100, 100);
m3.Top = 190;
m3.BackColor = System.Drawing.SystemColors.Desktop;
MyTimer1 = new Timer();
MyTimer1.Interval = 1;
MyTimer1.Tick += new System.EventHandler(movett);}
public double x1 {
set{
x0 = value;
double Mhor = m3.Parent.Width / 3000.0;
m3.Left = System.Convert.ToInt32(x0 * Mhor);}
get{
return x0;}}
public double y1 {
set{
y0 = value;
double Mhor = m3.Parent.Width / 3000.0;
m3.Top = -System.Convert.ToInt32(y0 * Mhor) + m3.Parent.Height / 2 - m3.Height;}
get{
return y0;}}
public void movett(object sender, EventArgs e){
if (d.archx >= x1) x1 = x1 + 2;
else x1 = x1 - 2;
if (d.archy >= y1) y1 = y1 + 2;
else y1 = y1 - 2;}}
public class arch { //класс архив нужен чтобы второй объект получал координаты
от первого
public double archx;
public double archy;}
public class MyLabel {
double y0;
double x0;
double a = 10;
double k = 0.0003;

```

```

public System.Windows.Forms.Label m;
public System.Windows.Forms.Label m1;
public System.Windows.Forms.Label m2;
public System.Windows.Forms.Timer MyTimer;
    public MyLabel(Control Sender) { //конструктор класса
m = new Label();
m.BackColor = System.Drawing.SystemColors.GradientActiveCaption;
m.Parent = Sender;
m.Text = " 1 ";
m.Click += new System.EventHandler(this.move);
m.Left = 150;
m.Top = 190;
m.Width = 100;
m.Height = 30;
    m1 = new Label();
m1.BackColor = System.Drawing.SystemColors.GradientActiveCaption;
m1.Parent = Sender;
m1.Text = " 2 ";
m1.Click += new System.EventHandler(this.move);
m1.Top = 190;
m1.Left = m.Width / 2 - (m1.Width / 2);
m1.Left = 4;
m1.Width = 50;
m1.Height = 30;
    m2 = new Label();
m2.BackColor = System.Drawing.SystemColors.GradientActiveCaption;
m2.Parent = Sender;
m2.Text = " 3 ";
m2.Click += new System.EventHandler(this.move);
m2.Width = 25;
m2.Left = m.Width / 2 - (m2.Width / 2);
m2.Left = 10;
m2.Top = 180;
m2.Width = 25;
m2.Height = 30;
    MyTimer = new Timer();
MyTimer.Interval = 2;
MyTimer.Tick += new System.EventHandler(this.move1);}
public void move(object sender, EventArgs e){
x = x + 1;    y = y + 1;}
public void move1(object sender, EventArgs e){
xcor = x + a;
d.archx = xcor;
k = 0.0003;
ycor = k * Math.Pow(x, 2);

```

```

d.archy = ycor;}
public double xcor{
set{
x0 = value;
double Mhor = m.Parent.Width / 3000.0;
m.Left = System.Convert.ToInt32(x0 * Mhor);
m1.Left = m.Left + m.Width / 2 - (m.Width / 4);
m2.Left = m.Left + m.Width / 2 - (m.Width / 10);}
get{
return x0;}}
public double ycor{
set{
y0 = value;
double Mhor = m.Parent.Height / 3000.0;
m.Top = (m.Parent.Height - (m.Height + 30)) - System.Convert.ToInt32(y0 * Mhor);
m1.Top = m.Top - m.Height;
m2.Top = m.Top - m.Height - m1.Height;}
get{
return y0;}}
public double x{ //класс координация меток на форме по X
set{
x0 = value;
double Mhor = m.Parent.Width / 3000.0;
m.Left = System.Convert.ToInt32(x0 * Mhor);
m1.Left = m.Left + m.Width / 2 - (m.Width / 4);
m2.Left = m.Left + m.Width / 2 - (m.Width / 10);}
get{
return x0;}}
public double y{ //класс координация меток на форме по Y
set{
y0 = value;
double Mhor = m.Parent.Height / 3000.0;
m.Top = (m.Parent.Height - (m.Height + 30)) - System.Convert.ToInt32(y0 * Mhor);
m1.Top = m.Top - m.Height;
m2.Top = m.Top - m.Height - m1.Height;}
get{
return y0;}}
// public void movet(object sender, EventArgs e)// {
// xcor = x + 1; // ycor = y + 1; // }}
public Form1(){
InitializeComponent();}
private void Form1_Load(object sender, EventArgs e){ }
private void button1_Click(object sender, EventArgs e)
{// d = new arch(this);
T = new MyLabel(this);

```

```
T.x = 0;
T.y = 0;
D = new MyLabel1(this);
private void button2_Click(object sender, EventArgs e){
if (l == false) l = true;
else l = false;
T.MyTimer.Enabled = 1;}
private void button3_Click(object sender, EventArgs e){
if (l == false) l = true;
else l = false;
D.MyTimer1.Enabled = 1;}} }
```

ПРАКТИЧЕСКАЯ РАБОТА № 24

Тема: Выявление первичных и вторичных ошибок

Цель работы: изучить взаимосвязь между причинами и последствиями ошибок, а также разработать подходы для минимизации проблем, изучить первичные и вторичные ошибки, научиться применять методы и способы выявления ошибок, оформлять отчет о дефектах программного продукта.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для тестирования и отладки, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1.

Порядок выполнения работы:

1. Протестируйте сайт Интернет-магазин.
2. Заполните таблицу 1

Таблица 1. Категории тяжести ошибки в программном обеспечении

Номер категории ошибки	Наименование категории тяжести ошибки	Описание последствий проявления ошибки

3. Разделите ошибки на первичные и вторичные
4. Сделайте вывод по проделанной работе

ПРАКТИЧЕСКАЯ РАБОТА № 25

Тема: Управление базовым планом с помощью инструментов в программе MS Project

Цель работы: освоить инструменты для работы с базовым планом, которые позволяют сравнивать фактический прогресс с планируемым и анализировать отклонения

Оборудование: ПК, интернет, программное обеспечение – MS Project, MS Word, инструкции по выполнению работы.

Справочный материал:

Базовый план представляет собой набор предварительных оценок начальных и конечных дат, длительностей, трудозатрат и затрат, сохраненных после завершения и настройки плана проекта, но до начала проекта. Базовый план является основным критерием оценки изменений проекта. Помимо этого, в базовом плане сохраняются около 20 наборов сведений, включая итоги и повременные сведения для задач, ресурсов и назначений. Можно сохранить до 11 базовых планов.

Поскольку базовый план содержит данные, с которыми сравнивается фактический ход выполнения проекта, этот план должен содержать наиболее точные предварительные оценки для длительности задач, начальных и конечных дат, затрат, а также других переменных проекта, которые требуется отслеживать. Если данные базового плана значительно отличаются от текущих данных, это означает, что базовый план был составлен неточно. Обычно такие различия возникают, если изменяются содержание проекта или основная цель проекта. Если заинтересованные стороны проекта согласны, что различия это оправдывают, то в любой момент можно изменить или доработать базовый план. Сохранение нескольких базовых планов особенно удобно для продолжительных проектов или для проектов, в которых запланированные задачи или затраты были значительно изменены, а данные первоначального базового плана больше не действительны.

Промежуточный план представляет собой набор текущих данных проекта, сохраненный после начала проекта, который сравнивается с базовым планом для анализа хода выполнения проекта. В промежуточном проекте сохраняются два набора сведений (начальные и конечные даты текущих задач). Допускается сохранение до 10 промежуточных планов. Если необходимо сделать снимки большого количества данных проекта на стадии планирования, можно сохранить несколько базовых планов. Затем, после начала проекта, если необходимо сохранить только даты начала и окончания задач, можно сохранить несколько промежуточных планов. Например, эти действия можно выполнять ежемесячно или один раз в квартал.

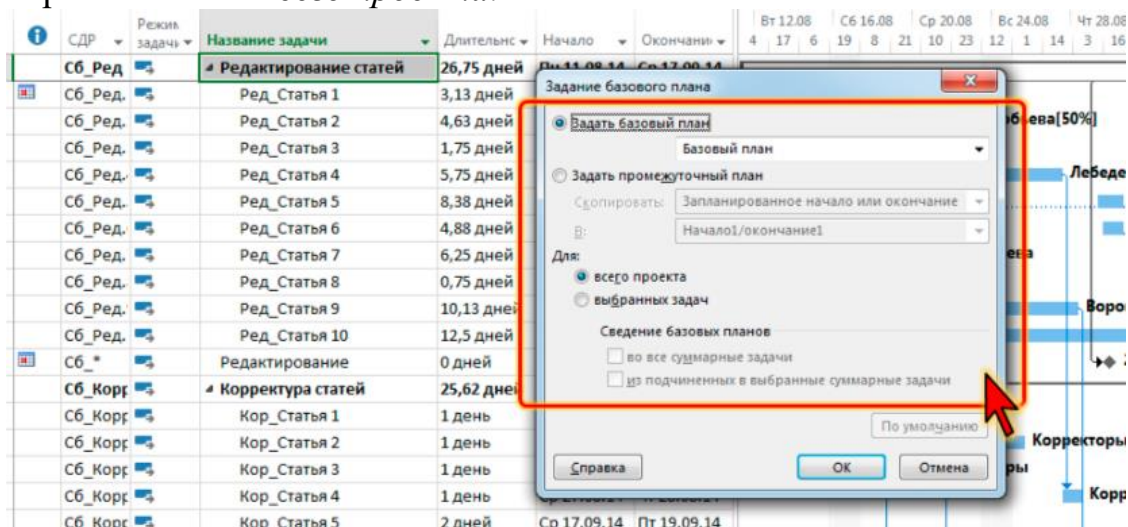
Содержание работы:

Задание 1. Создание базового плана

Обычно базовый план создают (сохраняют) после завершения и настройки плана проекта.

- На вкладке *Проект* в группе *Планирование* нажмите на кнопку *Задать базовый план* и в появившемся меню выберите команду *Задать базовый план*.

- В появившемся окне *Задание базового плана* в разделе *Для* выберите переключатель *Всего проекта*.



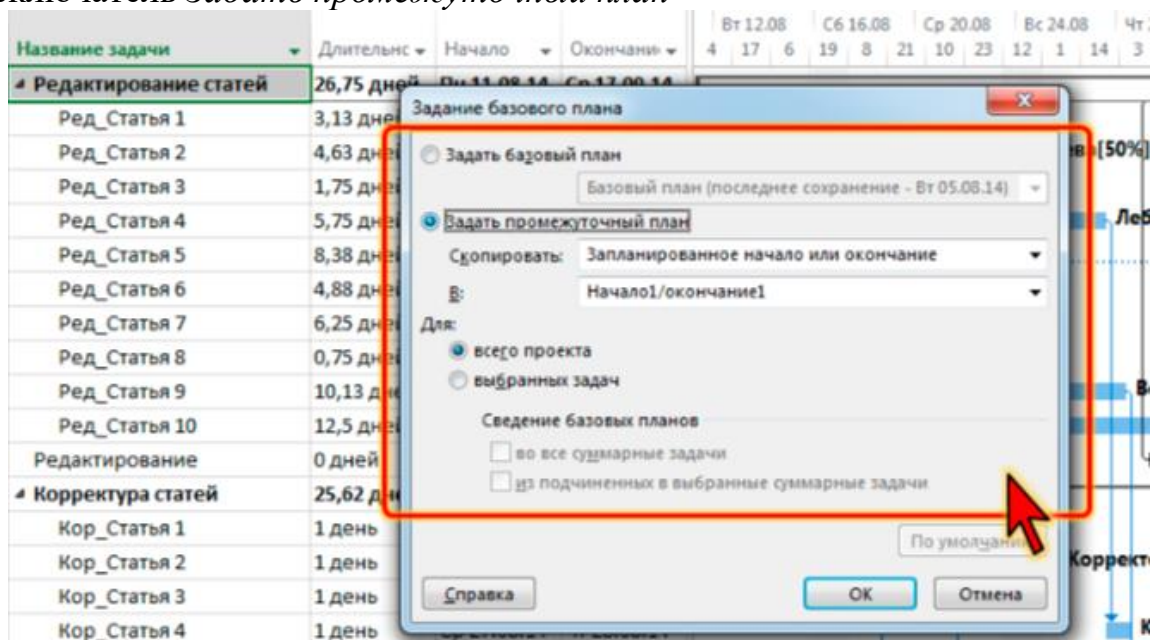
Созданному базовому плану будет добавлена дата сохранения.

Следует обратить внимание, что при сохранении другого базового плана этого же проекта в верхнем раскрывающемся списке следует выбрать пустой базовый план.

Задание 2. Сохранение базового плана

Если требуется сохранять только даты начала и окончания выполнения задач, можно сохранить промежуточный план.

- На вкладке *Проект* в группе *Планирование* нажмите на кнопку *Задать базовый план* и в появившемся меню выберите команду *Задать базовый план*.
- В появившемся окне *Задание базового плана* в установите переключатель *Задать промежуточный план*



- В раскрывающемся списке *Копировать* выберите начало и окончание действия базовых значений, которые нужно сохранить (текущие показатели начала и окончания, а также базовые значения не нумеруются).

- В поле раскрывающемся списке *В* выберите название промежуточного плана, в который нужно скопировать значения (промежуточные планы хранятся в полях *Начало/Окончание*).
- В разделе *Для* выберите *Весь проект* или *Выбранные задачи*, чтобы сохранить нужную часть календарного плана.

Следует обратить внимание, что при выборе одного базового плана в списке *Копировать* и другого базового плана в списке *В* будет сохранен базовый, а не промежуточный план. Копируются все данные базового плана. При выборе базового плана в списке *Копировать* и промежуточного плана начала и окончания в поле *В* в промежуточный план из базового плана копируются только даты начала и окончания.

И еще следует обратить внимание, что в отличие от базового плана, при сохранении промежуточного к его названию не будет добавлена дата сохранения.

Задание 3. Обновление базового или промежуточного плана

Если задача добавляется в проект после задания базового (промежуточно) плана, можно добавить ее в него.

- Выделите новую задачу, которая добавляется в базовый план.
- Если задача содержит подзадачи, обязательно выберите и их.
- На вкладке *Проект* группе *Планирование* нажмите на кнопку *Задать базовый план* и в появившемся меню выберите команду *Задать базовый план*.
- Для обновления базового (промежуточного) плана установите соответствующий переключатель и в раскрывающемся списке окна *Задание базового плана* выберите план, подлежащий обновлению.
- В разделе *Для* выберите переключатель *Выбранные задачи*.
- При добавлении подзадач в разделе *Сведение базовых планов* выберите один из следующих вариантов, чтобы указать способ добавления подзадач к суммарным задачам.

Задание 4. Сравнение базового и календарного плана

Сравнить базовый и текущий календарный план можно как в таблице, так и на диаграмме Ганта.

- Установите представление *Диаграмма Ганта с отслеживанием*.
- На вкладке *Вид* в группе *Данные* нажмите кнопку *Таблицы* и выберите *Отклонение*.
- Если в проекте есть несколько базовых планов, на вкладке *Формат* в группе *Стили отрезков* щелкните по кнопке *Базовый план* и выберите нужный.

На диаграмме Ганта под отрезками задач текущего плана появятся отрезки базового плана. В таблице будут отображены столбцы, характеризующие как текущий календарный план, так и базовый, а также отклонение текущего от базового.

Для сравнения несколько базовых планов используют представление *Диаграмма Ганта с несколькими планами*. По умолчанию

показаны три первых базовых плана *Базовый план*, *Базовый план1* и *Базовый план2*.

Задание 5. Удаление базового или промежуточного плана

Можно удалить (очистить) любой базовый или промежуточный план.

- На вкладке *Проект* в группе *Планирование* нажмите на кнопку *Задать базовый план* и в появившемся меню выберите команду *Очистить базовый план*.
- В появившемся окне *Очистить базовый* установите переключатель *Очистить базовый план*, а затем выберите в раскрывающемся списке название базового плана, который требуется очистить.
- Для очистки промежуточного плана установите переключатель *Очистить промежуточный план*, а затем выберите в раскрывающемся списке название промежуточного плана, который требуется очистить.
- Выберите параметр *всего проекта* или *выбранных задач*, чтобы очистить только ту часть календарного плана, которую требуется.

ПРАКТИЧЕСКАЯ РАБОТА № 26

Тема: Управление базовым планом с помощью инструментов в программе MS Project

Цель работы: освоить инструменты для работы с базовым планом, которые позволяют сравнивать фактический прогресс с планируемым и анализировать отклонения

Оборудование: ПК, интернет, программное обеспечение – MS Project, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1.

1. Запустите Project.
2. Откройте файл *Zadanie 01.mpp*.
3. Сохраните текущий календарный план как базовый план.
4. Выровняйте загрузку ресурсов.
5. Сохраните проект как файл под именем *Lesson_01_01*.

Задание 2

1. Откройте файл *Zadanie 02.mpp*.
2. На диаграмме Ганта отобразите базовый план от 16 августа 2025 года.
3. На диаграмме Ганта отобразите базовый план от 17 августа 2025 года.
4. На диаграмме Ганта отобразите базовый план от 19 августа 2025 года.
5. Базовый план от 16 августа 2025 года замените текущим планом.
6. Удалите базовый план от 17 августа 2025 года.
7. Сохраните проект как файл под именем *Lesson_01_02*.

Задание 3

1. Откройте файл *Zadanie 03.mpp*.
2. Установите дату начала проекта (сдвиньте проект) на две недели назад по отношению ко дню, в который выполняется упражнение. Если в результате сдвига проекта произойдет превышение доступности некоторых ресурсов, выполните автоматическое выравнивание загрузки.
3. Для всех задач проекта установите такое выполнение, которое должно быть к текущей дате при строгом выполнении проекта.
4. Укажите, что задачи *Ред_Статья 6* и *Ред_Статья 7* выполнены на половину.
5. Укажите, что задача *Ред_Раздел 3* выполнена полностью.
6. Укажите, что задача *Ред_Раздел 13* выполнена на 13 %.
7. Сохраните проект как файл под именем *Lesson_01_03*.

Задание 4

1. Откройте файл *Zadanie 04.mpp*.
2. Создайте три бюджетных ресурса: трудовой ресурс *Труд*, материальный ресурс *Бумага*, затратный ресурс *Затраты*.
3. Отобразите суммарную задачу проекта. Назначьте суммарной задаче бюджетные ресурсы.
4. Отобразите представление *Использование ресурсов*. В таблице *Затраты* вставьте столбцы *Бюджетные затраты* и *Бюджетные трудозатраты*.

5. Введите значения бюджетных ресурсов: *Труд* – 1500 часов, *Бумага* – 250 пачек, *Затраты* 60000 рублей.
6. Отобразите представление *Лист ресурсов*. С использованием настраиваемых полей добавьте столбец, позволяющий вводить текстовые данные.
7. Для каждого ресурса, включая бюджетные, введите тестовые идентификаторы, позволяющие сгруппировать их по типам (трудовые, материальные, затратные).
8. Отобразите представление *Использование ресурсов*. Сгруппируйте данные по созданному идентификатору.
9. Проверьте, достаточное ли количество ресурсов выделено проекту.
10. Сохраните проект как файл под именем *Lesson_01_04*.
11. Закройте Project.

ПРАКТИЧЕСКАЯ РАБОТА № 27

Тема: Работа с отчетами и представлениями

Цель работы: освоить инструменты для работы с отчетами и представлениями проекта.

Оборудование: ПК, интернет, программное обеспечение – MS Project, MS Word, инструкции по выполнению работы.

Справочный материал:


Самым простым отчетом, содержащим обобщенные данные о проекте, является окно статистики проекта.

Окно статистики проекта открывается кнопкой Статистика из окна сведений о проекте (Проект / Сведения о проекте). Оно разделено на две части. В верхней части отображается таблица с данными о датах начала и окончания проекта. В первой строке отображаются даты по текущему плану, во второй – по базовому плану, в третьей – фактические даты. В последней строке отображается отклонение фактических данных от данных базового плана.

В нижней части окна находится таблица со сводными данными по длительности, трудозатратам и затратам на проект, сгруппированными по колонкам. В верхней строке таблицы отображаются данные текущего плана, во второй строке – базового, в третьей – фактические, а в четвертой – оставшиеся. Под таблицей выводятся сведения о проценте завершения проекта по длительности и трудозатратам.

Содержание работы:

Задание 1. Сформировать отчет по проекту Разработка программы

1. Формирование статистики проекта. Открыть окно сведений о проекте Проект – Сведения о проекте, нажать кнопку Статистика, откроется окно статистики проекта.
2. Формирование стандартного отчета. Открыть окно с категориями стандартных отчетов – Отчет / Отчеты. Двойным щелчком мыши выбрать категорию Назначения – откроется окно со списком отчетов этой категории. Двойным щелчком мыши выбрать отчет Дела по исполнителям и времени – первая страница отчета выводится на экран. Отчет показывает ежедневное распределение трудозатрат исполнителей по задачам. При помощи кнопок  листаем страницы отчета по горизонтали. После просмотра для выхода используется кнопка Заккрыть.
3. Настройка стандартного отчета. В окне с категориями стандартных отчетов выбрать категорию Назначения – откроется окно со списком отчетов этой категории. Щелчком мыши выбрать отчет Дела по исполнителям и времени и нажать кнопку Изменить. В открывшемся окне свойств перекрестного отчета изменить в поле Столбец значение Дни на Недели и нажать Ок. На экране снова список отчетов категории Назначения. Двойным щелчком мыши открыть отчет Дела по исполнителям и времени. В результат в отличие от предыдущего варианта в данном отчете трудозатраты исполнителей сгруппированы по неделям.
4. Создание настраиваемого отчета. Открыть список категорий отчетов: Отчет – Отчеты. Двойным щелчком мыши открыть категорию Настраиваемые. Кнопкой

Создать открыть список типов отчетов. Выбрать Перекрестная таблица и нажать Ok. Заполнить окно свойств отчета в соответствии с изображением на рисунке. Нажать Ok.

Для формирования отчета, после нажатия кнопки Ok, нажать кнопку Просмотр – формируется отчет, изображенный на рисунке.

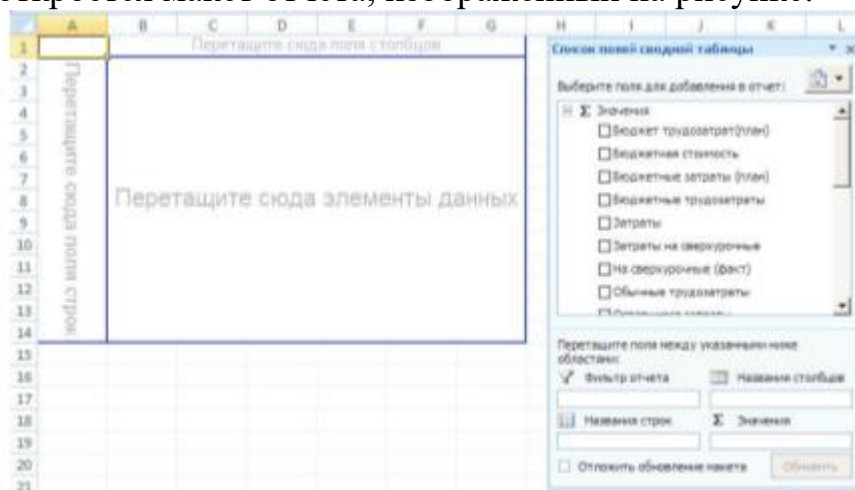
Затраты задач от 16.06.11
Разработка программыПроект

	16.01	17.01	24.01	31.01	07.02	14.02	21.02	28.02
Начало реализации проекта								
Постановка задачи	1 600 000,00р.	167 555,56р.	80 777,78р.					
Постановка	160 000,00р.	160 000,00р.	80 000,00р.					
Компьютер	1 500 000,00р.							
DVD-матрица	3 000,00р.							
Междугородние перелеты	10 888,89р.	7 555,56р.	3 777,78р.					
Расходные материалы	10 000,00р.							
Разработка интерфейса				1 555 500,00р.	140 500,00р.			
Программист1				52 500,00р.	122 500,00р.			
Компьютер				1 500 000,00р.				
DVD-матрица					15 000,00р.			
Междугородние перелеты				4 000,00р.	5 000,00р.			
Разработка модулей обработки данных					1 555 470,59р.	152 352,94р.	78 176,47р.	
Программист1					35 000,00р.	140 000,00р.	70 000,00р.	
Компьютер					1 500 000,00р.			
DVD-матрица					23 470,59р.	12 352,94р.	6 176,47р.	
Разработка структуры базы данных			1 627 500,00р.	109 500,00р.				
Программист1			105 000,00р.	105 000,00р.				
Компьютер			1 500 000,00р.					
DVD-матрица			18 000,00р.					
Междугородние перелеты			4 500,00р.	4 500,00р.				
Заполнение базы данных								1 594 000,00р.
Программист2								70 000,00р.
Компьютер								1 500 000,00р.
DVD-матрица								24 000,00р.
Программирование закрываю								
Отладка программного комплекса						244 200,00р.	259 300,00р.	52 500,00р.
Постановка						100 000,00р.	100 000,00р.	
Программист1						17 500,00р.	105 000,00р.	52 500,00р.
Программист2						122 500,00р.	52 500,00р.	
DVD-матрица						4 200,00р.	1 800,00р.	
Тестирование и исправление ошибок								277 100,00р.
Постановка								100 000,00р.
Программист1								122 500,00р.
Программист2								52 500,00р.
DVD-матрица								2 100,00р.
Составление программной документации								
Постановка								
DVD-матрица								
Междугородние перелеты								
Расходные материалы								

5. Формирование наглядного отчета. Открыть окно наглядных отчетов – Отчет / Наглядные отчеты. На вкладке Все выбрать отчет Сводный отчет о затратах ресурсов и нажать кнопку Просмотреть. Отчет формируется в системе Excel и имеет вид круговой диаграммы, изображенной на рисунке.



6. Создание наглядного отчета. В окне наглядных отчетов (Отчет / Наглядные отчеты) нажать кнопку Создать шаблон. В поле Выберите данные, по которым требуется создать отчет установить Сводка по задачам. Ok. В системе Excel откроется макет отчета, изображенный на рисунке.



Перетащить из списка полей Задачи в область полей строк, а Затраты – в область элементов данных. Результат изображен на рисунке.

	A	B	C	D
1	Затраты			
2	Задачи	Задачи 01	Задачи 02	Итого
3	Разработка программы/Проект	Разработка программы/Проект		0
4		Начало реализации проекта		0
5		Программирование	Программирование	0
6			Постановка задачи	1935222.222
7			Разработка интерфейса	1700000
8			Разработка модулей обработки данных	1787000
9			Разработка структуры базы данных	1737000
10			Заполнение базы данных	1804000
11			Программирование завершено	0
12		Программирование Итого		8963222.222
13		Отладка	Отладка	0
14			Отладка программного комплекса	556000
15			Тестирование и исправление ошибок	1106000
16			Составление программной документации	250000
17			Отладка завершена	0
18		Отладка Итого		1912000
19		Конец проекта		0
20	Разработка программы/Проект Итого			10875222.22
21	Общий итог			10875222.22

Задачи второго уровня отображаются в отчете при помощи кнопки выбора поля Задачи.

ПРАКТИЧЕСКАЯ РАБОТА № 28

Тема: Работа с отчетами и представлениями

Цель работы: Цель работы: освоить инструменты для работы с отчетами и представлениями проекта.

Оборудование: ПК, интернет, программное обеспечение – MS Project, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Самостоятельно ввести фактические данные о выполнении работ различными способами для задач, принадлежащих разным фазам в соответствии с таблицей

Способы анализа хода выполнения проекта

Способ ввода фактических данных	Фаза			Имя файла результата
	1 вариант	2 вариант	3 вариант	
Процент завершения	Утверждение проектов	Выбор системы	Потолки	Процент.mprp
Ввод фактических или оставшихся трудозатрат с перерывом в работе	Строительство фундамента	Приобретение программного обеспечения	Ванная	Факт.mprp
Ввод повременных данных задач	Каркас и крыша	Составление проекта сети	Пол	Задачи.mprp
Ввод повременных данных ресурсов	Коммуникации	Приобретение компьютеров и сетевого оборудования	Выравнивание стен	Ресурсы.mprp

Выполнить анализ хода выполнения проекта по методике освоенного объема на предполагаемую дату окончания работ и сделать вывод о скорости выполнения работ (раньше или позже плана) и предполагаемом объеме фактических затрат (больше или меньше плановых).

Задание 2. Для созданных проектов сформировать следующие виды отчетов:

- Статистику проекта.
- Любой из стандартных отчетов.
- Любой из стандартных отчетов с предварительно измененными параметрами.
- Свои собственные настраиваемые отчеты всех четырех видов.
- Предопределенный наглядный отчет. Результат сохранить в файл Отчет1.xls.
- Самостоятельно созданный наглядный отчет. Результат сохранить в файл Отчет2.xls.

ПРАКТИЧЕСКАЯ РАБОТА № 29

Тема: Тестирование программных продуктов

Цель работы: изучить классификацию видов тестирования, практически закрепить эти знания путем генерации тестов различных видов, научиться планировать тестовые активности в зависимости от специфики поставляемой на тестирование функциональности.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для тестирования, MS Word, инструкции по выполнению работы.

Справочный материал:

Типы тестов по покрытию (по глубине):

- 1.Smoke test – тестирование системы для определения корректной работы базовых функций программы в целом, без углубления в детали.
- 2.Minimal Acceptance Test (MAT, Positive test) - тестирование системы или ее части только на валидных данных
- 3.Аcceptance Test (AT) - полное тестирование системы или ее части как на корректных, так и на некорректных данных/сценариях.

Тестовые активности (типы тестов по покрытию (по ширине)):

- 1.Defect Validation – проверка результата исправления дефектов.
- 2.New Feature Test (NFT, AT of NF) – определение качества поставленной на тестирование новой функциональности, которая ранее не тестировалась.
- 3.Regression testing (регрессионное тестирование) – проводится с целью оценки качества ранее реализованной функциональности.

Типы тестов по знанию коду

- 1.Черный ящик – тестирование системы, функциональное или нефункциональное, без знания внутренней структуры и компонентов системы. У тестировщика нет доступа к внутренней структуре и коду приложения либо в процессе тестирования он не обращается к ним.
- 2.Белый ящик – тестирование основанное на анализе внутренней структуры компонентов или системы. У тестировщика есть доступ к внутренней структуре и коду приложения.
- 3.Серый ящик – комбинация методов белого и черного ящика, состоящая в том, что к части кода архитектуры у тестировщика есть, а к части кода – нет.

Типы тестов по степени автоматизации

- 1.Ручное – тестирование, в котором тест-кейсы выполняются тестировщиком вручную без использования средств автоматизации.
- 2.Автоматизированное – набор техник, подходов и инструментальных средств, позволяющий исключить человека из выполнения некоторых задач в процессе тестирования.

Типы тестов по изолированности компонентов

- 1.Unit/component (модульное) – тестирование отдельных компонентов (модулей) программного обеспечения.
- 2.Integration (интеграционное) – тестируется взаимодействие между интегрированными компонентами или системами.
- 3.System (системное) – тестируется работоспособность системы в целом с целью проверки того, что она соответствует установленным требованиям.

Типы тестов по подготовленности.

1. Интуитивное тестирование выполняется без подготовки к тестам, без определения ожидаемых результатов, проектирования тестовых сценариев.
2. Исследовательское тестирование – метод проектирования тестовых сценариев во время выполнения этих сценариев. Тестировщик совершает проверки, продумывает их, придумывает новые проверки, часто использует для этого полученную информацию.
3. Тестирование по документации – тестирование по подготовленным тестовым сценариям, руководству по осуществлению тестов.

Типы тестов по месту и времени проведения

1. User Acceptance Testing (UAT) (приемочное тестирование) – формальное тестирование по отношению к потребностям, требованиям и бизнес процессам пользователя, проводимое с целью определения соответствия системы критериям приёмки и дать возможность пользователям, заказчикам или иным авторизованным лицам определить, принимать систему.
2. Alpha Testing (альфа-тестирование) – моделируемое или действительное функциональное тестирование, выполняется в организации, разрабатывающей продукт, но не проектной командой (это может быть независимая команда тестировщиков, потенциальные пользователи, заказчики).
3. Beta Testing (бета-тестирование) – эксплуатационное тестирование потенциальными или существующими клиентами/заказчиками на внешней стороне (в среде, где продукт будет использоваться) никак связанными с разработчиками, с целью определения действительно ли компонент или система удовлетворяет требованиям клиента/заказчика и вписывается в бизнес-процессы.

Типы тестов по объекту тестирования

1. Functional testing (функциональное тестирование) – это тестирование, основанное на анализе спецификации, функциональности компонента или системы.
2. Safety testing (тестирование безопасности) – тестирование программного продукта с целью определить его безопасность
3. Security testing (тестирование защищенности) – это тестирование с целью оценить защищенность программного продукта.
4. Compatibility testing (тестирование совместимости) – процесс тестирования для определения возможности взаимодействия программного продукта, проверка работоспособности приложения в различных средах

Нефункциональное тестирование – это проверка характеристик программы.

1. Тестирование пользовательского интерфейса (GUI) – тестирование, выполняемое путем взаимодействия с системой через графический интерфейс пользователя.

- навигация
- цвета, графика, оформление
- содержание выводимой информации
- поведение курсора и горячие клавиши

- отображение различного количества данных
- изменение размеров окна или разрешения экрана

2. Тестирование удобства использования (Usability Testing) – тестирование с целью определения степени понятности, легкости в изучении и использовании, привлекательности

программного продукта для пользователя при условии использования в заданных условиях эксплуатации.

- визуальное оформление
- навигация
- логичность

3. Тестирование доступности (Accessibility testing) – тестирование, которое определяет степень

легкости, с которой пользователи с ограниченными способностями могут использовать систему или ее компоненты.

4. Тестирование интернационализации – тестирование способности продукта работать в локализованных средах

5. Тестирование локализации (Localization testing) – тестирование, проводимое с целью проверить качество перевода продукта с одного языка на другой.

6. Тестирование производительности или нагрузочное тестирование – процесс тестирования с целью определения производительности программного продукта.

Виды тестов:

– нагрузочное тестирование (Performance and Load testing) – вид тестирования производительности, проводимый с целью оценки поведения компонента или системы при возрастающей нагрузке, например количестве параллельных пользователей и/или операций, а также определения какую нагрузку может выдержать компонент или система;

– объемное тестирование (Volume testing) – позволяет получить оценку производительности при увеличении объемов данных в базе данных приложения;

– тестирование стабильности и надежности (Stability / Reliability testing) – позволяет проверять работоспособность приложения при длительном (многочасовом) тестировании со средним уровнем нагрузки.

– стрессовое тестирование (Stress testing) – вид тестирования производительности, оценивающий систему или компонент на граничных значениях рабочих нагрузок или за их пределами, или же в состоянии ограниченных ресурсов, таких как память или доступ к серверу.

7. Тестирование требований (Requirements testing) – проверка требований на соответствие основным характеристикам качества.

8. Тестирование прототипа (Prototype testing) – метод выявления структурных, логических ошибок и ошибок проектирования на ранней стадии развития продукта до начала фактической разработки.

9. Тестирование установки (Installability testing) и лицензирования – процесс тестирования устанавливаемости программного продукта.

Виды тестов:

- формальный тест программы установки приложения (проверка пользовательского интерфейса, навигации, удобства пользования, соответствия общепринятым стандартам оформления);
- функциональный тест программы установки;
- тестирование механизма лицензирования и функций защиты от пиратства;
- проверка стабильности приложения после установки.

10. Тестирование на отказ и восстановление (Failover and Recovery Testing) – тестирование при помощи эмуляции отказов системы или реально вызываемых отказов в управляемом окружении.

Тестирование программного продукта включает следующие этапы:

1. Изучение и анализ предмета тестирования.
2. Планирование тестирования.
3. Исполнение тестирования.

Содержание работы:

Задание 1. Дан массив, состоящий из 15 элементов целого типа. Получить новый массив, как разность между элементами исходного массива и его среднего арифметического.

```
using System;
namespace Tasks2{
class Program{
static void Main(string[] args){
    int[] mass = new int[15]; //создаем массив
        Random rnd = new Random();
        int summ = 0; //переменная для суммы всех элементов массива
        string s=null; //строка для вывода массива на экран
        for (int i = 0; i < 15; i++)      {
            mass[i] = rnd.Next(1, 100); //задаем рандомное значение элементу
массива, можно самому вручную, разницы нет
            s = s + mass[i].ToString() + " "; //дополняем строку
            summ = summ + mass[i]; }//суммируем элементы массива
            Console.WriteLine("Начальный массив = " + s); //выводим элементы
массива
            Console.WriteLine("Среднее арифметическое = " + (summ/15));
            s = null;
            for (int i = 0; i < 15; i++)      {
                mass[i] = mass[i] - (summ / 15); //теперь вычитаем из элементов массива
среднее арифметическое
                s = s + mass[i].ToString() + " ";      }
            Console.WriteLine("Новый массив = " + s);
            Console.ReadKey();}}}
```

1. Выбрать методы тестирования программного продукта.
2. Разработать план тестирования программного продукта.

3. Провести тестирование программы и представить результаты в виде таблицы 1

Таблица 1. Результаты тестирования

Тест (значения для входных данных)	Ожидаемый результат (значения для выходных данных)	Фактический результат (полученные значения выходных данных)	Результат тестирования (успешно/неуспешно)

4. Выработать рекомендации для корректировки тестируемой программы.

Контрольные вопросы

- 1.Что такое тестирование программного обеспечения?
- 2.Чем тестирование отличается от отладки?
- 3.Для чего проводится функциональное тестирование?
- 4.Что такое комплексное тестирование?
- 5.Каковы правила тестирования программы «как черного ящика»?
- 6.Как проводится тестирования программы по принципу «белого ящика»?
- 7.Что такое модульное тестирование?
- 8.Как осуществляется сборка программы при модульно тестировании?

ПРАКТИЧЕСКАЯ РАБОТА № 30

Тема: Тестирование программных продуктов

Цель работы: изучить классификацию видов тестирования, практически закрепить эти знания путем генерации тестов различных видов, научиться планировать тестовые активности в зависимости от специфики поставляемой на тестирование функциональности.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для тестирования, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1.

1. Выполнить генерацию тестов различных видов следующих программных продуктов:
 - программа регистрации посетителей;
 - программа проверки знаний.
2. Выбрать методы тестирования программного продукта.
3. Разработать план тестирования программного продукта.
4. Провести тестирование программы и представить результаты в виде таблицы 1

Таблица 1. Результаты тестирования

Тест (значения для входных данных)	Ожидаемый результат (значения для выходных данных)	Фактический результат (полученные значения выходных данных)	Результат тестирования (успешно/неуспешно)

5. Выработать рекомендации для корректировки тестируемой программы.

ПРАКТИЧЕСКАЯ РАБОТА № 31

Тема: Тестирование программных продуктов

Цель работы: изучить классификацию видов тестирования, практически закрепить эти знания путем генерации тестов различных видов, научиться планировать тестовые активности в зависимости от специфики поставляемой на тестирование функциональности.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для тестирования, MS Word, инструкции по выполнению работы.

Справочный материал:

Метод белого ящика (white box testing, open box testing, clear box testing, glass box testing) – у тестировщика есть доступ к внутренней структуре и коду приложения, а также есть достаточно знаний для понимания увиденного.

Разработка тестов методом белого ящика (white-box test design technique): Процедура разработки или выбора тестовых сценариев на основании анализа внутренней структуры компонента или системы.

Техники, основанные на структуре, или методе белого ящика:

- тестирование операторов;
- тестирование альтернатив.

Альтернатива (decision): Точка программы, в которой управление имеет два или более альтернативных путей. Узел с двумя или более связями для разделения ветвей. Тестирование условий альтернатив (decision condition testing): Разработка тестов методом белого ящика, при котором тестовые сценарии проектируются для исходов условий и результатов альтернатив.

Покрытие (coverage): Уровень, выражаемый в процентах, на который определенный элемент покрытия был проверен набором тестов. Покрытие альтернатив (decision coverage): Процент результатов альтернативы, который был проверен набором тестов. Стопроцентное покрытие решений подразумевает стопроцентное покрытие ветвей и стопроцентное покрытие операторов.

Покрытие кода (code coverage): Метод анализа, определяющий, какие части программного обеспечения были проверены (покрыты) набором тестов, а какие нет, например, покрытие операторов, покрытие альтернатив или покрытие условий. Еще выделяют серый ящик.

Содержание работы:

Задание 1. Разработать программу на любом языке программирования.

Даны длины сторон треугольника, определить вид треугольника и его площадь. Выполнить контроль вводимых чисел.

1. Разносторонний треугольник
2. Равнобедренный треугольник
3. Равносторонний треугольник

Ограничения:

- три числа не могут быть определены как стороны треугольника;
- если хотя бы одно из них меньше или равно 0;
- сумма двух из них меньше третьего.

Задание 2. Подготовить набор тестовых вариантов для обнаружения ошибок в программе. Результат оформить в следующем виде:

А	В	С	Ожидаемый результат	Полученный результат
Значение	Значение	Значение	Что должно получиться	Значения вводимых данных, либо ожидаемый результат
...

Задание 3. На основании проведенных тестов составьте рекомендации по исправлению ошибок, выявленных в ходе тестирования в виде отчета.

Пример:

1 тест. В ходе проведения первого теста было обнаружено, что при введении не корректных данных площадь все равно высчитывается.

Рекомендуется: в случае, если пользователь введет не корректные данные, следует выводить сообщение с просьбой исправить введенные значения. Добавить в программу проверку введенных значений на соответствие ограничения.

ПРАКТИЧЕСКАЯ РАБОТА № 32

Тема: Тестирование программных продуктов

Цель работы: изучить классификацию видов тестирования, практически закрепить эти знания путем генерации тестов различных видов, научиться планировать тестовые активности в зависимости от специфики поставляемой на тестирование функциональности.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для тестирования, MS Word, инструкции по выполнению работы.

Справочный материал:

Метод чёрного ящика (black box testing, closed box testing) – у тестирующего либо нет доступа к внутренней структуре и коду приложения, либо недостаточно знаний для их понимания, либо он сознательно не обращается к ним в процессе тестирования.

Разработка тестов методом черного ящика (black box test design technique) Процедура создания и/или выбора тестовых сценариев, основанная на анализе функциональной или нефункциональной спецификации компонента или системы без знания внутренней структуры.

Техники разработки тестов на основе спецификаций, или методе черного ящика:

- эквивалентное разбиение;
- анализ граничных значений;
- тестирование таблицы решений;

Эквивалентное разбиение (equivalence partitioning). Разработка тестов методом черного ящика, в которой тестовые сценарии создаются для проверки элементов эквивалентной области. Как правило, тестовые сценарии разрабатываются для покрытия каждой области как минимум один раз.

Входные данные для программного обеспечения или системы разбиваются на группы, от которых ожидается сходное поведение, то есть они должны обрабатываться аналогичным образом. Эквивалентные области (или классы) могут быть определены как для валидных, так и для невалидных данных, то есть тех значений, которые должны отвергаться. Эквивалентное разбиение применимо на всех уровнях тестирования. Эквивалентное разбиение может быть использовано с целью покрытия входных и выходных данных. Оно может применяться при ручном вводе данных, при передаче данных через интерфейсы в систему, или при проверке параметров интерфейсов в интеграционном тестировании.

Анализ граничных значений (boundary value analysis): Разработка тестов методом черного ящика, при котором тестовые сценарии проектируются на основании граничных значений.

Граничное значение (boundary value): Входное значение или выходные данные, которое находится на грани эквивалентной области или на наименьшем расстоянии от обеих сторон грани, например, минимальное или максимальное значение области. Анализ граничных значений может применяться на всех уровнях тестирования

Таблица решений (decision table): Таблица, отражающая комбинации входных данных и/или причин с соответствующими выходными данными и/или действиям (следствиям), которая может быть использована для проектирования тестовых сценариев. Таблицы решений – хороший метод для сбора системных требований, содержащих логические условия и документирования внутреннего дизайна системы. Они могут использоваться для записи сложных бизнес-правил, которые должна реализовывать система. Анализируются спецификации и определяются условия и действия системы. Входные условия и действия чаще всего формулируются таким образом, чтобы они могли принимать логические значения «истина» или «ложь».

Сильной стороной тестирования таблицы решений является то, что она создает комбинации условий, которые могли бы быть не проверены в ходе тестирования иным способом. Этот метод может быть применен ко всем ситуациям, в которых действие программного продукта зависит от нескольких логических альтернатив.

Содержание работы:

Задание 1. Написать калькулятор с небольшими багами. Обменяться программой с другими студентами. Провести тестирование и написать отчет в виде таблицы

Название теста	Описание сценария	Входные данные	Выходные данные	Удачное/неудачное тестирование	Предложения по исправлению найденных ошибок.	Пожелания пользователей
Функция суммы	Сложение двух положительных чисел; Проверка результата	Первая переменная=3 Вторая переменная=8	Результат=11	Неудачное	-	Поле для ввода значений и вывода, объединить

ПРАКТИЧЕСКАЯ РАБОТА № 33

Тема: Математическое моделирование методов шифрования

Цель работы: изучить основные математические основы методов шифрования, научиться моделировать процессы шифрования и расшифровки с помощью математических методов, а также применить полученные знания для решения практических задач.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для шифрования, MS Word, инструкции по выполнению работы.

Справочный материал:

Шифрование — это процесс преобразования открытых данных в зашифрованный вид с целью обеспечения их конфиденциальности. Моделирование методов шифрования позволяет понять их структуру и оценить безопасность.

Шифр — алгоритм преобразования текста.

Ключ — параметр, определяющий конкретное преобразование.

Шифротекст — результат шифрования.

Расшифровка — обратный процесс восстановления исходных данных.

В моделях шифрования используются различные математические структуры:

- Арифметические операции (сложение, умножение по модулю)
- Группы, кольца, поля
- Теория чисел (например, простые числа, делимость)
- Математические функции (например, функции Эйлера, обратные элементы)

Содержание работы:

Задание 1. Написать программу, шифрующую исходный текст методом простой замены. Каждая буква исходного текста заменяется буквой, стоящей справа от неё в русском «закольцованном» алфавите, т.е. буква «Я» заменяется на «А» (буквы заглавные).

2 Результаты шифрования и исходный текст размещайте в текстовом файле (результат выводить группами по пять букв).

Задание 2. Усовершенствовать программу, полученную в п.1, так, чтобы можно было регулировать направление замены, т.е. была бы возможность замены буквы текста на букву слева от неё в алфавите.

Задание 3. Написать программу, восстанавливающую текст, зашифрованный в усовершенствованной программе в задании 2 (выводить группами по пять букв). Направление замены регулируйте в программе.

ПРАКТИЧЕСКАЯ РАБОТА № 34

Тема: Математическое моделирование методов шифрования

Цель работы: изучить основные математические основы методов шифрования, научиться моделировать процессы шифрования и расшифровки с помощью математических методов, а также применить полученные знания для решения практических задач.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для шифрования, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Написать программу, использующую алгоритм шифрования данных DES, для преобразования исходного текста. Использовать «зеркальную» перестановку как начальную, одну замену с использованием шифра Гронсфельда и простую перестановку с использованием шифрующей таблицы.

Задание 2. Изменить программу так, чтобы блок замены выполнялся 16 раз.

Задание 3. Написать программу для преобразования текста, зашифрованного с использованием алгоритма шифрования данных DES, в исходный. Порядок работы обратный: перестановка обратная конечной, обратные замены и перестановка обратная начальной.

ПРАКТИЧЕСКАЯ РАБОТА № 35

Тема: Математическое моделирование методов шифрования

Цель работы: изучить основные математические основы методов шифрования, научиться моделировать процессы шифрования и расшифровки с помощью математических методов, а также применить полученные знания для решения практических задач.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для шифрования, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Написать программу для получения хеш-кода заданного по номеру i сообщения, используя упрощённый вариант хеш - функции X.509 с параметрами по номеру j . Вариант задания определяется цифрами номера студенческого билета (i – предпоследняя, j – последняя). Вектор инициализации H_0 следует выбирать самостоятельно.

i	сообщение	i	сообщение
0	ПРЕДЕЛ	5	ЧИСЛИТЕЛЬ
1	ИНТЕГРАЛ	6	АМПЕРСАНТ
2	МИНУС	7	КОРЕНЬ
3	МОДУЛЬ	8	ОСТАТОК
4	ПЛЮС	9	СТЕПЕНЬ

j	параметры	j	параметры
0	(3, 7)	5	(3, 17)
1	(13, 7)	6	(11, 5)
2	(3, 13)	7	(5, 17)
3	(5, 7)	8	(7, 11)
4	(13, 5)	9	(11, 3)

Контрольные вопросы:

1. Что понимается под термином хеш-функция?
2. Сформулируйте основное назначение хеш-функции.
3. Объяснить алгоритм хеширования сообщения на своём примере.

ПРАКТИЧЕСКАЯ РАБОТА № 36

Тема: Математическое моделирование методов шифрования

Цель работы: изучить основные математические основы методов шифрования, научиться моделировать процессы шифрования и расшифровки с помощью математических методов, а также применить полученные знания для решения практических задач.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для шифрования, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Напишите программы для следующих заданий

1. Зашифруйте слово "MATHEMATICS" с ключом $k = 5$
2. Сообщение "RJXXFLJ" зашифровано с ключом $k = 5$. Расшифруйте его.
3. Зашифруйте слово «ЗАГАДКА» ключом $k=5$, а затем зашифруйте полученное слово ключом $k=27$

ПРАКТИЧЕСКАЯ РАБОТА № 37

Тема: Составление программ шифрования методом замены

Цель работы: научиться составлять программы шифрования текста, основываясь на классических методах замены.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для шифрования, MS Word, инструкции по выполнению работы.

Справочный материал:

Шифрование методами замены (подстановки) подразумевает, что символы шифруемого текста заменяются другими символами, взятыми из одного или нескольких алфавитов.

Рассмотрим методы простой замены:

1. Чтобы зашифровать текст вместо первой буквы алфавита писалась последняя, вместо второй – предпоследняя и так далее. Этот древний метод шифрования назывался *атбаиш*.

Пример использования этого шифра можно найти в Библии: «... а царь Сессаха выпьет после них». Такого царя и царства не было. Священные иудейские тексты шифровались простой заменой. Читая по нему слово Сессах, на языке оригинала получаем Вавилон.

Для удобства шифрования таким методом можно алфавит пронумеровать сверху слева на право, а снизу справа на лево. Чтобы зашифровать букву открытого текста, нужно просмотреть порядковый номер буквы сверху и найти букву с таким же номером снизу.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

Например:

Открытый текст	Ф	О	Р	М	А
Шифр	М	Т	Р	Ф	Я

2. *Шифр Юлия Цезаря* – шифр замены, в котором автор записывал весь алфавит по кругу и каждую букву открытого текста для шифровки заменял на четвертую букву по часовой стрелке. Например, вместо А-Д, вместо В-Е и т.д.

Содержание работы:

Задание 1. Используя метод замены шифр Цезаря

Зашифруем методом Цезаря с ключом $k=7$ слово «*шифр*». Написать программу для шифрования и дешифрования.

Будем использовать русский алфавит без буквы ё, где букве А соответствует число 0, а следовательно букве Я – 31. Т.е. $n=32$.

Поставим в исходном слове в соответствие каждой букве число:

ш $\rightarrow 24 = x_1$

и $\rightarrow 8 = x_2$

ф $\rightarrow 20 = x_3$

р $\rightarrow 16 = x_4$

Тогда $y_1 = (x_1 + k) \bmod 32 = (24 + 7) \bmod 32 = 31 \bmod 32 = 31 \rightarrow я$

$y_2 = (x_2 + k) \bmod 32 = (8 + 7) \bmod 32 = 15 \bmod 32 = 15 \rightarrow п$

$$y_3 = (x_3 + k) \bmod 32 = (20 + 7) \bmod 32 = 27 \bmod 32 = 27 \rightarrow \text{ы}$$

$$y_4 = (x_4 + k) \bmod 32 = (16 + 7) \bmod 32 = 23 \bmod 32 = 23 \rightarrow \text{ч}$$

Таким образом, получили слово «япыч»

Дешифрование.

Для дешифрования необходимо каждому символу слова «япыч» поставить в соответствие число:

$$\text{я} \rightarrow 31 = y_1$$

$$\text{п} \rightarrow 15 = y_2$$

$$\text{ы} \rightarrow 27 = y_3$$

$$\text{ч} \rightarrow 23 = y_4$$

$$\text{Тогда } x_1 = (y_1 + (32 - k)) \bmod 32 = (31 + (32 - 7)) \bmod 32 = 56 \bmod 32 = 24 \rightarrow \text{ш}$$

$$x_2 = (y_2 + (32 - k)) \bmod 32 = (15 + 25) \bmod 32 = 40 \bmod 32 = 8 \rightarrow \text{и}$$

$$x_3 = (y_3 + (32 - k)) \bmod 32 = (27 + 25) \bmod 32 = 52 \bmod 32 = 20 \rightarrow \text{ф}$$

$$x_4 = (y_4 + (32 - k)) \bmod 32 = (23 + 25) \bmod 32 = 48 \bmod 32 = 16 \rightarrow \text{р}$$

Получили слово «шифр», следовательно шифрование было выполнено правильно.

Задание 2. Предлагается написать и отладить программы на любом языке программирования для шифрования и дешифрования фразы «Компьютерные сети» шифру Цезаря

- 1) Создайте проект консольного приложения
- 2) Определите алфавит длиной в 32 буквы русского алфавита без «ё», все в верхнем регистре, также создайте переменную с количеством букв.
- 3) Напишите метод `string CaesarEncrypt(string text, int k)`, который:
 - преобразует все буквы в верхний регистр;
 - для каждой буквы находит ее индекс;
 - вычисляет новый индекс с использованием формулы $(\text{index} + k + N) \% N$
 - получает букву с новым индексом;
 - пробелы, цифры и прочее остаются без изменений.
- 4) Напишите метод `string CaesarDecrypt(string text, int k)`, который выполняет обратные действия
- 5) В главном методе запросите у пользователя ключ, зашифруйте строку и выведите результат, также выведите расшифрованный вариант для проверки.

Задание 3. Написать программу шифрования по методу замены – шифр Атбаш

```
Program new;
Uses crt;
Var Alfavit, Alfavit2:string;
A: array[1..100] of integer;
S,S1,S2 :string;
Klyh: string[3];
Shifr: array[1..100] of Char;
Kl: array[1..3] of integer;
```

```

i,l,j,Cod: integer;
Key1:Char;
Begin
clrscr;
Alfavit:='абвгдеёжзийклмнопрстуфхцчшщъыьэюя';
Alfavit2:='яюэыьгщцхфутсрпномлкйизжёедгвба';
S1:='abcdefghijklmnopqrstuvwxyz';
S2:='ABCDEFGHIJKLMNOPQRSTUVWXYZ';
TextColor (4);
Writeln;
Writeln('ПРОГРАММА ШИФРОВАНИЯ ШИФРОМ АТБАШ');
Repeat
TextColor (13);
Writeln;
Writeln;
Writeln('Введите данные, которое хотите зашифровать');
TextColor (14);
Readln(S);
l:=0;
For i:=1 to Length(S) do
For j:=1 to Length(S2) do begin
If (S[i]=S1[j]) or (S[i]=S2[j]) then l:=l+1;
end;
if l<>0 Then
Writeln ('ОШИБКА!');
Writeln('Выражение должно быть написано только строчной кириллицей!');
Until l=0;
For i:=1 to Length(s) do
For j:=1 to 33 do begin
if S[i]=Alfavit[j] then begin
Shifr[i]:=Alfavit2[j];
end;
end;
TextColor(4);
Writeln('Шифровка: ',Shifr,' ');
Readln;
Repeat
TextColor(13);
Writeln('Введите сообщение, которое хотите расшифровать');
TextColor(14);
Readln(S);
l:=0;
For i:=1 to Length(S) do
For j:=1 to Length(S2) do begin
If (S[i]=S1[j]) or (S[i]=S2[j]) then l:=l+1;

```

```

end;
if l<>0 Then
Writeln('Выражение должно быть написано только кириллицей');
Until l=0;
For i:=1 to Length(S) do
For j:=1 to 33 do begin
if Shifr[i]=Alfavit[j] then begin
S[i]:=Alfavit2[j];
end;
end;
TextColor (4);
Writeln ('Расшифрованное выражение:',S,' ');
Readln;
GotoXY(2,24);
Write('ESC - Выход из программы ');
Repeat
Key1:=ReadKey;
case Key1 of
chr(27),chr(27):exit;
else End;
until (Key1=chr(27)) or (Key1=chr(27));
Readln;
end.

```

Результаты выполнения программы шифрования методом Атбаш:

Введите данные, которые хотите зашифровать:

Безопасность

Шифровка:

Юьштсяпутпог

Введите сообщение которое хотите расшифровать

Юьштсяпутпог

Расшифрованное выражение:

Безопасность

Задание 4. Зашифровать словосочетание «Информационная безопасность» шифром Атбаш.

ПРАКТИЧЕСКАЯ РАБОТА № 38

Тема: Составление программ шифрования методом замены

Цель работы: научиться составлять программы шифрования текста, основываясь на классических методах замены.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для шифрования, MS Word, инструкции по выполнению работы.

Справочный материал:

Шифр «Полибианский квадрат». Такое название шифр получил благодаря его изобретателю – греческому историку Полибию. Ключом являлся квадрат размерностью 5×5 , который заполнялся алфавитом в случайном порядке. Для шифрования в квадрате Полибий находил букву текста и вставлял в шифровку нижнюю от нее в том же столбце. Если буква была в нижней строке – то для шифровки он выбирали верхнюю из того же столбца.

Например, для русскоязычного сообщения весь алфавит не уместится в квадрат размерностью 5×5 , поэтому можно взять таблицу любой размерности, причем и размер таблицы, и расположение в нем алфавита будут являться ключами к расшифровке.

п	ц	у	к	е
з	щ	ш	г	н
х	ъ	ф	.	ы
л	о	р	а	в
д	,	ж	э	ю
м	и	т	ь	б
	с	ч	я	!

Открытый текст
Шифр

т а й н а
ч э с ы э

В средние века в торговле применялась и *шифровка цифр*. В качестве ключа необходимо выбрать слово длиной в 10 символов, в котором буквы не повторяются. Каждая буква нумеруется от 0 до 9. При шифровке какого-либо числа все цифры в нем заменяются на буквы из ключевого слова, соответствующие данной цифре.

Например:

Открытый текст	Встречайте 25.06.07
Ключ	РЕСПУБЛИКА
	0 1 2 3 4 5 6 7 8 9
Шифр	Встречайте СБ РЛ РИ

Содержание работы:

Задание 1. Предлагается написать и отладить программы на любом языке программирования для шифрования и дешифрования фразы «*personal computer*» по шифру Полибианский квадрат, размером 5×5 , случайным образом заполненный прописными буквами латинского алфавита.

При этом: все тексты состоят из малых английских букв; текст читается как строка и поэтому не может содержать более 256 символов. Можно не ограничивать объем текста, если учащиеся знакомы с типом файлы.

```
program polib_kvadrat;  
var n,i,r,j,k:integer; s,s1:string; a:array[1..5,1..5] of char;  
begin  
  writeln ('vvedite rasmer kvadrata'); readln(n);  
  writeln ('vvedite bukvi alfavita v kvadrat rasmera',' ',n);  
  for k:=1 to n do begin  
    for j:=1 to n do read(a[k,j]); readln; end;  
  writeln('vvedite stroku'); readln(s);  
  writeln('vvedite regim: 1-shifrovanie, 2-rasshifrovanie');  
  readln(r); s1:=""; {формируемая новая строка}  
  if r=1 then begin  
    i:=1;  
    while i<=length(s) do if s[i]=' ' then delete(s,i,1) else i:=i+1;  
    for i:=1 to length(s) do  
      for k:=1 to n do  
        for j:=1 to n do if s[i]=a[k,j] then  
          if k<n then s1:=s1+a[k+1,j] else s1:=s1+a[1,j];  
        end  
      else  
        for i:=1 to length(s) do  
          for k:=1 to n do  
            for j:=1 to n do if s[i]=a[k,j] then  
              if k>1 then s1:=s1+a[k-1,j] else s1:=s1+a[n,j];  
            writeln (s1);  
          end.
```

Результат

Входные данные (размер квадрата, квадрат с буквами, исходный текст, режим работы)

5

abcde

fghji

klmno

prstu

vwx yz

1

personal computer

Выходные данные (шифртекст)

viwxutfrhusvzyiw

Задание 2. Зашифровать словосочетание «Информационная безопасность» цифровым шифром.

Задание 3. Предлагается написать и отладить программы на любом языке программирования для шифрования и дешифрования фразы «Языки программирования» цифровым шифром.

1) Создайте проект консольного приложения

2) Определите алфавит длиной в 32 буквы русского алфавита без «ё», все в верхнем регистре, также создайте переменную с количеством букв.

3) Напишите метод `string CaesarEncrypt(string text, int k)`, который:

- преобразует все буквы в верхний регистр;

- для каждой буквы находит ее индекс;

- вычисляет новый индекс с использованием формулы $(\text{index} + k + N) \% N$

- получает букву с новым индексом;

- пробелы, цифры и прочее остаются без изменений.

4) Напишите метод `string CaesarDecrypt(string text, int k)`, который выполняет обратные действия

5) В главном методе запросите у пользователя ключ, зашифруйте строку и выведите результат, также выведите расшифрованный вариант для проверки.

ПРАКТИЧЕСКАЯ РАБОТА № 39

Тема: Составление программ шифрования методом замены

Цель работы: научиться составлять программы шифрования текста, основываясь на классических методах замены.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для шифрования, MS Word, инструкции по выполнению работы.

Справочный материал:

Шифр Гронсфельда (простой замены)

АБВГДЕЖЗИКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ

Шифр состоит в модификации шифра Цезаря числовым ключом. Для этого под сообщением необходимо написать ключ. Если ключ короче сообщения, то его повторить циклически. Шифровка получается аналогично шифру Цезаря, но отсчитывать необязательно только четвертую букву по алфавиту, а ту, которая сдвинута на соответствующую цифру ключа.

Например:

Сообщение С О В Е Р Ш Е Н Н О С Е К Р Е Т Н О

Ключ 3 1 4 3 1 4 3 1 4 3 1 4 3 1 4 3 1 4

Шифровка Ф П Ж И С Ъ И О С С А Х И Л Ф И У Т

Шифр Гронсфельда имеет массу модификаций: вроде записи текста шифровки буквами другого алфавита, двойное шифрование разными ключами.

Один из *шифров Трисимуса*, который основывается на шифре «Полибианский квадрат». В таблицу размером, вмещающим весь алфавит, вписывается сначала ключевое слово без повторяющихся букв, а затем весь алфавит по порядку. Пустые клетки можно заполнить любыми знаками. В таблице нужно найти букву открытого текста, а для шифровки взять букву строчкой ниже в этом же столбце. Если буква открытого текста находится в нижней строке – то для шифровки выбирается верхняя из того же столбца.

Например: ключи – размер таблицы 5×6 и слово «Республика»

Р	Е	С	П	У	Б
Л	И	К	А	В	Г
Д	Ж	З	М	Н	О
Т	Ф	Х	Ц	Ч	Ш
Щ	Ы	Ь	Э	Ю	Я

Открытый текст С Л И В А
Шифр К Д Ж Н М

Используя простую замену, можно зашифровать любой текст, однако такой шифр имеет низкую стойкость, так как зашифрованный текст в большинстве случаев имеет те же статистические характеристики, что и исходный. К тому же, если объем зашифрованного текста намного больше, чем одна строка, то частоты появления букв в зашифрованном тексте будут еще ближе к частотам появления букв в алфавите и расшифровка окажется еще проще. Поэтому простую замену используют лишь в тех случаях, когда шифруемый текст короток.

Содержание работы:

Задание 1. Реализовать шифр Гронсфельда замены с ключом в виде числовой последовательности

```

using System;
using System.Text;
class Program {
    // Шифр Гронсфельда
    static string GronsfeldEncrypt(string text, int[] key) {
        string alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        StringBuilder result = new StringBuilder();
        text = text.ToUpper();
        for (int i = 0; i < text.Length; i++) {
            char ch = text[i];
            int index = alphabet.IndexOf(ch);
            if (index >= 0) {
                int shift = key[i % key.Length];
                int newIndex = (index + shift) % alphabet.Length;
                result.Append(alphabet[newIndex]);
            }
            else {
                result.Append(ch); // оставляем символ без изменений
            }
        }
        return result.ToString();
    }
}

```

Задание 2. Реализовать шифр Трисимуса — тройной шифр, основанный на последовательных сдвигах по алфавиту.

```

static string TrisumusEncrypt(string text, int key1, int key2, int key3) {
    string alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    string CaesarShift(string input, int shift) {
        StringBuilder output = new StringBuilder();
        input = input.ToUpper();
        foreach (char ch in input) {
            int index = alphabet.IndexOf(ch);
            if (index >= 0) {
                int newIndex = (index + shift) % alphabet.Length;
                output.Append(alphabet[newIndex]);
            }
            else {
                output.Append(ch);
            }
        }
        return output.ToString();
    }
    string step1 = CaesarShift(text, key1);
    string step2 = CaesarShift(step1, key2);
    string encrypted = CaesarShift(step2, key3);
    return encrypted;
}
static void Main() {
}

```

Задание 3. Предлагается написать и отладить программы на любом языке программирования для шифрования и дешифрования фразы «Я учусь на программиста» с помощью шифра Трисимуса

Задание 4. Предлагается написать и отладить программы на любом языке программирования для шифрования и дешифрования фразы «Внедрение компьютерных систем» с помощью шифра Гронсфельда

ПРАКТИЧЕСКАЯ РАБОТА № 40

Тема: Составление программ шифрования методом замены

Цель работы: научиться составлять программы шифрования текста, основываясь на классических методах замены.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для шифрования, MS Word, инструкции по выполнению работы.

Справочный материал:

Для повышения стойкости шифра используют многоалфавитные замены, в которых для замены символов исходного текста используются символы нескольких алфавитов.

Одной из разновидностей такого метода является схема *шифрования Вижинера*. Шифр очень устойчивый к вскрытию. Таблица Вижинера представляет собой квадратную матрицу с n^2 элементами, где n – число символов используемого алфавита. Каждая строка получена циклическим сдвигом алфавита на один символ

	А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П	Э	Ю	Я	-
А	А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П	Э	Ю	Я	-
Б	-	А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	Ъ	Э	Ю	Я
В	Я	-	А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	Ы	Ъ	Э	Ю
...
Я	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П	Я	-	А	Б
-	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П	Ю	Я	-	А

При шифровании сообщения его выписывают в строку, а под ним буквенный ключ. Если ключ оказался короче сообщения, то его циклически повторяют. Шифровку получают, находя символ в колонке таблицы по букве текста и строке, соответствующей букве ключа.

Например: Сообщение П Р И Е З Ж А Ю Ш Е С Т О Г О

Ключ А Г А В А А Г А В А А Г А В А

Шифровка П Н И Г З Ж Ю Ю Ю А Е О Т М

Предыдущие шифры называются монограммными, так как шифрование ведется по одной букве. Шифрование по 2 букве называются *биграммными*.

Наиболее известный шифр биграммами называется Playfair (использовался в I Мировой войне). Открытый текст разбивается на пары (биграммы). Текст шифруется по следующим правилам:

–Если обе буквы биграммы исходного текста принадлежат одному столбцу таблицы, то буквами шифра считаются буквы, которые лежат под ними. Если буква открытого текста находится в нижнем ряду, то для шифра берется соответствующая буква из верхнего ряда. Биграмма из одной буквы или пары одинаковых букв тоже подчиняются этому правилу.

–Если обе буквы биграммы исходного текста принадлежат одной строке таблицы, то буквами шифра считаются буквы, которые лежат справа от них. Если буква открытого текста находится в правой колонке (в последней), то для шифра берется соответствующая буква из первой колонки.

–Если обе буквы биграммы открытого текста лежат в разных столбцах, то вместо них берутся такие 2 буквы, чтобы вся четверка их представляла

прямоугольник. При этом последовательность букв в шифре должна быть зеркальной исходной паре.

Пример для таблицы 5×6 с ключом «Республика»

Открытый текст ПУ СТ ЪК ОН СУ ЛЫ БУ ДУ ТЬ ДИ ТЕ ЛЬ НЫ

Шифр УБ РХ СЗ ДО ПБ ИЩ РБ НР ШР ЖЛ ФР КЩ ЗЮ

Содержание работы:

Задание 1. Написать и отладить программы на любом языке программирования для шифрования и дешифрования фразы «*personal computer*» по шифру Виженера с ключом «file».

При этом: все тексты состоят из малых английских букв; текст читается как строка и поэтому не может содержать более 256 символов. Можно не ограничивать объем текста, если учащиеся знакомы с типом файлы.

```
program vigener;
var n,i,r:integer; s,s1,s2:string;
begin
writeln('vvedite stroku'); readln(s);
writeln('vvedite kluch'); readln(s1);
writeln('vvedite regim: 1-shifrovanie, 2-rasshifrovanie');
readln(r); s2:="";
for i:=1 to length(s) div length(s1)+1 do {размножаем ключ до длины текста}
s2:=s2+s1;
if r=1 then begin i:=1;
while i<=length(s) do
if s[i]=' ' then delete(s,i,1) else i:=i+1;
for i:=1 to length(s) do begin
n:=ord(s[i])+(ord(s2[i])-ord('a'))+1;
if n>ord('z') then n:=ord('a')+(n-ord('z'))mod 26-1;
s[i]:=chr(n); end; end
else
for i:=1 to length(s) do begin
n:=ord(s[i])-(ord(s2[i])-ord('a'))-1;
if n<ord('a') then n:=ord('z')-(ord('a')-n)mod 26+1;
s[i]:=chr(n); end;
writeln(s);
end.
```

Результат:

Входные данные (исходный текст, ключ) *personal computer*
file

Выходные данные (шифртекст) *vndxiwmtqixuicqw*

Задание 2. Задан некоторый текст зашифрованный шифром Виженера, требуется определить ключевое слово и прочитав открытый текст.

Шифрованный текст

Влцдугтжбюцхъяррмшбрхцэооэцгбрьцмйфктъюьмшэсяцпунуящэйтабэдкцибр
ыцгбрпачкьуцпъбьсэгкцъгуушарцёэвьрюуюоэкаабрняфукабъарпяъафкьиъжяф
фнйо яфывбнэнфуюгбрьсшьжэтбэёчюьрюегофкбъчябашвёэуъюаднчжчужцёэ
влрнчулбюпцуруньшсэюъзкцхъяррнрювяспэмасчкпэужъжыатуфуярюраврту
бурыпэщлафоуфбюацмнубсюкйтабэдийонооэгюоожбгкбрънцэпотчмёодзцвбцщцв
щепчдчдрьюьскасэгъппэгюкдойрсерэвоопчщшоказръббнэугнялёкьсрбёуыз
бдэулбюасшоуэтьшкрсдугэфлбубуъчнчтртпэгюкиугюэмэгюккьпэгяпуфуэзь
радзьжчюрмфцхраююанчёчюыхъьцомэфъцпоирькнщпэтэузуябашушбаыэйчдф
рпэцъь рыцъцпоилуфэдцойэдытррачкубуфнйтабэдкцкрннцюабугюуубу
рьпйюэъжтгюркующюуфъэгясуоичщцдцсфырэдщэуяфшёчцойрщвяхвмк
ршрпгюопэуцйтабэдкцибрыцыяжтюрбуэтэбдуящэубъибрювьежагибргаббрым
пуноцшяжцечкфодшоъчжшйуъчхщвуэбдлдъэгясуахзцэбдэулькнь
щбжяцэърёдъьвювлрнуяфуоухфекьгцччгэъжтанопчынажпачкьуьмэнкйрэфщэъ
ьбудэндадьярьеюэлэтчоубъцэфэвлнёэгфдсэвэёкбсчоукгаутэыпуббцкпэгюч
саъбэнэфъркацхёваетуфяепьрювьржадфёжбъфутощоявьыгупчршуитеачйчи
рамчюфчоуяюонкяжы кгсцбрысшчйотъьжрсщчл

Задание 3. Написать и отладить программу на языке программирования для шифрования и дешифрования по шифру биграмм.

Программа выполняет:

- Создает полный список биграмм из алфавита.
- Создает случайный ключ — перестановку всех биграмм.
- Производит шифрование введенной строки.
- Выполняет дешифрование полученного зашифрованного текста.

```
using System;
using System.Collections.Generic;
using System.Text;
class Program {
    // Сгенерировать список всех возможных биграмм для алфавита
    static List<string> GenerateAllBigrams(string alphabet) {
        var bigrams = new List<string>();
        for (int i = 0; i < alphabet.Длина; i++) {
            for (int j = 0; j < alphabet.Length; j++) {
                bigrams.Add($"{alphabet[i]}{alphabet[j]}");
            }
        }
        return bigrams;
    }
    // Создать ключ — произвольную перестановку биграмм
    static Dictionary<string, string> СоздатьEncryptionKey(List<string> allBigrams) {
        var shuffled = new List<string>(allBigrams);
        var rnd = new Random();
        for (int i = shuffled.Count - 1; i > 0; i--) {
            int j = rnd.Next(i + 1);
            var temp = shuffled[i];
            shuffled[i] = shuffled[j];
            shuffled[j] = temp;
        }
    }
}
```

```

var key = new Dictionary<string, string>();
for (int i = 0; i < allBigrams.Count; i++) {
    key[allBigrams[i]] = shuffled[i]; }
return key; }
// Создать обратный ключ для дешифровки
static Dictionary<string, string> CreateDecryptionKey(Dictionary<string, string>
encryptionKey) {
    var decryptionKey = new Dictionary<string, string>();
    foreach (var pair in encryptionKey) {
        decryptionKey[pair.Value] = pair.Key; }
    return decryptionKey; }
// Разделить текст на биграммы
static List<string> GetBigrams(string text, string alphabet) {
    var bigrams = new List<string>();
    // Удаляем все символы не из алфавита
    var filteredText = "";
    foreach (var ch in text) {
        if (alphabet.Contains(ch)) filteredText += ch; }
    // Если длина нечетная, добавляем 'X' или любой символ
    if (filteredText.Length % 2 != 0)
        filteredText += "X";
    for (int i = 0; i < filteredText.Length; i += 2) {
        bigrams.Add(filteredText.Substring(i, 2)); }
    return bigrams; }
// Шифрование текста
static string Encrypt(string plaintext, Dictionary<string, string> key, string alphabet)
{
    var bigrams = GetBigrams(plaintext, alphabet);
    var encrypted = new StringBuilder();
    foreach (var bigram in bigrams) {
        if (key.ContainsKey(bigram)) encrypted.Append(key[bigram]);
        else encrypted.Append(bigram); // на случай отсутствия, хотя должно быть
    } return encrypted.ToString(); }
// Дешифрование текста
static string Decrypt(string ciphertext, Dictionary<string, string> decryptionKey,
string alphabet) {
    var bigrams = new List<string>();
    for (int i = 0; i < ciphertext.Length; i += 2) {
        bigrams.Add(ciphertext.Substring(i, 2)); }
    var decrypted = new StringBuilder();
    foreach (var bigram in bigrams) {
        if (decryptionKey.ContainsKey(bigram))
            decrypted.Append(decryptionKey[bigram]);
        else decrypted.Append(bigram); }
    return decrypted.ToString(); }

```

```

static void Main() {
    string alphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"; // алфавит
    Console.WriteLine("Введите сообщение для шифрования:");
    string message = Console.ReadLine().ToUpper();
    // Генерация всех биграмм
    var allBigrams = GenerateAllBigrams(alphabet);
    // Создание ключа
    var encryptionKey = CreateEncryptionKey(allBigrams);
    var decryptionKey = CreateDecryptionKey(encryptionKey);
    // Шифрование
    string encrypted = Encrypt(message, encryptionKey, alphabet);
    Console.WriteLine($"Зашифрованное сообщение: {encrypted}");
    // Дешифрование
    string decrypted = Decrypt(encrypted, decryptionKey, alphabet);
    Console.WriteLine($"Расшифрованное сообщение: {decrypted}");
} }

```

Задание 4. Написать и отладить программу на языке программирования для шифрования и дешифрования по шифру биграмм фразу «Изучайте язык программирования»

ПРАКТИЧЕСКАЯ РАБОТА № 41

Тема: Составление программ шифрования методом перестановки

Цель работы: научиться составлять программы шифрования текста, основываясь на классических методах перестановки

Оборудование: ПК, интернет, программное обеспечение – программные продукты для шифрования, MS Word, инструкции по выполнению работы.

Справочный материал:

Шифрование методами перестановки подразумевает, что символы шифруемого текста внутри шифруемого блока символов переставляются по определенным правилам. Наиболее часто встречаются в автоматизированных системах следующие разновидности этого метода.

1. Самая простая перестановка —это *обратное написание* текста так, что прочесть его можно лишь в отражении зеркала. Для усложнения шифрограмму можно одновременно разбить на пятерки букв.

Например:

Открытый текст ПУСТЬ БУДЕТ ТАК, КАК МЫ ХОТЕЛИ

Шифр I ИЛЕТО ХЫМКА ККАТТ ЕДУБЬ ТСУП.

В последней группе (пятерке) не хватает одной буквы. Значит, прежде чем шифровать исходное выражение, следует его дополнить незначащей буквой (например, О) до числа, кратного пяти:

Шифр II ОИЛЕТ ОХЫМК АККАТ ТЕДУБ ЪТСУП.

Кажется, ничего сложного, но при расшифровке появятся серьезные неудобства.

2. Другую разновидность этого метода можно представить как метод перестановки по таблице. Таблицами для перестановки стали пользоваться в Эпоху Возрождения. Ключом в них служат размер таблицы, фраза, задающая перестановку или специальная особенность таблиц

а) *Простая перестановка по таблице:* для шифровки выбрать размер таблицы (в зависимости от длины фразы), вписать открытый текст по столбцам, тогда шифровка получается при считывании по строкам, для усложнения шифротекст можно разбить на любое одинаковое количество букв.

Например:

Открытый текст Неясное становится еще более непонятным

Ключ размер таблицы 5×7, выписывание шифра по строкам

Шифр I НОНСБ НЯЕЕО ЯОЕТЯ СВЕЛП НСТИЩ ЕОЫНА ТЕЕНМ

Н	О	Н	С	Б	Н	Я
Е	Е	О	Я	О	Е	Т
Я	С	В	Е	Л	П	Н
С	Т	И	Щ	Е	О	Ы
Н	А	Т	Е	Е	Н	М

Выписывать шифротекст из таблицы можно по строкам, столбцам, прямой или обратной спирали, диагоналям, то есть шифровать и дешифровать можно в различных направлениях. Например, по диагонали:

Шифр II Н ОЕ НЕЯ СОСС БЯВТН НОЕИА ЯЕЛЩТ ТПЕЕ НОЕ ЫН М

б) *Одиночная перестановка по ключу.* Аналогично шифрованию простой перестановки открытый текст вписывается в таблицу. Далее либо столбцы, либо строки можно беспорядочно пронумеровать цифрами или буквами ключевого слова. Затем переставить эти столбцы (строки) поменять местами в соответствии с номером (буквой) по порядку. Шифротекст можно выписать из получившейся таблицы по любому направлению.

Например:

Открытый текст Неясное становится еще более непонятным

Ключи размер таблицы 5×7, перестановка столбцов– цифры 4756213 или ключевое слово «Лунатик», выписывание шифра по строкам

Шифр НБЯНН СОЕОТ ЕОЯЕП ЛПЯВЕ СОЕЫС ИЩТНЕ МНТЕА

Л	У	Н	А	Т	И	К
4	7	5	6	2	1	3
Н	О	Н	С	Б	Н	Я
Е	Е	О	Я	О	Е	Т
Я	С	В	Е	Л	П	Н
С	Т	И	Щ	Е	О	Ы
Н	А	Т	Е	Е	Н	М

до перестановки

И	Т	К	Л	Н	А	У
1	2	3	4	5	6	7
Н	Б	Я	Н	Н	С	О
Е	О	Т	Е	О	Я	Е
П	Л	П	Я	В	Е	С
О	Е	Ы	С	И	Щ	Т
Н	Е	М	Н	Т	Е	А

после перестановки

в) *Одиночная перестановка символов с пропусками.* По аналогии с шифром одиночной перестановки фраза вписывается в таблицу, столбцы пронумеровываются не по порядку. Из этой таблицы составляется новая, в которой расставляются пропуски в соответствии с полученным цифровым ключом. Далее столбцы переставляются по порядку и выписывается шифровка в любом направлении.

Например:

Открытый текст Неясное становится еще более непонятным

Ключи размер таблицы 5×7, перестановка столбцов– цифры 4756213, выписывание шифра по строкам

Шифр СБННН ОО ОЯЕЯЕСВ ЕЯ ТНТИЕПСЛ ЫН ЕОЩЕНМ ТЕА

4	7	5	6	2	1	3
Н	О	Н	С	Б	Н	Я
Е	Е	О	Я	О	Е	Т
Я	С	В	Е	Л	П	Н
С	Т	И	Щ	Е	О	Ы
Н	А	Т	Е	Е	Н	М

До перестановки и пропусков

4	7	5	1	6	2	3
Н	О	Н	С	=	Б	Н
Я	Е	Е	О	Я	=	О
Е	Т	Я	С	=	В	=
Е	Л	П	Н	С	Т	И
=	Щ	Е	=	О	Ы	Н
=	А	Т	Е	Е	Н	М

С пропусками до перестановки

1	2	3	4	5	6	7
С	Б	Н	Н	Н	=	О
О	=	О	Я	Е	Я	Е
С	В	=	Е	Я	=	Т
Н	Т	И	Е	П	С	Л
=	Ы	Н	=	Е	О	Щ
Е	Н	М	=	Т	Е	А

После перестановки

г) *Двойные перестановки столбцов и строк.* Аналогично шифру одиночной перестановки открытый текст вписывается в таблицу, и строки, и столбцы пронумеровываются не по порядку, а затем в соответствии с номерами переставляются и столбцы, и строки. Шифровка выписывается из таблицы в любом направлении.

Например:

Открытый текст Приезжаю шестого

Ключи размер таблицы 4×4, перестановка для строк – цифры 4213, перестановка для столбцов – 3124, выписывание шифра по строкам

Шифр ШЕ СЖАЗЮОГТОРИПЕ

	3	1	2	4
4	П	Р	И	Е
2	З	Ж	А	Ю
1		Ш	Е	С
3	Т	О	Г	О

до перестановки

	1	2	3	4
4	Р	И	П	Е
2	Ж	А	З	Ю
1	Ш	Е		С
3	О	Г	Т	О

перестановка столбцов

	1	2	3	4
1	Ш	Е		С
2	Ж	А	З	Ю
3	О	Г	Т	О
4	Р	И	П	Е

перестановка строк

Содержание работы:

Задание 1. Написать программу для шифрования «Обратное написание текста»

```
program zadomnapered;
uses crt;
var s,srez:string;
i,tmp:integer;
begin clrscr;
srez:="";
writeln('Введите слово:');
readln(s);
tmp:=length(s);
for i:=1 to tmp do srez[i]:=s[tmp-i+1];
writeln('перевернутое слово:');
for i:=1 to tmp do write(srez[i]);
readln;
end.
```

Результат выполнения программы

Введите слово: Привет

Перевернутое слово: Тевирп

Задание 2. Написать и отладить программу на языке программирования для шифрования и дешифрования фразы, зашифрованную столбцовой перестановкой:

- ОКЕСНВРП_ЫРЕАДЕЫН_В_РСИКО;
- ДСЛИЕЗТЕА_Ь_ЛЬЮВМИ_АОЧХК;
- НМВИАИ_НЕВЕ_СМСТУОРДИАНКМ;
- ЕДСЗЬНДЕ_МУБД_УЭ_КРЗЕМНАЫ;
- СОНРЧОУО_ХДТ_ИЕИ_ВЗКАТРРИ.

ПРАКТИЧЕСКАЯ РАБОТА № 42

Тема: Составление программ шифрования методом перестановки

Цель работы: научиться составлять программы шифрования текста, основываясь на классических методах перестановки

Оборудование: ПК, интернет, программное обеспечение – программные продукты для шифрования, MS Word, инструкции по выполнению работы.

Справочный материал:

1. *Шифр «Магический квадрат».* Магический квадрат – это квадратная таблица с вписанными в клетки последовательными натуральными числами от 1, которые дают в сумме по каждому столбцу, каждой строке и каждой диагонали одно и тоже число.

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

Число вариантов таких квадратов:

$$3 \times 3 = 1$$

$$4 \times 4 = 880$$

$$5 \times 5 = 250000 \text{ и т.д.}$$

Чтобы зашифровать открытый текст с помощью такого квадрата, нужно пронумеровать все буквы в фразе по порядку и вставить их в квадрат вместо соответствующих цифр.

Например:

Открытый текст П Р И Е З Ж А Ю Ш Е С Т О Г О _ 1 2 3 4 5 6 7 8 9
10 11 12 13 14 15 16

Ключ

вышеприведенный квадрат размером 4×4

Шифр

ОИРТЗШЕЮ ЖАСЕГОП

О	И	Р	Т
З	Ш	Е	Ю
	Ж	А	С
Е	Г	О	П

2. Увлечение теорией магических квадратов привело к открытию нового класса шифров перестановок, названных *решетками* или *трафаретами*.

Они представляют собой квадратные таблицы, где четверть ячеек прорезана так, что при четырех поворотах они покрывают, весь квадрат.

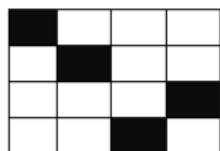
Число вариантов решеток:

$$2 \times 2 = 1$$

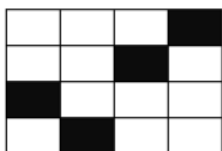
$$4 \times 4 = 256$$

$$6 \times 6 \text{ свыше } 100 \text{ тыс. и т.д.}$$

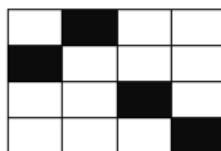
Шифрование осуществляется вписыванием в прорезанные ячейки текста с поворотами решетки, которые продолжаютя до тех пор, пока весь квадрат не будет заполнен.



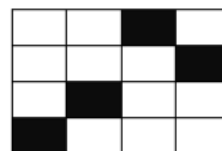
0°



90°



180°



270°

Например:

Открытый текст

ПРИЕЗЖАЮ ШЕСТОГО

Ключ

решетка, представленная выше

Шифр

ЗТП ОЖШРЕИГАЕСЮО

Несмотря на кажущуюся сложность, шифры типа решеток довольно просто вскрываются, и не должны использоваться в виде самостоятельного шифра. Однако они очень удобны и долго использовались в практике для усиления шифров замены.

Содержание работы:

Задание 1. Создать программу, которая: определяет магический квадрат (например, 3x3 или 4x4), шифрует текст, заменяя каждую букву на пару чисел (координаты), дешифрует текст по этим координатам.

```
using System;
using System.Collections.Generic;
using System.Text;
class MagicSquareCipher {
    static char[,] magicSquare;
    static int size;
    // Создаёт магический квадрат из символов
    static void CreateMagicSquare(char[] symbols) {
        size = (int)Math.Sqrt(symbols.Length);
        magicSquare = new char[size, size];
        int index = 0;
        for (int i = 0; i < size; i++) {
            for (int j = 0; j < size; j++) {
                magicSquare[i, j] = symbols[index++];
            }
        }
    }
    // Находит координаты символа в магическом квадрате
    static (int row, int col) FindPosition(char c) {
        for (int i = 0; i < size; i++) {
            for (int j = 0; j < size; j++) {
                if (magicSquare[i, j] == c)
                    return (i, j);
            }
        }
        // Если символ не найден, можно вернуть (-1, -1)
        return (-1, -1);
    }
    // Шифрование: заменяет символы на координаты
    static string Encrypt(string input) {
        var sb = new StringBuilder();
        foreach (char c in input) {
            var (row, col) = FindPosition(c);
            if (row == -1 || col == -1) {
                // Если символ не в квадрате, пропускаем или добавляем как есть
                Console.WriteLine($"Символ '{c}' не найден в магическом квадрате.");
                continue;
            }
            sb.Append($"{row} {col} "); // можно разделять пробелом
        }
    }
}
```



```

    }
    return sb.ToString(); }
// Дешифрование: по координатам восстанавливаем символы
static string Decrypt(string encrypted) {
    var result = new StringBuilder();
    var parts = encrypted.Split(new char[] { ' ' },
StringSplitOptions.RemoveEmptyEntries);
    foreach (var part in parts) {
        if (part.Length != 2)
            continue;
        int row = int.Parse(part[0].ToString());
        int col = int.Parse(part[1].ToString());
        if (row >= 0 && row < size && col >= 0 && col < size) {
            result.Append(magicSquare[row, col]);
        }
        else {
            Console.WriteLine($"Координаты {part} вне диапазона.");
        }
    }
    return result.ToString(); }
static void Main() {
    // Определим магический квадрат
    // Например, для алфавита A-Z (26 символов) создадим 5x5 (25
символов), добавим что-то еще
    // Или возьмем 4x4 (16 символов), например, цифры или буквы
    // В этом примере возьмем алфавит A-Z и добавим символы для
полного покрытия
    string symbolsString =
"ABCDEFGHIJKLMNOPQRSTUVWXYZ012345";
    char[] symbols = symbolsString.ToCharArray();
    CreateMagicSquare(symbols);
    Console.WriteLine("Магический квадрат:");
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            Console.Write($"{magicSquare[i, j]} ");
        }
        Console.WriteLine();
    }
    Console.WriteLine("Введите сообщение для шифрования:");
    string message = Console.ReadLine().ToUpper();
    string encrypted = Encrypt(message);
    Console.WriteLine($"Зашифрованное сообщение: {encrypted}");
    string decrypted = Decrypt(encrypted);
    Console.WriteLine($"Расшифрованное сообщение: {decrypted}");
} }

```

Как работает программа:

- Вы задаёте символы магического квадрата.
- Для каждого символа сообщения ищется его позиция (строка и столбец).
- При шифровании — выводятся координаты.
- При дешифровании — по координатам восстанавливается символ

Задание 2. Нужно зашифровать фразу «ПРИМЕРМАРШРУТНОЙПЕРЕСТАНОВКИ», используя прямоугольник размера 4×7 . Исходное сообщение вписывают в таблицу по горизонтали, начиная с левого верхнего угла, поочерёдно слева направо и справа налево. Выписывают сообщение по другому маршруту: по вертикали, начиная с верхнего правого угла и двигаясь поочерёдно сверху вниз и снизу вверх. В результате зашифрованная фраза будет выглядеть так: «МАСТАЕРРЕШРНОЕРМИУПВКЙТРПНОЙ»

Задание 3: Напишите программу на языке программирования, которая шифрует сообщение методом перестановки. Входные данные: строка текста и ключ — порядок индексов символов. Например, для ключа $[2, 0, 1]$ и строки "abc" шифрование должно вернуть "cab".

ПРАКТИЧЕСКАЯ РАБОТА № 43

Тема: Составление программ шифрования методом перестановки

Цель работы: научиться составлять программы шифрования текста, основываясь на классических методах перестановки

Оборудование: ПК, интернет, программное обеспечение – программные продукты для шифрования, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Написать и отладить программу на языке программирования для шифрования и дешифрования фразы зашифрованную двойной перестановкой (сначала были переставлены столбцы, затем строки):

- a) СЯСЕ_ _ЛУНЫИАККННОГЯДУЧАТН;
- b) МСЕЫ_ЛЫВЕНТОСАНТУЕИ_РЛПОБ;
- c) АМНРИД_УЕБСЫ_ЕЙРСООКОТНВ_;
- d) ОПЧУЛС_БОУНЕВ_ОЖАЕОНЕЩЕИН;
- e) ЕШИАНИРЛПГЕЧАВРВ_СЕЫНА_ЛО.

Задание 2. Дано сообщение и ключ, представленный в виде строки чисел через запятую, например: "2,0,1". Реализуйте функцию, которая преобразует сообщение, переставляя символы в соответствии с заданным ключом, и возвращает зашифрованное сообщение.

ПРАКТИЧЕСКАЯ РАБОТА № 44

Тема: Составление программ шифрования методом перестановки

Цель работы: научиться составлять программы шифрования текста, основываясь на классических методах перестановки

Оборудование: ПК, интернет, программное обеспечение – программные продукты для шифрования, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Создайте программу, которая принимает на вход строку и ключ, разбивает строку на блоки по длине ключа и для каждого блока применяет перестановку символов в соответствии с ключом. Если последний блок короче ключа, заполните его пробелами.

Задание 2. Реализуйте функцию расшифровки, которая восстанавливает исходное сообщение из зашифрованного с помощью того же ключа. Проверьте работу функции на примерах.

Задание 3. Модифицируйте программу так, чтобы она могла шифровать и расшифровывать сообщения методом перестановки, принимая от пользователя вводные данные для сообщения и ключа.

ПРАКТИЧЕСКАЯ РАБОТА № 45

Тема: Составление программ шифрования через картинку

Цель работы: изучить методы шифрования информации с использованием изображений, научиться создавать программы для внедрения шифрования данных в изображения и извлечения зашифрованной информации, а также понять принципы стеганографии и криптографии.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для шифрования, среда программирования, MS Word, инструкции по выполнению работы.

Справочный материал:

Стеганография — метод скрытия информации в носителе (например, изображении), так, чтобы скрытая информация была невидимой для посторонних.

LSB (Least Significant Bit) — наименее значимый бит в байте, который можно использовать для внедрения данных с минимальным визуальным воздействием.

Шифрование — процесс преобразования открытой информации в зашифрованный вид с помощью криптографических алгоритмов, обеспечивающих конфиденциальность.

Внедрение данных — процесс вставки зашифрованных данных в изображение.

Извлечение данных — восстановление скрытой информации из изображения.

Основные методы:

- Внедрение данных в младшие биты пикселей изображения (LSB).
- Шифрование данных перед внедрением для повышения безопасности.
- Использование бинарных файлов и текста.
- Восстановление данных после внедрения и шифрования.

Содержание работы:

Задание 1. Реализовать функцию для внедрения текста в изображение через изменение младших битов пикселей.

```
from PIL import Image
def hide_text_in_image(image_path, text, output_path):
    img = Image.open(image_path)
    binary_text = ''.join([format(ord(c), '08b') for c in text])
    pixels = list(img.getdata())
    # Вставляем биты текста в младшие биты каждого пикселя
    new_pixels = []
    bit_idx = 0
    for pixel in pixels:
        r, g, b = pixel
        if bit_idx < len(binary_text):
            r = (r & 0xFE) | int(binary_text[bit_idx])
            bit_idx += 1
        if bit_idx < len(binary_text):
```

```

    g = (g & 0xFE) | int(binary_text[bit_idx])
    bit_idx += 1
if bit_idx < len(binary_text):
    b = (b & 0xFE) | int(binary_text[bit_idx])
    bit_idx += 1
new_pixels.append((r, g, b))
img.putdata(new_pixels)
img.save(output_path)

```

Задание 2. Реализовать функцию для извлечения текста из изображения, скрытого через LSB.

```

def extract_text_from_image(image_path, text_length):
    img = Image.open(image_path)
    pixels = list(img.getdata())
    bits = ""
    for pixel in pixels:
        for color in pixel[:3]:
            bits += str(color & 1)
            if len(bits) >= text_length * 8:
                break
    if len(bits) >= text_length * 8:
        break
    chars = [chr(int(bits[i:i+8], 2)) for i in range(0, len(bits), 8)]
    return "".join(chars)

```

Задание 3. Создать документацию и руководство пользователя. Описать последовательность работы программы, требования к входным данным и инструкцию по использованию.

ПРАКТИЧЕСКАЯ РАБОТА № 46

Тема: Составление программ шифрования через картинку

Цель работы: изучить методы шифрования информации с использованием изображений, научиться создавать программы для внедрения шифрования данных в изображения и извлечения зашифрованной информации, а также понять принципы стеганографии и криптографии.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для шифрования, среда программирования, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Изменить функцию для внедрения бинарных данных (например, файла).

```
def hide_binary_in_image(image_path, binary_data, output_path):
    img = Image.open(image_path)
    pixels = list(img.getdata())
    bits = ".join([format(byte, '08b') for byte in binary_data])
    new_pixels = []
    bit_idx = 0
    for pixel in pixels:
        r, g, b = pixel
        if bit_idx < len(bits):
            r = (r & 0xFE) | int(bits[bit_idx])
            bit_idx += 1
        if bit_idx < len(bits):
            g = (g & 0xFE) | int(bits[bit_idx])
            bit_idx += 1
        if bit_idx < len(bits):
            b = (b & 0xFE) | int(bits[bit_idx])
            bit_idx += 1
        new_pixels.append((r, g, b))
    img.putdata(new_pixels)
    img.save(output_path)
```

Задание 2. Реализовать извлечение бинарных данных из изображения.

```
def extract_binary_from_image(image_path, data_length_bytes):
    img = Image.open(image_path)
    pixels = list(img.getdata())
    total_bits = data_length_bytes * 8
    bits = ""
    for pixel in pixels:
        for color in pixel[:3]:
            bits += str(color & 1)
            if len(bits) >= total_bits:
                break
    if len(bits) >= total_bits:
```

```
        break
binary_data = bytearray()
for i in range(0, len(bits), 8):
    byte = bits[i:i+8]
    binary_data.append(int(byte, 2))
return bytes(binary_data)
```

Задание 3. Создать документацию и руководство пользователя. Описать последовательность работы программы, требования к входным данным и инструкцию по использованию.

ПРАКТИЧЕСКАЯ РАБОТА № 47

Тема: Составление программ шифрования через картинку

Цель работы: изучить методы шифрования информации с использованием изображений, научиться создавать программы для внедрения шифрования данных в изображения и извлечения зашифрованной информации, а также понять принципы стеганографии и криптографии.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для шифрования, среда программирования, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Приведите примеры устройств, используемых для генерации одноразовых паролей. Опишите алгоритм генерации одноразовых паролей.

Задание 2. Проведите сравнительный анализ программ, используемых для создания цифровых водяных знаков: PhotoWatermark Professional, Image Tuner, EasyWatermark, CryptoFoto.

Задание 3. Опишите процесс создания печатного водяного знака в программе Image Tuner.

ПРАКТИЧЕСКАЯ РАБОТА № 48

Тема: Составление программ шифрования через картинку

Цель работы: изучить методы шифрования информации с использованием изображений, научиться создавать программы для внедрения шифрования данных в изображения и извлечения зашифрованной информации, а также понять принципы стеганографии и криптографии.

Оборудование: ПК, интернет, программное обеспечение – программные продукты для шифрования, среда программирования, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Рассмотреть работу двух программ, позволяющих проводить стеганографические преобразования.

Задание 2. Выбрать контейнер и выполнить внедрение в него некоторой информации.

Задание 3. Попробовать извлечь информацию из стегоконтейнера, созданного другой программой.

Задание 4. От чего зависит криптостойкость стеганографических систем?

ПРАКТИЧЕСКАЯ РАБОТА № 49

Тема: Установка и настройка антивирусных программ

Цель работы: научиться устанавливать и настраивать антивирусную программу на ПК, настраивать обновления с помощью зеркала.

Оборудование: ПК, интернет, программное обеспечение – антивирусная программа, инструкции по выполнению работы

Порядок выполнения работы:

Задание 1. Установить и настроить антивирусную программу на выбор Avast, DrWed, Nod32

Задание 2. Настроить обновления антивирусной программы с помощью зеркала.

Зеркало обновлений – это компьютер, настроенный как источник обновлений для антивируса на других ПК в локальной сети. Используется в сетях, где не все ПК имеют доступ в Интернет.

Для создания зеркала обновлений:

1. Откройте меню антивируса, щёлкнув по значку в трее и выбрав *Центр безопасности*;
2. Щёлкните по значку замка, разрешите внесение изменений и нажмите кнопку с шестерёнкой (*Настройки*)
3. Перейдите на вкладку *Обновление* и выберите *Дополнительные настройки*.
4. Установите переключатель *Зеркало обновлений* в положение *Вкл*.
Открывается окно настройки зеркала:

Зеркало обновлений

Путь к папке, в которую копируются обновления

C:\Program Files

Обзор...

Адрес: 0.0.0.0

Порт: 8080

ОК Отменить

5. Нажмите *Обзор* и укажите папку, куда будут сохраняться обновления. Она будет источником обновления для остальных ПК в локальной сети.
6. *Адрес* – адрес данного ПК в локальной сети, который будет использоваться как адрес для обновления через антивирусную сеть другими ПК в этой сети.
Порт – номер порта, который будет использоваться для раздачи обновлений.

ПРАКТИЧЕСКАЯ РАБОТА № 50

Тема: Обнаружение вируса и устранение последствий его влияния

Цель работы: изучение методов обнаружения вирусов и методов удаления последствий заражения вирусами с использованием антивирусной утилиты

Оборудование: ПК, интернет, программное обеспечение – антивирусные программы, MS Word, инструкции по выполнению работы.

Справочный материал:

Методы обнаружения вирусов: сканирование, обнаружение изменений, эвристический анализ, использование резидентных сторожей, вакцинирование программ, аппаратно-программная защита от вирусов.

Методы удаления последствий заражения вирусам:

1. предполагает восстановление системы после воздействия известных вирусов (разработчики программы-фага, удаляющей вирус, должен знать структуру вируса и его характеристики размещения в среде обитания);

2. позволяет восстанавливать файлы и загрузочные сектора, зараженные неизвестными вирусами (для восстановления файлов программа восстановления должна заблаговременно создать и хранить информацию о файлах, полученную в условиях отсутствия вирусов.

Содержание работы:

Задание 1. Изучить категории вредоносных программ и изучить работу с антивирусной утилитой. Заполнить таблицу с описанием вирусов.

Категории вредоносных программ	Наименование и описание вируса	Видимые проявления
Adware и SpyWare		
Backdoor		
Hoax		
Trojan		
Trojan-Clicke		
Trojan-Downloader		
Trojan-Spy		
Trojan-PSW		
Net-Worm		
Worm		
Trojan-Dropper		
Trojan-Proxy		
Email-Worm		
FraudToo		
Trojan-Ransom		

Задание 2. Провести сравнительный анализ 4 антивирусных программ по 10 критериям. Критерии определить и сформулировать самостоятельно (например, обнаружение вирусов, обнаружение деструктивных кодов,

готовность быстрого реагирования на появление новых видов угроз и др.). Анализ оформить в виде таблицы:

Критерий сравнения	Антивирусная программа 1	Антивирусная программа 2	Антивирусная программа 3	Антивирусная программа 4
1.	+	-	-	+
2.	-	+	+	-
...				
10.				

Задание 2 должно заканчиваться обобщающим выводом. Указать, какому антивирусному средству вы отдадите предпочтение, применяете ли вы его возможности? Почему? Ответ обосновать.

Задание 3. Сравнить системные требования Антивируса Касперского с конфигурацией Вашего компьютера и убедиться, что установка этого приложения возможна. Заполните следующую таблицу:

Параметр	Требования антивируса Касперского	Параметры системы
Процессор		
Оперативная память		
Свободное место на диске		
Браузер		

Задание 4. С помощью антивирусной программы проверить наличие вирусов на компьютере и флеш-накопителе.

Контрольные вопросы:

1. В чем различие программ-фагов и программ-ревизоров?
2. В чем заключаются достоинства программ-ревизоров и программ-вакцин?
3. Перечислите меры защиты от компьютерных вирусов.
4. Что такое эвристический анализ антивирусной программы?
5. Какая антивирусная программа работает в режиме «избыточного» сканирования и в чем его смысл?

ПРАКТИЧЕСКАЯ РАБОТА № 51

Тема: Использование программ дефрагментации дисков

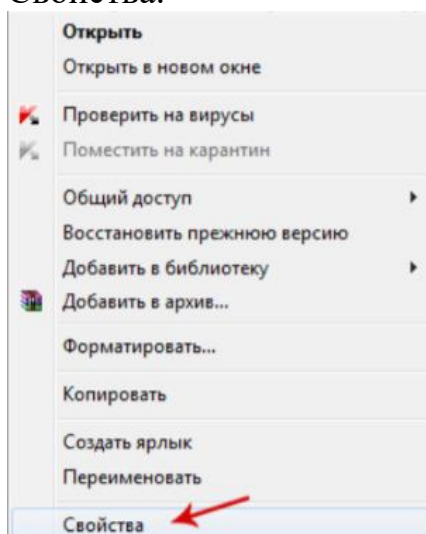
Цель работы: приобрести практический опыт по дефрагментации диска

Оборудование: ПК, интернет, программное обеспечение – операционная система Windows 10, программы для дефрагментации дисков, MS Word, инструкции по выполнению работы.

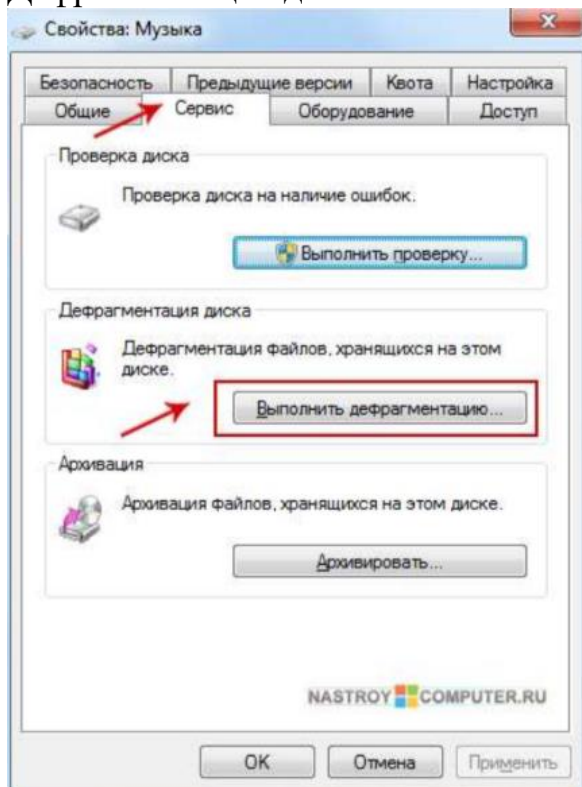
Содержание работы:

Задание 1. Провести дефрагментацию диска

1. Для начала заходим в Мой компьютер и щелкаем правой кнопкой по тому жесткому диску, для которого хотим провести дефрагментацию и нажимаем Свойства.



2. В открывшемся окне переходим на вкладку Сервис и в блоке Дефрагментация диска нажимаем на кнопку Выполнить дефрагментацию.

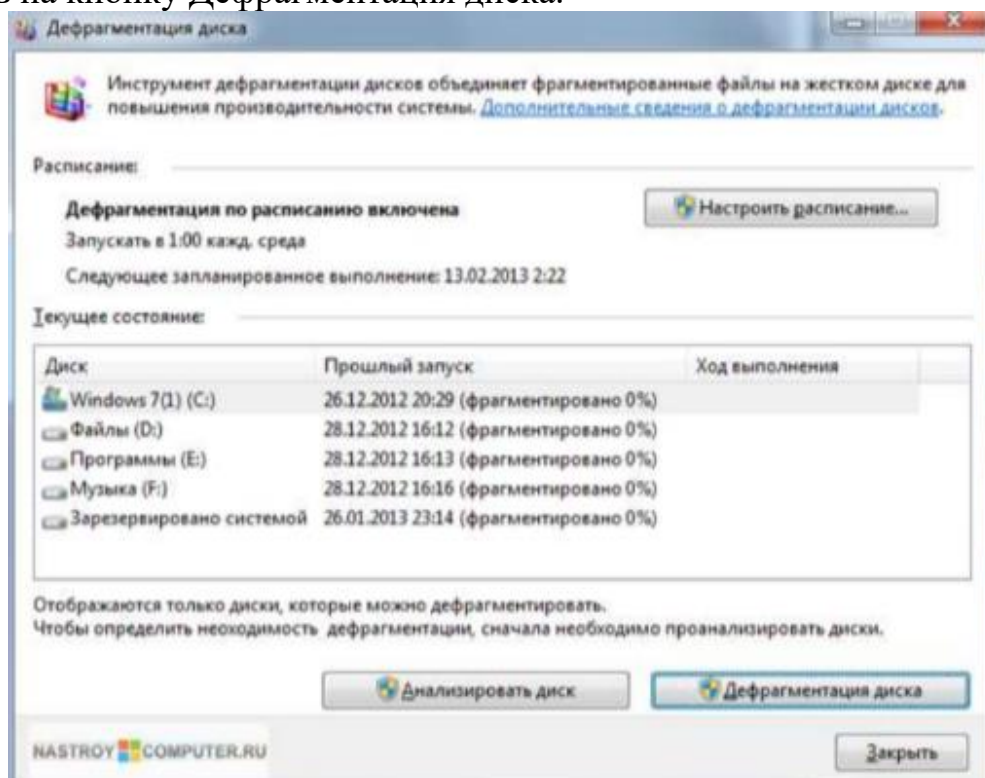


3. В новом окне в блоке Расписание вы можете настроить расписание, в которое будет проводиться дефрагментация. К настройке расписания жесткого диска относятся:

- Периодичность (ежедневно, еженедельно, ежемесячно).
- День (Любой день недели).
- Время (от 00:00 до 23:00).
- Диски (Выбор диска, отметить галочкой один или несколько).

Перед тем как приступить к дефрагментации, сначала рекомендуется проанализировать жесткий диск. Чтобы выполнить эту процедуру нажмите Анализировать диск.

После завершения анализа можно приступить к процессу фрагментации нажав на кнопку Дефрагментация диска.



Если в окне напротив жесткого диска написано фрагментировано 0%, значит с жестким диском все в порядке и проводить дефрагментацию не нужно.

Задание 2. Дефрагментация дискового пространства. Произвести дефрагментацию жесткого диска, не перемещая файлы программ в начало диска.

1. Произведите запуск программы Defrag.
2. Выберите кнопку Настройка.
3. В окне Настройка дефрагментации снимите флажок Переместить файлы программ для ускорения и запуска.
4. В этом же окне установите флажок Использовать эти параметры только один раз.
5. Нажмите кнопку <ОК>.

ПРАКТИЧЕСКАЯ РАБОТА № 52

Тема: Работа с программой восстановления файлов и очистки дисков

Цель работы: изучить программы по восстановлению файлов, научиться использовать специализированные программы для сервисного обслуживания компьютера и операционной системы.

Оборудование: ПК, интернет, программное обеспечение – операционная система Windows 10, программы для восстановления файлов и очистки дисков, MS Word, инструкции по выполнению работы.

Справочный материал:

Все программы по восстановлению утерянных файлов работают благодаря особенности записи данных на ПК. Физический ресурс памяти компьютера поделен на логические части. Пользователь видит их в виде «диск С», «диск D» и далее. Каждый имеет свой шаблон записи или файловую систему. На одном компьютере может работать несколько систем сохраняющих контент каждая по-своему.

Невидимыми для пользователя являются метаданные. Они располагаются на том же уровне, что и логические диски и хранят сведения о структуре расположения папок и документов, их имена, тип файловой системы.

Эта служебная информация необходима компьютеру для лучшей организации работы и не показывается на экране. Каждый логический диск тоже имеет скрытые сведения и копии этих сведений.

Технические сведения полезны т.к. содержат точный физический адрес файла. Утилита обращается к этим записям и восстанавливает материал. Если метаданные повреждены, утилита обращается к одной из копий.

Когда происходит визуальное удаление материала, на самом деле изменяется только техническая информация о файле или папке. Если не произошла на его место перезапись нового файла, первоначальная информация по-прежнему содержится в компьютере.

Поиск по техническим сведениям помогает воссоздать содержимое в исходном виде.

Если произошло разрушение метаданных, то утилита восстановит материал, но содержаться он будет не под своим, а под виртуальным именем (набор цифр и букв).

Еще один принцип работы заключается в обращении к сигнатуре. Это сведения о расширении. Например, запись о музыке в формате MP3 начинается с обозначения ID3... Программа ищет все известные расширения и реанимирует их по категориям: музыка, фото, Pdf — документы, txt, и так далее.

Недостаток этого метода в отсутствии первоначальной структуры. Все «возвращенное» будет в хаотичном порядке и под чужим (произвольным) именем. Но этот способ позволяет восстановить гораздо больше информации, чем первый.

Восстановление файлов при повреждении жесткого диска

Чаще всего пользователи хранят важные сведения на жестком диске. Но если он повреждается, то это влечет частичную потерю информации, которую

можно восстановить при помощи софта. Но это подходит только в случае целостности дисков.

Когда жесткий диск поврежден механически — разорваны дорожки, или аппаратно — не определяется компьютером, то лучше обратиться в специальные лаборатории.

Важное предостережение: перед началом восстановления информации с жесткого диска следует создать его образ.

Клонировать содержимое диска можно при помощи этих же программ. Это защитит первоисточник от перезаписи и позволит экспериментировать с различными вариантами работы приложения.

Не следует сохранять восстановленную информацию на этот же диск; необходимо найти другой носитель.

Примеры программ:

1. R-studio - В случае удаления важного контента с внутренней памяти компьютера

2. Recuva – многофункциональная утилита.

3. Recovery-software составляет целую группу программ, специализирующихся на восстановлении флешек, так и жестких дисков.

4. Data Rescue специализируется на поврежденных или переформатированных носителях.

5. UndeletPlus отличается большим спектром выбора носителя: жесткие диски, карты памяти фотоаппаратов, мультимедийные плееры, почтовые аккаунты, флешкарты.

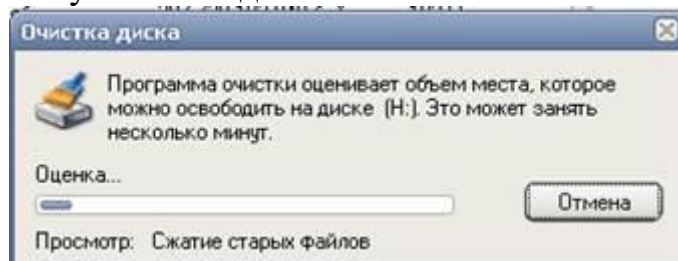
Содержание работы:

Задание 1. Использование программы очистки диска. Произвести очистку диска.

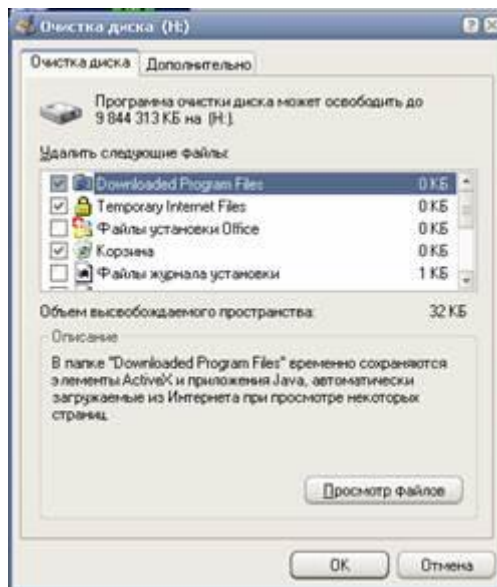
Программа очистки диска помогает очистить пространство на жестком диске. Программа очистки диска проверяет диск и выводит перечень временных файлов, файлов кэша Интернета, а также ненужных программных файлов, удаление которых не приведет к негативным последствиям. Можно выбрать удаление некоторых или всех этих файлов.

Для выполнения некоторых задач может потребоваться войти в систему с учетной записью «Администратор» или члена группы «Администраторы».

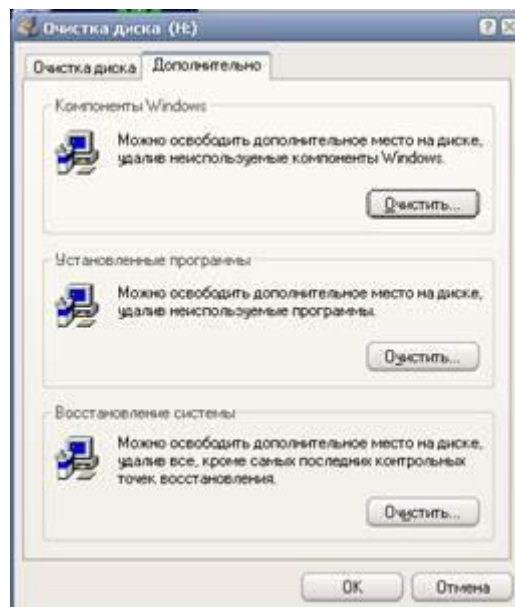
1. Запустите программу Очистка диска.



2. После оценки объема места, которое можно освободить на диске, отметить файлы к удалению и нажать ОК.



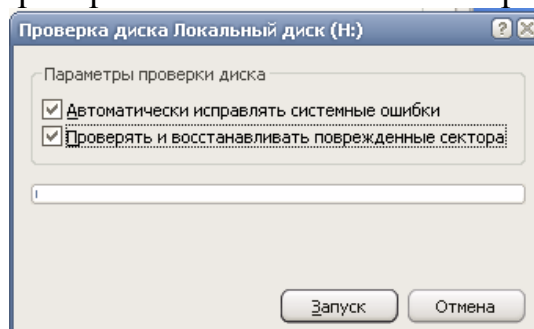
3. При необходимости провести дополнительную очистку диска, удалив неиспользуемые компоненты Windows, неиспользуемое программное обеспечение.



4. Провести анализ освободившегося пространства.

Задание 2. С помощью меню сервис выполнить проверку диска

1. Запустить программу проверки диска. Установить флажки автоматического исправления ошибок, проверки и восстановления поврежденных секторов.

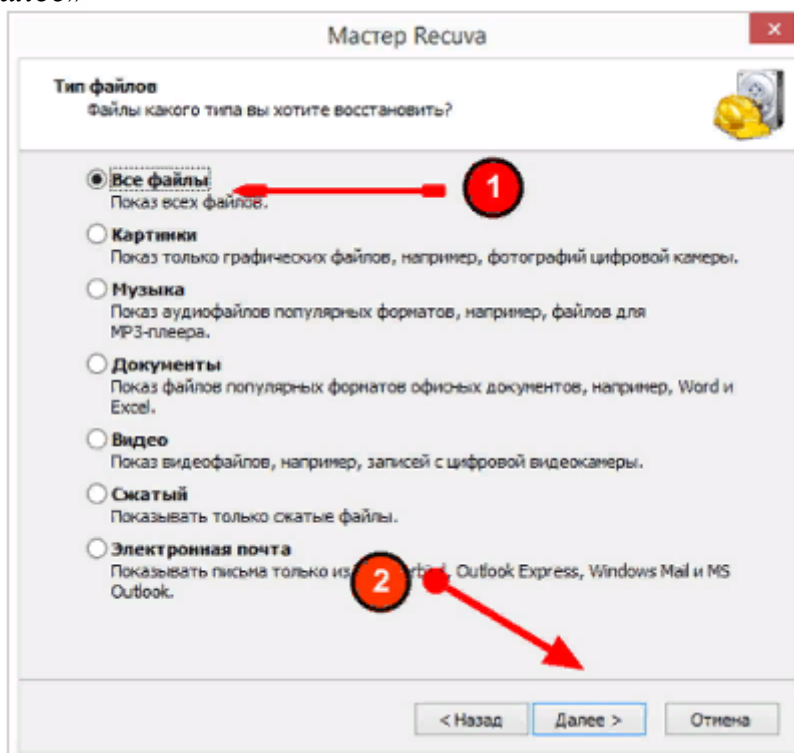


2. Описать процедуру проверки, сделать анализ

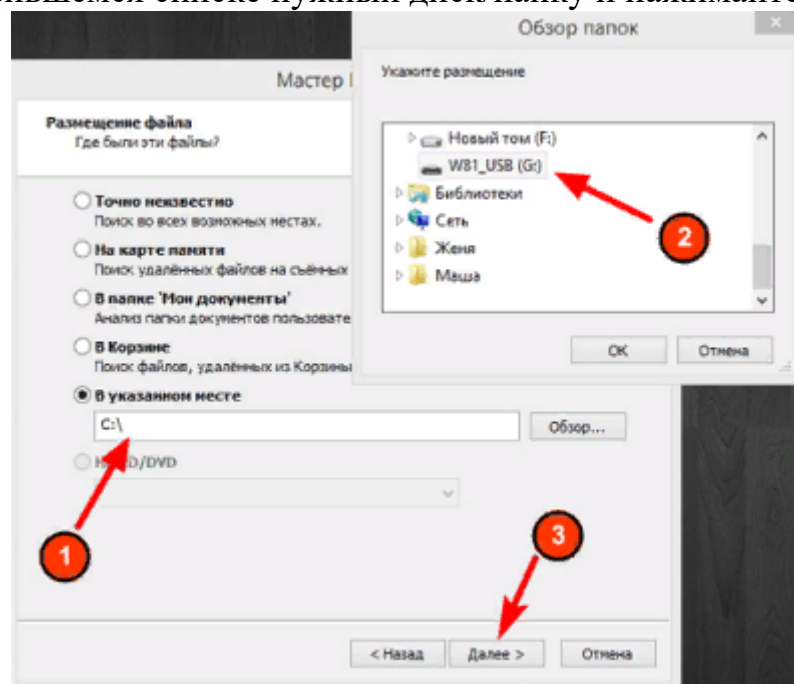
Задание 3. С помощью программы Recuva восстановить данные

1. Запустить программу Recuva.

2. После установки и запуска программы выберете тип восстанавливаемых данных. Если вы, к примеру, хотите восстановить только музыку, удаленную с флешки, выбирайте третий пункт. Если же желаете увидеть все найденные к восстановлению данные, оставляйте флажок на первом пункте «все файлы», и нажимайте «далее»

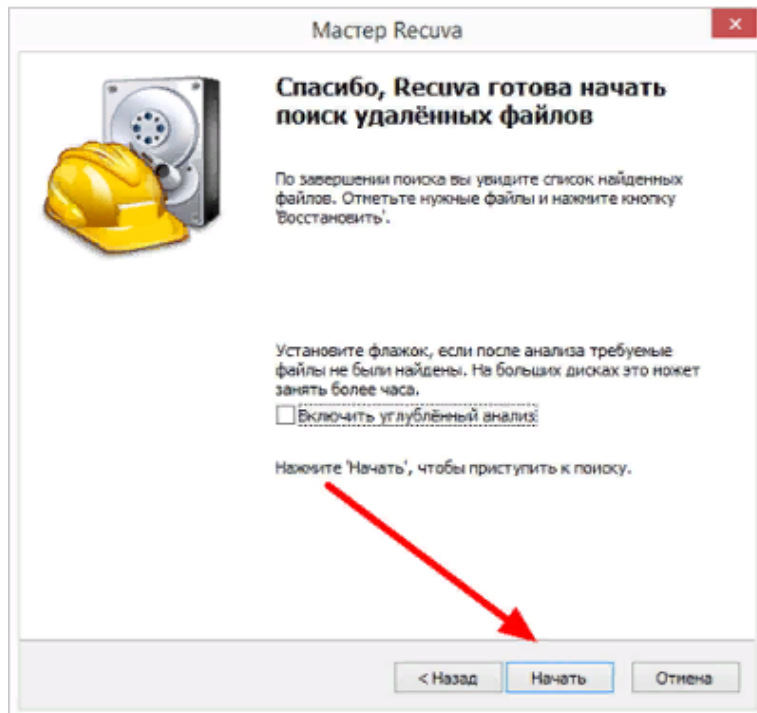


3. Если вы точно знаете месторасположение удаленных файлов (например, флешка), выбирайте пятый пункт «В указанном месте», отмечайте двойным кликом в появившемся списке нужный диск/папку и нажимайте «Далее».

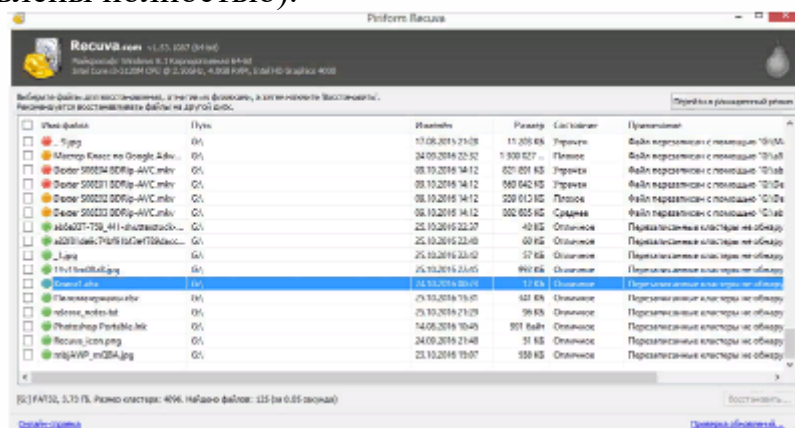


4. После того, как вы выбрали нужный диск/папку, перед вами появится окно завершения работы мастера Recuva, где он предложит вам включить углубленный анализ.

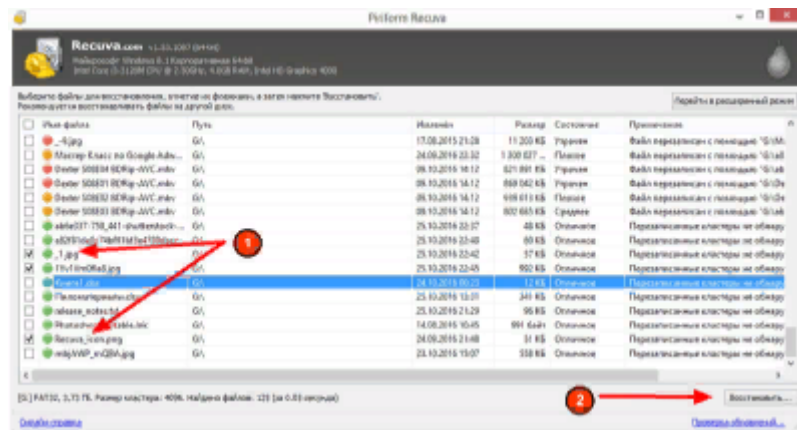
При первичном проходе делать этого не рекомендуется, т.к. включение данной функции сильно увеличит время сканирования, при этом результаты не факт что будут отличаться. Так что оставляйте пока как есть и нажимайте «Начать».



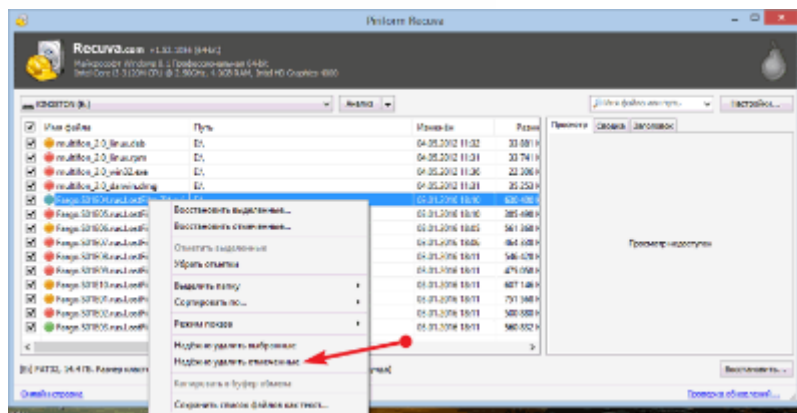
5. После того, как закончится сканирование, перед вами появится окно с найденными удаленными файлами. Как видите, есть три типа цветовой индикации найденных файлов — красные (восстановить не получится), желтые (возможно частичное восстановление) и зеленые (могут быть восстановлены полностью).



6. Отмечайте искомые файлы, нажимайте «Восстановить...» и выбирайте папку, куда будут помещены восстановленные файлы.



7. В том же окне выбора файлов отметьте те, которые хотите удалить навсегда, вызовите правой кнопкой мыши контекстное меню и выбирайте пункт «Надежно удалить отмеченные». Через некоторое время Recuva удалит эти файлы без возможности восстановления.



Информационное обеспечение обучения по дисциплине

Основные учебные издания:

1. Бабаш, А. В., Криптографические методы и средства защиты информации: учебник / А. В. Бабаш, Е. К. Баранова. — Москва: КноРус, 2026. — 222 с. — ISBN 978-5-406-15474-8. — URL: <https://book.ru/book/959682>
2. Шитов, В. Н., Устройство и функционирование информационной системы: учебник / В. Н. Шитов. — Москва: КноРус, 2024. — 333 с. — ISBN 978-5-406-12882-4. — URL: <https://book.ru/book/953436>

Дополнительные учебные издания:

3. Проскуряков, А. В. Качество и тестирование программного обеспечения. Метрология программного обеспечения: учебное пособие / А. В. Проскуряков. — Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2022. — 197 с. — ISBN 978-5-9275-4044-0. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROОбразование: [сайт]. — URL: <https://profspo.ru/books/125702>

Интернет-ресурсы:

Электронно-библиотечная система:

4. ЭБС «Znanium»
5. ЭБС «PROОбразование»
6. ЭБС «Book.ru»