

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
«05» июня 2024 г.



МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ


по дисциплине

МДК.02.01 «Технология разработки программного обеспечения»

направление подготовки

09.02.07 «Информационные системы и программирование»

Методические указания рассмотрены
на заседании предметной (цикловой) комиссии
общепрофессиональных дисциплин,
профессиональных модулей специальностей
технического профиля
«14» июня 2024 года, протокол №12

Председатель ПЦК  /Ю.А.Табарова/

Петровск 20234

Пояснительная записка

Методические указания по выполнению практических работ подготовлены на основе рабочей программы учебной дисциплины МДК.02.01 «Технология разработки программного обеспечения», 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 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях.

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде.

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

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности

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

ОК 10. Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

При выполнении практических работ студент должен *знать*:

- модели процесса разработки программного обеспечения;
- основные принципы процесса разработки программного обеспечения;
- основные подходы к интегрированию программных модулей;
- основы верификации и аттестации программного обеспечения

При выполнении практических работ студент должен *уметь*:

- использовать выбранную систему контроля версий;
- использовать методы для получения кода с заданной функциональностью и степенью качества

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

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

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

Комплект методических указаний по выполнению практических работ по дисциплине МДК.02.01 «Технология разработки программного обеспечения» содержит 9 практических занятий.

**Перечень практических работ
по дисциплине МДК.02.01 «Технология разработки программного
обеспечения»**

ПРАКТИЧЕСКАЯ РАБОТА № 1

Тема: Анализ предметной области

ПРАКТИЧЕСКАЯ РАБОТА № 2

Тема: Разработка и оформление технического задания

ПРАКТИЧЕСКАЯ РАБОТА № 3

Тема: Построение диаграммы Вариантов использования и диаграммы Последовательности

ПРАКТИЧЕСКАЯ РАБОТА № 4

Тема: Построение диаграммы Кооперации и диаграммы Развертывания

ПРАКТИЧЕСКАЯ РАБОТА № 5

Тема: Построение диаграммы Деятельности, диаграммы Состояний и диаграммы Классов

ПРАКТИЧЕСКАЯ РАБОТА № 6

Тема: Построение диаграммы компонентов

ПРАКТИЧЕСКАЯ РАБОТА № 7

Тема: Построение диаграмм потоков данных

ПРАКТИЧЕСКАЯ РАБОТА № 8

Тема: Разработка тестового сценария. Оценка необходимого количества тестов

ПРАКТИЧЕСКАЯ РАБОТА № 9

Тема: Разработка тестовых пакетов

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.

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.

1. Оформление текстовых документов

Текст работы печатается на одной стороне листа формата А4, должен быть только чёрного цвета и иметь поля (верхнее, нижнее – 2 см, левое – 3 см, правое – 1,5 см). Шрифт Times New Roman размером 14, межстрочный интервал 1,5, абзацный отступ 1,25.

Правила оформления таблиц, рисунков, графиков

Все таблицы и рисунки должны иметь названия и порядковую нумерацию (например, Таблица 1, Рисунок 3). Нумерация таблиц и рисунков должна быть сквозной для всего текста до приложений. Таблицы, рисунки каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения (напр., Таблица В.1).

Оформление таблицы.

Название таблицы помещается слева над таблицей без абзацного отступа, в одной строке с ее номером через тире (14 шрифтом).

В каждой таблице следует указывать единицы измерения показателей. Если единица измерения в таблице является общей для всех числовых табличных данных, то ее приводят в заголовке таблицы после ее названия.

При переносе: слово “Таблица” указывают один раз слева над первой частью таблицы, над другими частями пишут слова “Продолжение таблицы” или “Окончание таблицы” справа, с указанием номера (обозначения) таблицы. Если в конце страницы таблица прерывается и ее продолжение будет на следующей странице, то в первой части таблицы нижнюю горизонтальную черту, ограничивающую таблицу, не проводят.

Таблица 1 – Распределение ответов респондентов на вопросы анкеты по возрастным группам (в процентах)

Варианты ответов	Возрастные группы				Всего по выборке
	18-24 года	25-29 лет	30-45 лет	старше 45 лет	

Не допускается прямое копирование в текст Диплома выходных таблиц отчета компьютерной программы STATISTICA. Таблицы должны быть построены заново.

Оформление рисунка.

Все иллюстративные материалы (рисунки, диаграммы, графики) в Дипломе имеют название «Рисунок». На графический материал должна быть дана ссылка в тексте документа.

Иллюстрации могут быть в компьютерном исполнении, в том числе и цветные.

Порядковый номер рисунка и – через тире – его название проставляются под рисунком по центру строки (см. Рисунок 1).



Рисунок 1 – Пятиконечная звезда

2. Составление программы на языке программирования

Правила оформления кода:

1. Используйте разумные имена для переменных и функций

Программа должна быть хорошо понятна человеку при чтении. Если при чтении программы приходится понимать назначение переменных и функций по тому, как они используются, то читать код становится гораздо

сложнее. Неудачно выбранные имена могут привести к тому, что смысл программы может быть неправильно понят. Выбирайте такие имена, которые бы объясняли смысл переменных и функций, тогда код станет гораздо понятнее, и не потребуется писать множество комментариев.

При написании составных слов, например в именах переменных, пишите их слитно без пробелов, при этом каждое новое слово пишется с большой буквы.

2. Не дублируйте код

Если в программе есть одинаковые выражения или фрагмента кода, вынесите этот код в отдельную функцию. Верным признаком необходимости создания новой функции является желание скопировать фрагмент кода из одного места программы в другое. В таком случае сразу перенесите этот фрагмент в отдельную функцию.

Если фрагменты кода похожи, но не идентичны, подумайте, не получится ли и их вынести в одну функцию, возможно добавив параметры или условия.

3. Не используйте «магические константы»

Использование неименованных «магических» констант в коде нежелательно:

- при чтении кода может быть не понятно, что это за число, и почему оно именно такое;
- чаще всего одно и то же число потребуется написать в нескольких местах кода. Если его придётся изменять, можно пропустить одно из использований, что приведёт к ошибке.

Если в коде нужно использовать константу, дайте ей имя, используя `const`.

4. Расставляйте пробелы вокруг бинарных операторов. Это улучшает читаемость формул.

5. Всегда выделяйте блоки условных операторов и циклов скобками. В любой блок условия или цикла может захотеться добавить новое выражение. При этом можно забыть добавить скобки. Лучше сразу добавить скобки, чтобы потом не было с этим проблем.

6. Расставляйте скобки одинаково. Выберите и используйте для себя один из стилей расстановки скобок. Это улучшает читаемость структуры программы.

7. Не делайте строки слишком длинными. Строка программы должна помещаться на экране. Обычно рекомендуют ограничить максимальную ширину строки в 80 символов. Если определение или вызов функции получается слишком широким поместите по одному параметру на каждой строке. Длинные математические выражения разбивайте на несколько строк, разбивая по границам логических блоков выражения.

8. Объявляйте переменные непосредственно перед использованием. Обязательно указывайте начальное значение для переменных. Значение неинициализированной переменной может быть любым. Использование (чтение) такого значения приведёт к недетерминированной работе программы, а в некоторых случаях является неопределённым поведением.

Чтобы избежать проблем всегда инициализируйте переменные прямо в момент их создания.

9.Единый стиль оформления кода во всем проекте;

10. Визуальное выделение наиболее значимых частей — используя *вертикальное форматирование*, мы выделяем объявление переменных, цикл заполнения массива случайными числами и цикл обработки по формуле. Если ваша функция выполняет несколько действий — то разумно разделить соответствующие блоки кода пустыми строками.

ПРАКТИЧЕСКАЯ РАБОТА № 1.

Тема: Анализ предметной области

Цель работы: описать и проанализировать ИС, определить необходимые элементы комплекса технических средств ИС и системного программного обеспечения ИС.

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

Справочный материал:

Руководители программных проектов выполняют такую же работу, что и руководители технических проектов. Вместе с тем процесс разработки ПО существенно отличается от процессов реализации технических проектов. Ниже приведен небольшой список этих отличий.

1. Программный продукт нематериален. Менеджер судостроительного проекта или проекта постройки здания видит результат выполнения своего проекта. Если реализация проекта отстает от графика, то это видно по незавершенности конструкции. В противоположность этому процент незавершенности программного проекта нельзя увидеть или потрогать. Менеджер программного проекта может полагаться только на документацию, которая фиксирует процесс разработки программного продукта.

2. Не существует стандартных процессов разработки программного обеспечения. Не существует четкой зависимости между процессом создания ПО и типом создаваемого программного продукта. Нельзя точно предсказать, на каком этапе процесса разработки ПО могут возникнуть проблемы, угрожающие всему проекту.

3. Большие программные проекты – это часто одноразовые проекты. Большие программные проекты значительно отличаются от проектов, реализованных ранее. Поэтому, чтобы уменьшить неопределенность в планировании проекта, руководители проектов должны обладать большим практическим опытом. Но постоянные технологические изменения в компьютерной технике оценивают предыдущий опыт. Перечисленные особенности могут привести к тому, что реализация проекта выйдет за рамки временного графика или бюджета.

Процессы управления программными проектами:

Невозможно описать и стандартизировать все работы, выполняемые менеджером проекта по созданию ПО, но в большинстве случаев к ним относятся: написание предложений по созданию ПО; планирование и составление графика работ проекта; оценивание стоимости проекта; контроль процессов выполнения работ; подбор персонала; написание отчетов и представлений.

Время выполнения больших программных проектов может занимать несколько лет. В течение этого времени цели и намерения организации, заказавшей программный проект, могут существенно измениться. Может оказаться, что разрабатываемый программный продукт стал уже ненужным или исходные требования к ПО устарели и их нужно кардинально менять. В

такой ситуации руководство организации-разработчика может принять решение о прекращении разработки ПО или об изменении проекта в целом.

Планирование проекта.

Эффективное управление проектами напрямую зависит от правильного планирования работ. План, разработанный на начальном этапе проекта, рассматривается всеми его участниками как руководящий документ, выполнение которого должно привести к успешному завершению проекта. Этот первоначальный план должен максимально подробно описывать все этапы реализации проекта.

План проекта должен показать ресурсы, необходимые для реализации проекта, разделение работ на этапы и временной график выполнения этих этапов. Детализация планов проектов очень разнится в зависимости от типа разрабатываемого программного продукта и организации-разработчика. Но в любом случае большинство планов содержит следующие разделы:

1. Введение. Краткое описание целей проекта и проектных ограничений (бюджетных, временных и т.д.).

2. Организация выполнения проекта. Описание способа подбора команды разработчиков и распределение обязанностей между членами команды.

3. Анализ рисков. Описание возможных проектных рисков, вероятность их проявления и стратегий, направленных на их уменьшение.

4. Аппаратные и программные ресурсы для реализации проекта. Перечень аппаратных средств и программного обеспечения, необходимого для разработки программного продукта.

5. Разбиение работа на этапы. Проект разбивается на отдельные процессы, определяются этапы выполнения проекта, приводится описание результатов каждого этапа и контрольные отметки.

6. График работ. В графике работ отображаются зависимости между отдельными этапами разработки ПО, оценки времени их выполнения и распределение членов команды проекта по отдельным этапам.

7. Механизмы контроля и мониторинга за ходом выполнения проекта. Описываются механизмы и сроки предоставления отчетов о ходе работ, а также механизмы мониторинга всего проекта.

При планировании проекта разработки ПО определяются контрольные точки, отмечающие окончание определенного этапа работ. Для каждой точки создается отчет, который предоставляется руководству проекта.

При определении контрольных точек весь процесс создания ПО должен быть разбит на отдельные этапы с указанным результатом каждого этапа.

Информационная система – это совокупность механизмов, обеспечивающих полное осуществление информационного процесса. Вне ИС информация может лишь сохраняться в виде записей на тех или иных физических носителях, но не может быть ни принятой, ни переданной, ни использованной.

Методологическую основу изучения ИС составляет системный подход, в соответствии с которым любая система представляет собой совокупность взаимосвязанных объектов, функционирующих совместно для достижения общей цели.

Информационная система представляет собой совокупность функциональной структуры, информационного, математического, технического, организационного и кадрового обеспечения, которые объединены в единую системы в целях сбора, хранения, обработки и выдачи необходимой информации для выполнения функций управления.

Задачи информационных систем.

Корпоративные системы позволяют решить следующие задачи: гарантировать требуемое качество управления предприятием; повысить оперативность и эффективность взаимодействия между подразделениями; обеспечить управляемость качеством выпускаемой продукции; увеличить экономическую эффективность деятельности предприятия; создать систему статистического учета на предприятии; осуществлять прогноз развития предприятия; создать систему стратегического и оперативного планирования, систему прогнозирования.

Содержание работы:

Задание 1.

1. Выберите предметную область из предложенных в таблице.
2. Выберите название ИС в рамках предметной области
3. Определите цель ИС
4. Проведите анализ осуществимости ИС.
 - 4.1. Что произойдет с организацией, если система не будет введена в эксплуатацию?
 - 4.2. Какие текущие проблемы существуют в организации и как новая система поможет их решить?
 - 4.3. Каким образом (и будет ли) ИС способствовать целям бизнеса?
 - 4.4. Требуется ли разработка ИС технологии, которая до этого раньше не использовалась в организации?
5. Где будет размещена ИС? Кто является пользователем ИС?
6. Комплекс технических средств ИТ
 - 6.1. Какие средства компьютерной техники необходимы для ИС?
 - 6.2. Какие средства коммуникационной техники необходимы для ИС?
 - 6.3. Какие средства организационной техники необходимы для ИС?
 - 6.4. Какие средства оперативной полиграфии необходимы для ИС?
7. Опишите системное ПО ИТ.

Таблица. Варианты предметных областей

Предметная область	Сущность задачи
Страховая медицинская компания	Страховая медицинская компания (СМК) заключает договоры добровольного медицинского страхования с населением и договоры с лечебными учреждениями на лечение застрахованных клиентов. При возникновении

	<p>страхового случая клиент подает заявку на оказание медицинских услуг по условиям договора инспектору, который работает с данным клиентом. Инспектор направляет данного клиента в лечебное учреждение. Отчеты о своей деятельности инспектор предоставляет в бухгалтерию. Бухгалтерия проверяет оплату договоров, перечисляет денежные средства за оказанные услуги лечебным учреждениям, производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики. СМК не только оплачивает лечение застрахованного лица при возникновении с ним страхового случая, но и, при возникновении каких-либо осложнений после лечения, оплачивает лечение этих осложнений</p>
Агентство недвижимости	<p>Агентство недвижимости занимается покупкой, продажей, сдачей в аренду объектов недвижимости по договорам с их собственниками. Агентство управляет объектами недвижимости как физических, так и юридических лиц. Собственник может иметь несколько объектов. В случае покупки или аренды клиент может произвести осмотр объекта. В качестве одной из услуг, предлагаемых агентством, является проведение инспектирования текущего состояния объекта для адекватного определения его рыночной цены. По результатам своей деятельности агентство производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики</p>
Кадровое агентство	<p>Кадровое агентство способствует трудоустройству безработных граждан. Агентство ведет учет и классификацию данных о безработных на основании резюме от них. От предприятий города поступают данные о свободных вакансиях, на основании которых агентство предлагает различные варианты трудоустройства соискателям. В случае положительного исхода поиска вакансия считается заполненной, а безработный становится трудоустроенным. По результатам своей деятельности кадровое агентство производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики</p>
Компания по разработке программных продуктов	<p>Компания заключает договор с клиентом на разработку программного продукта согласно техническому заданию. После утверждения технического задания определяется состав и объем работ, составляется предварительная смета. На каждый проект назначается ответственный за его выполнение – куратор проекта, который распределяет</p>

	<p>нагрузку между программистами и следит за выполнением технического задания. Когда программный продукт готов, то его внедряют, производят обучение клиента и осуществляют дальнейшее сопровождение. По результатам своей деятельности компания производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики</p>
Туроператор	<p>Туроператор предоставляет возможность своим клиентам осуществить туристическую или деловую поездку в различные города России и мира. При разработке нового тура сначала анализируется текущая ситуация на рынке туризма и выбирается направление тура. После этого определяется статус тура, бронируются места в гостиницах и билеты на переезд к месту тура, разрабатывается культурная/ деловая/ развлекательная программа, утверждаются сроки тура. На каждый тур назначается ответственное лицо от туроператора, которое будет вести данный тур для улаживания проблем в случае возникновения каких-нибудь чрезвычайных или форс-мажорных ситуаций. Клиент приходит в офис туроператора, где вместе с менеджером выбирает уже разработанный тур и оформляет путевку. После возвращения из тура клиент может высказать свои замечания или пожелания, которые будут учтены при доработке существующих туров или при разработке новых. Также, для дальнейшего улучшения тура, туроператор проводит анализ отчетов от посредников (гостиница, гиды и т.д.). По результатам своей деятельности туроператор производит отчисления в налоговые органы и предоставляет отчетность в органы государственной статистики</p>

ПРАКТИЧЕСКАЯ РАБОТА № 2

Тема: Разработка и оформление технического задания

Цель работы: приобретение навыков разработки технического задания на программный продукт, ознакомиться с правилами написания технического задания.

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

Справочный материал:

Техническое задание (ТЗ) – исходный документ для разработки информационных систем.

ТЗ содержит основные технические требования, предъявляемые к программному продукту, и исходные данные для разработки. В ТЗ указываются назначение программного продукта, область его применения, стадии разработки конструкторской, проектной, программной документации, ее состав, сроки исполнения, а также особые требования, обусловленные спецификой продукта либо условиями его эксплуатации. Как правило, ТЗ составляется на основе анализа результатов предварительных требований, исследований. Типовые требования к составу и содержанию ТЗ приведены в таблице.

Таблица 1. Состав и содержание ТЗ (ГОСТ 34.602-89)

№ п/п	Раздел	Содержание
1	Общие сведения	<ul style="list-style-type: none">- полное наименование системы и ее условное обозначение- шифр темы или шифр (номер) договора;- наименование предприятия разработчика и заказчика системы, их реквизиты- перечень документов, на основании которых создается ИС- плановые сроки начала и окончания работ- сведения об источниках и порядке финансирования работ- порядок оформления и предъявления заказчику результатов работ по созданию системы, ее частей и отдельных средств
2	Назначение и цели создания (развития) системы	<ul style="list-style-type: none">- вид автоматизируемой деятельности- перечень объектов, на которых предполагается использование системы- наименования и требуемые значения технических, технологических, производственно-экономических и др. показателей объекта, которые должны быть достигнуты при внедрении ИС
3	Характеристика объектов	<ul style="list-style-type: none">- краткие сведения об объекте автоматизации- сведения об условиях эксплуатации и

	автоматизации	характеристиках окружающей среды
4	Требования к системе	<p>Требования к системе в целом:</p> <ul style="list-style-type: none"> - требования к структуре и функционированию системы (перечень подсистем, уровни иерархии, степень централизации, способы информационного обмена, режимы функционирования, взаимодействие со смежными системами, перспективы развития системы) - требования к персоналу (численность пользователей, квалификация, режим работы, порядок подготовки) - показатели назначения (степень приспособляемости системы к изменениям процессов управления и значений параметров) - требования к надежности, безопасности, эргономике, транспортабельности, эксплуатации, техническому обслуживанию и ремонту, защите и сохранности информации, защите от внешних воздействий, к патентной чистоте, по стандартизации и унификации <p>Требования к функциям (по подсистемам):</p> <ul style="list-style-type: none"> - перечень подлежащих автоматизации задач - временной регламент реализации каждой функции - требования к качеству реализации каждой функции, к форме представления выходной информации, характеристики точности, достоверности выдачи результатов - перечень и критерии отказов <p>Требования к видам обеспечения:</p> <ul style="list-style-type: none"> - математическому (состав и область применения математических моделей и методов, типовых и разрабатываемых алгоритмов) - информационному (состав, структура и организация данных, обмен данными между компонентами системы, информационная совместимость со смежными системами, используемые классификаторы, СУБД, контроль данных и ведение информационных массивов, процедуры придания юридической силы выходным документам) - лингвистическому (языки программирования, языки взаимодействия пользователей с системой, системы кодирования, языки ввода- вывода)

		<ul style="list-style-type: none"> - программному (независимость программных средств от платформы, качество программных средств и способы его контроля, использование фондов алгоритмов и программ) - техническому - метрологическому - организационному (структура и функции эксплуатирующих подразделений, защита от ошибочных действий персонала) - методическому (состав нормативно-технической документации)
5	Состав и содержание работ по созданию системы	<ul style="list-style-type: none"> - перечень стадий и этапов работ - сроки исполнения - состав организаций — исполнителей работ - вид и порядок экспертизы технической документации - программа обеспечения надежности - программа метрологического обеспечения
6	Порядок контроля и приемки системы	<ul style="list-style-type: none"> - виды, состав, объем и методы испытаний системы - общие требования к приемке работ по стадиям - статус приемной комиссии
7	Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие	<ul style="list-style-type: none"> - преобразование входной информации к машиночитаемому виду - изменения в объекте автоматизации - сроки и порядок комплектования и обучения персонала
8	Требования к документированию	<ul style="list-style-type: none"> - перечень подлежащих разработке документов - перечень документов на машинных носителях
9	Источники разработки	<ul style="list-style-type: none"> - документы и информационные материалы, на основании которых разрабатывается ТЗ и система

Содержание работы:

Разработка ТЗ выполняется в следующей последовательности. Прежде всего устанавливают набор выполняемых функций, а также перечень и характеристики исходных данных. Затем определяют перечень результатов, их характеристики и способы представления.

Далее уточняют среду функционирования ПО: конкретную комплектацию и параметры технических средств, версию используемой ОС и, возможно, версии и параметры другого установленного ПО, с которым предстоит взаимодействовать будущему программному продукту.

В случаях, когда разрабатываемое ПО собирает и хранит некоторую информацию или включается в управление каким-либо техническим процессом, необходимо также четко регламентировать действия программы в случае сбоев оборудования и энергоснабжения.

1. Общие положения

1.1 Техническое задание оформляют в соответствии с ГОСТ 19.106-78 на листах формата А4 и А3 по ГОСТ 2.301-68, как правило, без заполнения полей листа. Номера листов (страниц) проставляют в верхней части листа над текстом.

1.2 Лист утверждения и титульный лист оформляют в соответствии с ГОСТ 19.104-78. Информационную часть (аннотацию и содержание), лист регистрации изменений допускается и в документ не включать.

1.3 Для внесения изменений и дополнений в техническое задание на последующих стадиях разработки программы или программного изделия выпускают дополнение к нему. Согласование и утверждение дополнения к техническому заданию проводят в том же порядке, который установлен для технического задания.

1.4. Техническое задание должно содержать следующие разделы:

- введение;
- наименование и область применения;
- основание для разработки;
- назначение разработки;
- технические требования к программе или программному изделию;
- технико-экономические показатели;
- стадии и этапы разработки;
- порядок контроля и приемки;
- приложения.

В зависимости от особенностей программы или программного изделия допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них. При необходимости допускается в техническое задание включать приложения.

2. Содержание разделов

2.1 Введение должно включать краткую характеристику области применения программы или программного продукта, а также объекта (например, системы), в котором предполагается их использовать. Основное назначение введения - продемонстрировать актуальность данной разработки и показать, какое место эта разработка занимает в ряду подобных.

2.2 В разделе «Наименование и область применения» указывают наименование, краткую характеристику области применения программы или программного изделия и объекта, в котором используют программу или программное изделие.

2.3 В разделе «Основание для разработки» должны быть указаны:

- документ (документы), на основании которых ведется разработка. Таким документом может служить план, приказ, договор и т. п.;
- организация, утвердившая этот документ, и дата его утверждения;

- наименование и (или) условное обозначение темы разработки.

2.4 В разделе «Назначение разработки» должно быть указано функциональное и эксплуатационное назначение программы или программного изделия.

2.5 Раздел «Технические требования к программе или программному изделию» должен содержать следующие подразделы:

- требования к функциональным характеристикам;
- требования к надежности;
- условия эксплуатации;
- требования к составу и параметрам технических средств;
- требования к информационной и программной совместимости;
- требования к маркировке и упаковке;
- требования к транспортированию и хранению;
- специальные требования.

В подразделе «Требования к функциональным характеристикам» должны быть указаны требования к составу выполняемых функций, организации входных и выходных данных, временным характеристикам и т. п.

2.5.2 В подразделе «Требования к надежности» должны быть указаны требования к обеспечению надежного функционирования (обеспечение устойчивого функционирования, контроль входной и выходной информации, время восстановления после отказа и т. п.).

2.5.3 В подразделе «Условия эксплуатации» должны быть указаны условия эксплуатации (температура окружающего воздуха, относительная влажность и т. п. для выбранных типов носителей данных), при которых должны обеспечиваться заданные характеристики, а также вид обслуживания, необходимое количество и квалификация персонала.

2.5.4 В подразделе «Требования к составу и параметрам технических средств» указывают необходимый состав технических средств с указанием их технических характеристик.

2.5.5 В подразделе «Требования к информационной и программной совместимости» должны быть указаны требования к информационным структурам на входе и выходе и методам решения, исходным кодам, языкам программирования. При необходимости должна обеспечиваться защита информации и программ.

2.5.6 В подразделе «Требования к маркировке и упаковке» в общем случае указывают требования к маркировке программного изделия, варианты и способы упаковки.

2.5.7 В подразделе «Требования к транспортированию и хранению» должны быть указаны для программного изделия условия транспортирования, места хранения, условия хранения, условия складирования, сроки хранения в различных условиях.

2.5.8 В разделе «Технико-экономические показатели» должны быть указаны:

ориентировочная экономическая эффективность, предполагаемая годовая потребность, экономические преимущества разработки по сравнению с лучшими отечественными и зарубежными образцами или аналогами.

2.6 В разделе «Стадии и этапы разработки» устанавливают необходимые стадии разработки, этапы и содержание работ (перечень программных документов, которые должны быть разработаны, согласованы и утверждены), а также, как правило, сроки разработки и определяют исполнителей.

2.7 В разделе «Порядок контроля и приемки» должны быть указаны виды испытаний и общие требования к приемке работы.

2.8 В приложениях к техническому заданию при необходимости приводят:

- перечень научно-исследовательских и других работ, обосновывающих разработку;
- схемы алгоритмов, таблицы, описания, обоснования, расчеты и другие документы, которые могут быть использованы при разработке;
- другие источники разработки.

В случаях, если какие-либо требования, предусмотренные техническим заданием, заказчик не предъявляет, следует в соответствующем месте указать «Требования не предъявляются».

Задания 1.

1. Разработать техническое задание по варианту выбранному в практической работе №1
2. Оформить отчет

Порядок выполнения отчета по практической работе

1. Разработать техническое задание на программный продукт
2. Оформить работу в соответствии с ГОСТ 19.106-78. При оформлении использовать MS Office.
3. Сдать и защитить работу

Защита отчета по практической работе заключается в предъявлении преподавателю полученных результатов (на экране монитора и печатном виде), демонстрации полученных навыков и ответах на вопросы преподавателя.

ПРАКТИЧЕСКАЯ РАБОТА № 3

Тема: Построение диаграммы вариантов использования и диаграммы последовательности

Цель работы: исследование процесса построения диаграмм вариантов использования и диаграмм последовательности в заданной предметной области.

Оборудование: ПК, программное обеспечение – MS Word, Model Maker, инструкции по выполнению работы.

Справочный материал:

Постановка задачи (описание предметной области).

Магазин осуществляет продажу товаров клиенту путем оформления документов «Заказ». Директор магазина – Дмитрий, принял решение автоматизировать документооборот продаж товара и пригласил для выполнения работ программиста Петра. Поговорив с Дмитрием, в соответствии с концепцией жизненного цикла программы Петр приступил к описанию бизнес-процессов, сопровождающих продажу товара. Взяв за основу язык UML, он начал с построения контекстной диаграммы вариантов использования Use Case diagram. Диаграмма должна ответить на вопрос – «Что должно делаться в системе и кто участник этих процессов?»

Содержание работы:

Задание 1. Создание диаграммы вариантов использования и действующих лиц.

Порядок выполнения задания:

1. Дважды щелкнув мышью на Главной диаграмме Вариантов Использования (*Main*) в браузере, откройте ее.
2. С помощью кнопки *Use Case* (Вариант использования) панели инструментов поместите на диаграмму новый вариант использования. Назовите его "Ввести новый заказ".
3. Повторив этапы 2 и 3, поместите на диаграмму остальные варианты использования:

Изменить существующий заказ

Напечатать инвентарную опись

Обновить инвентарную опись

Оформить заказ

Отклонить заказ

Выполнить поставку заказа

4. С помощью кнопки *Actor* (Действующее лицо) панели инструментов поместите на диаграмму новое действующее лицо.

5. Назовите его "Продавец".

6. Повторив шаги 4 и 5, поместите на диаграмму остальных действующих лиц:

Управляющий магазином

Клерк магазина

Бухгалтерская система

7. Создание абстрактного варианта использования (не требующего дальнейшей декомпозиции). Щелкните правой кнопкой мыши на варианте использования "Отклонить заказ" на диаграмме. В открывшемся меню выберите пункт *Open Specification* (Открыть спецификацию). Установите флажок *Abstract* (Абстрактный), чтобы сделать этот вариант использования абстрактным.

Добавление ассоциаций

1. С помощью кнопки *Unidirectional Association* (Однонаправленная ассоциация) панели инструменте нарисуйте ассоциацию между действующим лицом *Продавец* и вариантом использования "Ввести заказ".
2. Повторив шаг 1, поместите на диаграмму остальные ассоциации, согласно рис. 1.

Добавление связи расширения

С помощью кнопки *Generalization* (Обобщение) панели инструментов нарисуйте связь между вариантом использования "Отклонить заказ" и вариантом использования "Оформить заказ". Стрелка должна быть направлена от первого варианта использования ко второму. Связь расширения означает, что вариант использования "Отклонить заказ" при необходимости дополняет функциональные возможности варианта использования "Оформить заказ".

Щелкните правой кнопкой мыши на новой связи между вариантами использования "Отклонить заказ" и "Оформить заказ".

В открывшемся меню выберите пункт *Open Specification* (Открыть спецификацию). В раскрывающемся списке стереотипов введите слово *extends* (расширение), затем нажмите ОК. Надпись «*extends*» появится на линии данной связи.

Добавление описаний к вариантам использования

Выделите в браузере вариант использования "Ввести новый заказ". В окне документации введите следующее описание: "Этот вариант использования дает клиенту возможность ввести новый заказ в систему". С помощью окна документации добавьте описания ко всем остальным вариантам использования.

Добавление описаний к действующему лицу.

Выделите в браузере действующее лицо *Продавец*. В окне документации введите следующее описание: "Продавец — это служащий, старающийся продать товар". С помощью окна документации добавьте описания к остальным действующим лицам.

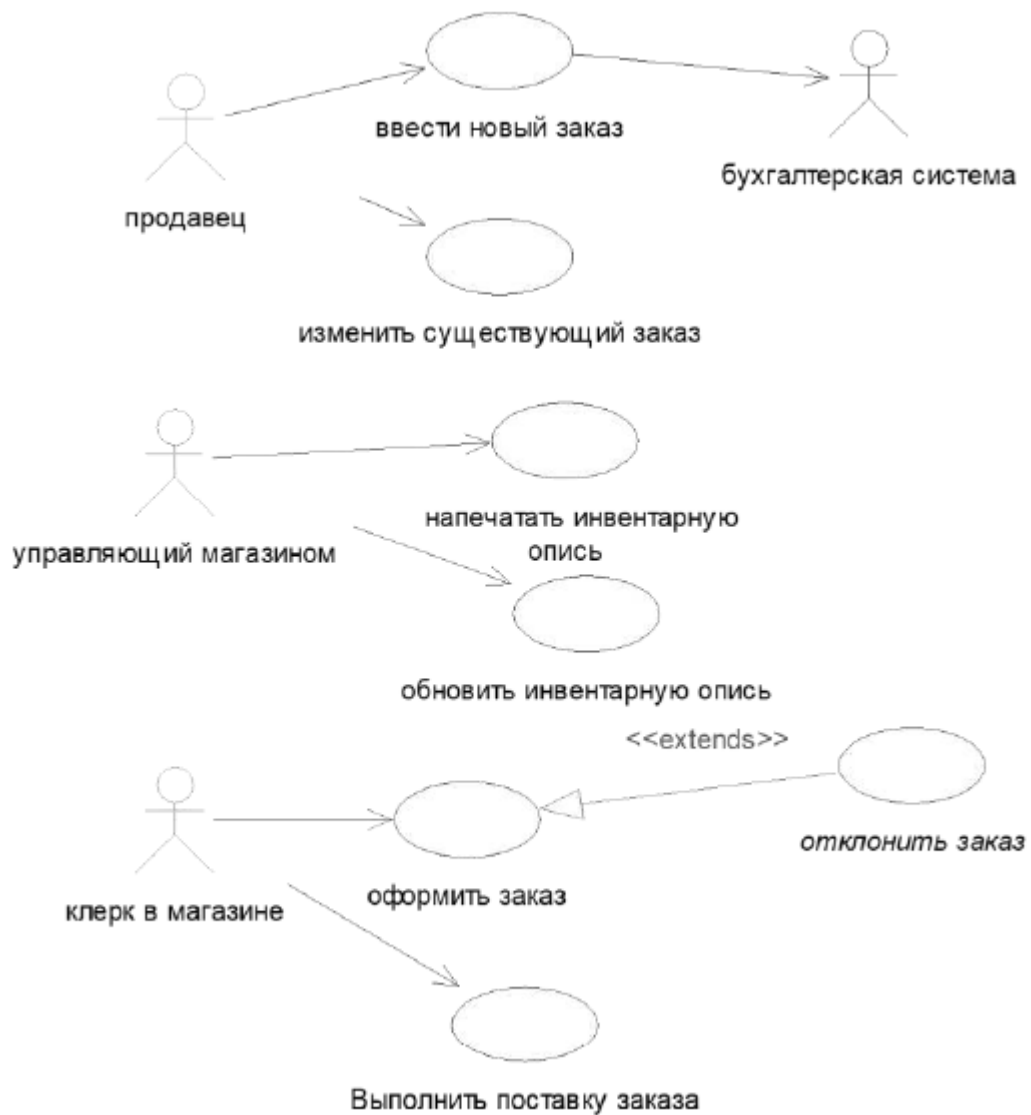


Рис.1. Диаграмма вариантов использования задачи о заказе товара

Задание 2. Создание диаграммы последовательности

Согласовав основные бизнес-процессы с Дмитрием. Петр приступил к построению модели бизнес-процессов, чтобы ответить на вопрос «Как это должно делаться в системе?». Для начала он выбрал наиболее важный вариант использования – «Ввод нового заказа» и построил для него диаграмму последовательности.

Порядок выполнения задания:

Настройка программной среды

1. В меню модели выберите пункт *Tools – Options* (Инструменты – Параметры).
2. Перейдите на вкладку *Diagram* (Диаграмма).
3. Установите флажки *Sequence numbering*, *Focus of control*.
4. Нажмите ОК, чтобы выйти из окна параметров.

Создание диаграммы Последовательности

1. Щелкните правой кнопкой мыши на Логическом представлении браузера.

2. В открывшемся меню выберите пункт *New – Sequence Diagram* (Создать – Диаграмма последовательности).
3. Назовите новую диаграмму: *Ввод заказа*.
4. Дважды щелкнув на этой диаграмме, откройте ее.

Добавление на диаграмму действующего лица и объектов.

1. Перетащите действующее лицо *Продавец* из браузера на диаграмму.
2. Нажмите кнопку *Object* (Объект) панели инструментов.
3. Щелкните мышью в верхней части диаграммы, чтобы поместить туда новый объект.
4. Назовите объект *Выбор варианта заказа*.
5. Повторив шаги 3 и 4, поместите на диаграмму объекты:
 - *Форма деталей заказа*
 - *Заказ №1234*

Добавление сообщений на диаграмму

1. На панели инструментов нажмите кнопку *Object Message* (Сообщение объекта).
 2. Проведите мышью от линии жизни действующего лица *Продавец* к линии жизни объекта *Выбор варианта заказа*.
 3. Выделив сообщение, введите его имя — *Создать новый заказ*.
 4. Повторив шаги 2 и 3, поместите на диаграмму сообщения:
 - *Открыть форму* — между *Выбор Варианта Заказа* и *Форма деталей Заказа*
 - *Ввести номер заказа, заказчика и число заказываемых предметов* — между *Продавец* и *Форма Деталей Заказа*
 - *Сохранить заказ* — между *Продавец* и *Форма Деталей Заказа*
 - *Создать пустой заказ* — между *Форма Деталей Заказа* и *Заказ N1234*
 - *Ввести номер заказа, заказчика и число заказываемых предметов* — между *Форма Деталей Заказа* и *Заказ N1234*
 - *Сохранить заказ* — между *Форма Деталей Заказа* и *Заказ N1234*
- Завершен первый этап работы. Готовая диаграмма последовательности представлена на рис. 2.

Теперь нужно позаботиться об управляющих объектах и о взаимодействии с базой данных. Как видно из диаграммы, объект *Форма Деталей Заказа* имеет множество ответственностей, с которыми лучше всего мог бы справиться управляющий объект. Кроме того, новый заказ должен сохранять себя в базе данных сам. Вероятно, эту обязанность лучше было бы переложить на другой объект.

Добавление на диаграмму дополнительных объектов.

1. Нажмите кнопку *Object* панели инструментов.
2. Щелкните мышью между объектами *Форма Деталей Заказа* и *Заказ №1234*, чтобы поместить туда новый объект.
3. Введите имя объекта — *Управляющий заказами*.
4. Нажмите кнопку *Object* панели инструментов.
5. Новый объект расположите справа от *Заказ №1234*.

6. Введите его имя - *Управляющий транзакциями*.

Назначение ответственностей объектам

1. Выделите сообщение 5: *Создать пустой заказ*.
2. Нажав комбинацию клавиш CTRL+D, удалите это сообщение.
3. Повторите шаги 1 и 2 для удаления двух последних сообщений:
 - *Вести номер заказа, заказчика и число заказываемых предметов*
 - *Сохранить заказ*
4. Нажмите кнопку *Object Message* панели инструментов.
5. Поместите на диаграмму новое сообщение, расположив его под сообщением 4 между *Форма деталей заказа* и *Управляющий заказами*.
6. Назовите его *Сохранить заказ*.
7. Повторите шаги 4 — 6, добавив сообщения с шестого по девятое и назвав их:
 - *Создать новый заказ* — между *Управляющий заказами* и *Заказ №1234*
 - *Ввести номер заказа, заказчика и число заказываемых предметов* - между *Управляющий заказами* и *Заказ №1234*
 - *Сохранить заказ* - между *Управляющий заказами* и *Управляющий транзакциями*
 - *Информация о заказе* — между *Управляющий транзакциями* и *Заказ №1234*
8. На панели инструментов нажмите кнопку *Message to Self* (Сообщение себе).
9. Щелкните на линии жизни объекта *Управляющий транзакциями* ниже сообщения 9, добавив туда рефлексивное сообщение.
10. Назовите его *Сохранить информацию о заказе в базе данных*.

Соотнесение объектов с классами

1. Щелкните правой кнопкой мыши на объекте *Выбор варианта заказа*.
2. В открывшемся меню выберите пункт *Open Specification* (Открыть спецификацию).

В раскрывающемся списке классов выберите пункт *<New>* (Создать). Появится окно спецификации классов.

4. В поле *Name* введите *Выбор заказа*.
5. Щелкните на кнопке *ОК*. Вы вернетесь в окно спецификации объекта.
6. В списке классов выберите класс *Выбор Заказа*.
7. Щелкните на кнопке *ОК*, чтобы вернуться к диаграмме. Теперь объект называется *Выбор варианта заказа: Выбор Заказа*.
8. Для соотнесения остальных объектов с классами повторите шаги с 1 по 7:
 - Класс *Детали заказа* соотнесите с объектом *Форма деталей заказа*
 - Класс *Упр_заказами* — с объектом *Управляющий заказами*
 - Класс *Заказ* — с объектом *Заказ N 1234*
 - Класс *Упр_транзакциями* — с объектом *Управляющий транзакциями*

Соотнесение сообщений с операциями

1. Щелкните правой кнопкой мыши на сообщении 1: *Создать новый заказ*.

2. В открывшемся меню выберите пункт *<new operation>* (создать операцию). Появится окно спецификации операции.
 3. В поле *Name* введите имя операции — *Создать*.
 4. Нажмите на кнопку ОК, чтобы закрыть окно спецификации операции и вернуться к диаграмме.
 5. Еще раз щелкните правой кнопкой мыши на сообщении 1.
 6. В открывшемся меню выберите новую операцию *Создать()*.
 7. Повторите шаги с 1 по 6, чтобы соотнести с операциями все остальные сообщения:
- Сообщение 2: *Открыть форму* соотнесите с операцией *Открыть()*
 - Сообщение 3: *Ввести номер заказа, заказчика и число заказываемых предметов* — с операцией *Ввести номер заказа, заказчика и число заказываемых предметов()*
 - Сообщение 4: *Сохранить заказ* — с операцией *Сохранить заказ()*
 - Сообщение 5: *Сохранить заказ* — с операцией *Сохранить заказ()*
 - Сообщение 6: *Создать пустой заказ* – с операцией *Создать пустой заказ()*
 - Сообщение 7: *Ввести номер заказа, заказчика и число заказываемых предметов* – с одноименной операцией.
 - Сообщение 8 *Сохранить заказ* – с операцией *Сохранить заказ()*
 - Сообщение 9 *Информация о заказе* – с одноименной операцией
 - Сообщение 10 *Сохранить информацию о заказе* с одноименной операцией.

Диаграмма должна выглядеть, как на рис. 3.



Рис. 2. Диаграмма последовательности без управляющих элементов.

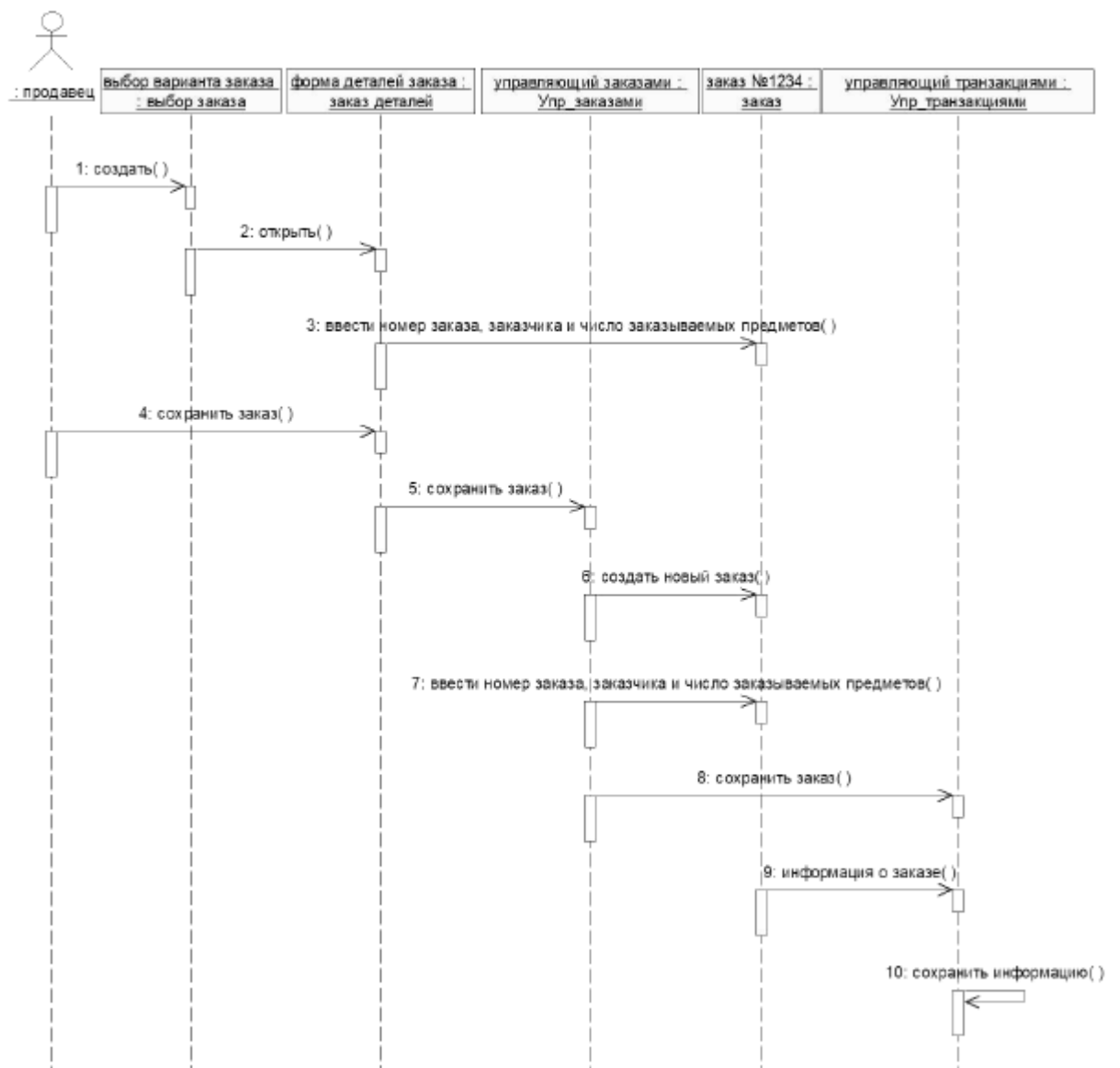


Рис. 3. Окончательный вид диаграммы последовательности

ПРАКТИЧЕСКАЯ РАБОТА № 4

Тема: Построение диаграммы кооперации и диаграммы развертывания

Цель работы: исследование моделирования процессов, описывающих взаимодействие объектов в диаграмме кооперации и диаграмме развертывания в заданной предметной области.

Оборудование: ПК, программное обеспечение – Model Maker, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Построение диаграммы кооперации

1. Щелкните правой кнопкой мыши на Логическом представлении в браузере.
2. В открывшемся меню выберите пункт *New – Collaboration Diagram* (Создать – Кооперативная диаграмма).
3. Назовите эту диаграмму *Ввод заказа*.
4. Дважды щелкнув мышью на диаграмме, откройте ее.

Добавление действующего лица и объектов на диаграмму

1. Перетащите действующее лицо *Продавец* из браузера на диаграмму.
2. Нажмите кнопку *Object* (Объект) панели инструментов.
3. Щелкните мышью где-нибудь внутри диаграммы, чтобы поместить туда новый объект.
4. Назовите объект *Выбор варианта заказа*.
5. Повторив шаги 3 и 4, поместите на диаграмму объекты:
 - *Форма деталей заказа*
 - *Заказ №1234*

Добавление сообщений на диаграмму

1. На панели инструментов нажмите кнопку *Object Link* (Связь объекта).
2. Проведите мышью от действующего лица *Продавец* к объекту у *Выбор варианта заказа*.
3. Повторите шаги 1 и 2, соединив связями следующие объекты:
 - Действующее лицо *Продавец* и объект *Форма деталей Заказа*
 - Объект *Форма деталей Заказа* и объект *Выбор Варианта Заказа*
 - Объект *Форма деталей Заказа* объект *Заказ N1234*
4. На панели инструментов нажмите кнопку *Link Message* (Сообщение связи).
5. Щелкните мышью на связи между *Продавец* и *Форма деталей Заказа*.
6. Выделив сообщение, введите его имя — *Создать новый заказ*;
7. Повторив шаги с 4 по 6, поместите на диаграмму сообщения:
 - *Открыть форму* — между *Выбор Варианта Заказа* и *Форма Деталей Заказа*.
 - *Ввести номер заказа, заказчика и число заказываемых предметов* — между *Продавец* и *Форма Деталей Заказа*
 - *Сохранить заказ* — между *Продавец* и *Форма деталей Заказа*
 - *Создать пустой заказ* — между *Форма деталей Заказа* и *Заказ №1234*
 - *Ввести номер заказа, заказчика и число заказываемых предметов* — между *Форма деталей Заказа* и *Заказ №1234*
 - *Сохранить заказ* — между *Форма деталей Заказа* и *Заказ №1234*

Теперь нужно поместить на диаграмму дополнительные элементы, а также рассмотреть ответственности объектов.

Добавление на диаграмму дополнительных объектов

1. Нажмите кнопку *Object* панели инструментов.
2. Щелкните мышью где-нибудь на диаграмме, чтобы поместить туда новый объект.
3. Введите имя объекта — *Управляющий заказами*.
4. На панели инструментов нажмите кнопку *Object*.
5. Поместите на диаграмму еще один объект.
6. Введите его имя — *Управляющий транзакциями*.

Назначение ответственностей объектам

1. Выделите сообщение 5: *Создать пустой заказ*. Выделяйте слова, а не стрелку.
2. Нажав комбинацию клавиш CTRL+D, удалите это сообщение.
3. Повторите шаги 1 и 2 для удаления сообщений 6 и 7:
 - *Ввести номер заказа, заказчика и число заказываемых предметов*
 - *Сохранить заказ*
4. Выделите связь между объектами *Форма деталей Заказа* и *Заказ №1234*
5. Нажав комбинацию клавиш CTRL+D, удалите эту связь
6. На панели инструментов нажмите кнопку *Object Link* (Связь объекта).
7. Нарисуйте связь между *Форма деталей Заказа* и *Управляющий Заказами*.
8. На панели инструментов нажмите кнопку *Object Link* (Связь объекта).
9. Нарисуйте связь между *Управляющий Заказами* и *Заказ №1234*
10. На панели инструментов нажмите кнопку *Object Link* (Связь объекта).
11. Нарисуйте связь между *Заказ №1234* и *Управляющий Транзакцией*.
12. На панели инструментов нажмите кнопку *Object Link* (Связь объекта).
13. Нарисуйте связь между *Управляющий Заказами* и *Управляющий Транзакцией*.
14. На панели инструментов нажмите кнопку *Link Message* (Сообщение связи).
15. Щелкните мышью на связи между объектами *Форма деталей Заказа* и *Управляющий Заказами*, чтобы ввести новое сообщение.
16. Назовите это сообщение *Сохранить заказ*.
17. Повторите шаги 14 — 16, добавив сообщения с шестого по девятое, и назвав их:
 - *Создать новый заказ* — между *Управляющий Заказами* и *Заказ №1234*
 - *Ввести номер заказа, заказчика и число заказываемых предметов* — между *Управляющий Заказами* и *Заказ №1234*
 - *Сохранить заказ* — между *Управляющий Заказами* и *Управляющий Транзакцией*
 - *Информация о заказе* — между *Управляющий Транзакцией* и *Заказ №1234*
18. На панели инструментов нажмите кнопку *Link to Self* (Связь с собой).
19. Щелкнув на объекте *Управляющий Транзакцией*, добавьте к нему рефлексивное сообщение.

20. На панели инструментов нажмите кнопку *Link Message* (Сообщение связи).

21. Щелкните мышью на рефлексивной связи *Управляющий Транзакциями*, чтобы ввести туда сообщение.

22. Назовите новое *Сохранить информацию о заказе в базе данных*.

Соотнесение объектов с классами (если классы были созданы при разработке описанной выше диаграммы Последовательности)

1. Найдите в браузере класс *Выбор Заказа*.

2. Перетащите его на объект *Выбор варианта заказа* на диаграмме.

3. Повторите шаги 1 и 2 соотнеся остальные объекты и соответствующие им классы:

- Класс *заказ деталей* соотнесите с объектом *Форма деталей заказа*

- Класс *Упр_заказами* — с объектом *Управляющий Заказами*

- Класс *Заказ* — с объектом *Заказ №1234*

- Класс *Упр_транзакциями* — с объектом *Управляющий транзакциями*

Соотнесение объектов с классами (если вы не создавали описанную выше диаграмму Последовательности)

1. Щелкните правой кнопкой мыши на объекте *Форма деталей Заказа*.

2. В открывшемся меню выберите пункт *Open Specification* (Открыть спецификацию).

3. В раскрывающемся списке классов выберите пункт *<New>* (Создать). Появится окно спецификации классов.

4. В поле имени введите *Выбор заказа*.

5. Щелкните на кнопке ОК. Вы вернетесь в окно спецификации объекта.

6. В списке классов выберите класс *Выбор заказа*.

7. Щелкните на кнопке ОК, чтобы вернуться к диаграмме. Теперь объект называется *Выбор варианта заказа: Выбор Заказа*

8. Для соотнесения остальных объектов с классами повторите шаги с 1 по 7:

- Класс *Детали заказа* соотнесите с объектом *Форма деталей заказа*

- Класс *Упр_заказами* — с объектом *Управляющий заказами*

- Класс *Заказ* — с объектом *Заказ N 1234*

- Класс *Упр_транзакциями* — с объектом *Управляющий транзакциями*

Соотнесение сообщений с операциями (если операции были созданы при разработке описанной выше диаграммы Последовательности)

1. Щелкните правой кнопкой мыши на сообщении 1: Создать новый заказ.

2. В открывшемся меню выберите пункт *Open Specification* (Открыть спецификацию).

3. В раскрывающемся списке имен укажите имя операции — *Создать()*.

4. Нажмите на кнопку ОК.

5. Повторите шаги 1—4 для соотнесения с операциями остальных сообщений:

- Сообщение 2: *Открыть форму* соотнесите с операцией *Открыть()*

- Сообщение 3: *Ввести номер заказа, заказчика и число заказываемых предметов* — с операцией *Ввести номер заказа, заказчика и число заказываемых предметов()*
- Сообщение 4: *Сохранить заказ* — с операцией *Сохранить заказ()*
- Сообщение 5: *Сохранить заказ* — с операцией *Сохранить заказ()*
- Сообщение 6: *Создать пустой заказ* – с операцией *Создать пустой заказ()*
- Сообщение 7: *Ввести номер заказа, заказчика и число заказываемых предметов* – с одноименной операцией.
- Сообщение 8 *Сохранить заказ* – с операцией *Сохранить заказ()*
- Сообщение 9 *Информация о заказе* – с одноименной операцией
- Сообщение 10 *Сохранить информацию о заказе* с одноименной операцией

Соотнесение сообщений с операциями (если вы не создавали описанную выше диаграмму Последовательности)

1. Щелкните правой кнопкой мыши на сообщении 1: *Создать новый заказ()*.
 2. В открывшемся меню выберите пункт *<new operation>* (создать операцию). Появится окно спецификации операции.
 3. В поле имени введите имя операции — *Создать()*.
 4. Нажмите на кнопку ОК, чтобы закрыть окно спецификации операции и вернуться к диаграмме.
 5. Еще раз щелкните правой кнопкой мыши на сообщении 1.
 6. В открывшемся меню выберите пункт *Open Specification* (Открыть спецификацию).
 7. В раскрывающемся списке *Name* (Имя) укажите имя новой операции.
 8. Нажмите на кнопку ОК.
 9. Повторите шаги 1—8, чтобы создать новые операции и соотнести с ними остальные сообщения:
- Сообщение 2: *Открыть форму* соотнесите с операцией *Открыть()*
 - Сообщение 3: *Ввести номер заказа, заказчика и число заказываемых предметов* — с операцией *Ввести номер заказа, заказчика и число заказываемых предметов()*
 - Сообщение 4: *Сохранить заказ* — с операцией *Сохранить заказ()*
 - Сообщение 5: *Сохранить заказ* — с операцией *Сохранить заказ()*
 - Сообщение 6: *Создать пустой заказ* – с операцией *Создать пустой заказ()*
 - Сообщение 7: *Ввести номер заказа, заказчика и число заказываемых предметов* – с одноименной операцией.
 - Сообщение 8: *Сохранить заказ* – с операцией *Сохранить заказ()*
 - Сообщение 9: *Информация о заказе* – с одноименной операцией
 - Сообщение 10: *Сохранить информацию о заказе* с одноименной операцией
- Ваша диаграмма должна выглядеть, как показано на рис. 1

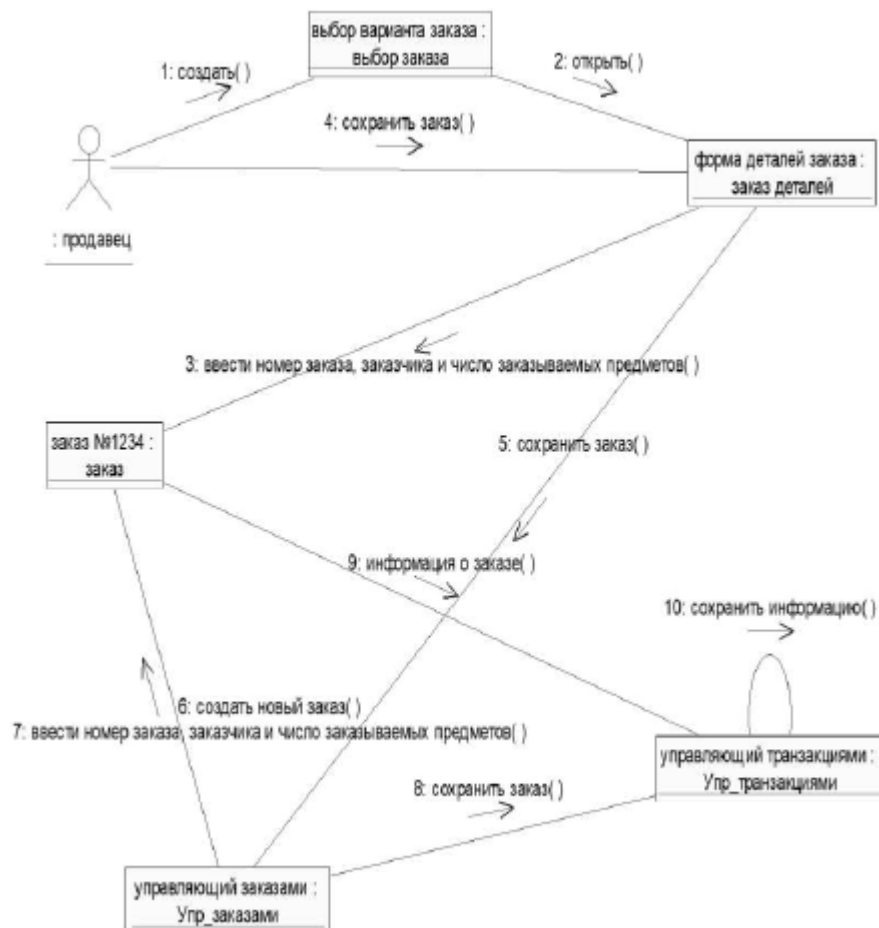


Рис. 1. Окончательный вид кооперативной диаграммы.

ПРАКТИЧЕСКАЯ РАБОТА № 5

Тема: Построение диаграммы деятельности, диаграммы состояний и диаграммы классов

Цель работы: исследование процесса построения диаграммы состояний и диаграммы классов в заданной предметной области.

Оборудование: ПК, программное обеспечение – Model Maker, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Постройте диаграмму состояний

Порядок выполнения задания:

Создание диаграммы

1. Найдите в браузере класс *Заказ*.
2. Щелкните на классе правой кнопкой мыши и в открывшемся меню укажите пункт *New – Statechart Diagram* (Создать диаграмму состояний).

Добавление начального и конечного состояний

1. Нажмите кнопку *Start State* (Начальное состояние) панели инструментов.
2. Поместите это состояние на диаграмму.
3. Нажмите кнопку *End State* (Конечное состояние) панели инструментов.
4. Поместите это состояние на диаграмму.

Добавление суперсостояния

1. Нажмите кнопку *State* (Состояние) панели инструментов.
2. Поместите это состояние на диаграмму.

Добавление оставшихся состояний

1. На панели инструментов нажмите кнопку *State* (Состояние).
2. Поместите состояние на диаграмму.
3. Назовите состояние *Отменен*.
4. На панели инструментов нажмите кнопку *State* (Состояние).
5. Поместите состояние на диаграмму.
6. Назовите состояние *Выполнен*.
7. На панели инструментов нажмите кнопку *State* (Состояние).
8. Поместите состояние на диаграмму внутрь суперсостояния.
9. Назовите состояние *Инициализация*.
10. На панели инструментов нажмите кнопку *State* (Состояние).
11. Поместите состояние на диаграмму внутрь суперсостояния.
12. Назовите состояние *Выполнение заказа приостановлено*.

Описание состояний

1. Дважды щелкните мышью на состоянии *Инициализация*.
2. Перейдите на вкладку *Detail* (Подробно).
3. Щелкните правой кнопкой мыши в окне *Actions* (Действия).
4. В открывшемся меню выберите пункт *Insert* (Вставить).
5. Дважды щелкните мышью на новом действии.
6. Назовите его *Сохранить дату заказа*.
7. Убедитесь, что в окне *When* (Когда) указан пункт *On Entry* (На входе).
8. Повторив шаги 3—7, добавьте следующие действия:

- *Собрать клиентскую информацию*, в окне *When* укажите *DO* (Выполнять между входом и выходом)

- *Добавить к заказу новые позиции*, укажите *DO*

9. Нажмите два раза на ОК, чтобы закрыть спецификацию.

10. Дважды щелкните мышью на состоянии *Отменен*.

11. Повторив шаги 2—7, добавьте действия: *Сохранить дату отмены*, укажите *On Exit* (На выходе)

12. Нажмите два раза на ОК, чтобы закрыть спецификацию.

13. Дважды щелкните мышью на состоянии *Выполнен*.

14. Повторив шаги 2—7, добавьте действие: *Выписать счет*, укажите *On Exit*

15. Нажмите два раза на ОК, чтобы закрыть спецификацию.

Добавление переходов

1. Нажмите кнопку *Transition* (Переход) панели инструментов.

2. Щелкните мышью на начальном состоянии.

3. Проведите линию перехода к состоянию *Инициализация*.

4. Повторив шаги с первого по третий, создайте следующие переходы:

- От состояния *Инициализация* к состоянию *Выполнение заказа приостановлено*

- От состояния *Выполнение заказа приостановлено* к состоянию *Выполнен*

- От суперсостояния к состоянию *Отменен*

- От состояния *Отменен* к конечному состоянию

- От состояния *Выполнен* к конечному состоянию

5. На панели инструментов нажмите кнопку *Transition to Self* (Переход к себе).

6. Щелкните мышью на состоянии *Выполнение заказа приостановлено*

Описание переходов

1. Дважды щелкнув мышью на переходе от состояния *Инициализация* к состоянию *Выполнение заказа приостановлено*, откройте окно спецификации перехода.

2. В поле *Event* (Событие) введите фразу *Выполнить заказ*.

3. Щелкнув на кнопке ОК, закройте окно спецификации.

4. Повторив шаги с первого по третий, добавьте событие *Отменить заказ* к переходу между суперсостоянием и состоянием *Отменен*.

5. Дважды щелкнув мышью на переходе от состояния *Выполнение заказа приостановлено* к состоянию *Выполнен*, откройте окно его спецификации.

6. В поле *Event* (Событие) введите фразу *Добавить к заказу новую позицию*.

7. Перейдите на вкладку *Detail* (Подробно).

8. В поле *Guard Condition* (Сторожевое Условие) введите *Не осталось незаполненных позиций*.

9. Щелкнув на кнопке ОК, закройте окно спецификации.

10. Дважды щелкните мышью на рефлексивном переходе (*Transition to Self*) состояния *Выполнение заказа приостановлено*.

11. В поле *Event* (Событие) введите фразу *Добавить к заказу новую позицию*.

12. Перейдите на вкладку *Detail* (Подробно).

13. В поле *Guard Condition* (Сторожевое Условие) введите *Остаются незаполненные позиции*.

14. Щелкнув на кнопке ОК, закройте окно спецификации.



Рис.1. Диаграмма состояний для класса Заказ

Задание 2. Построение диаграммы активности для варианта использования «Выполнить поставку Заказа».

Побеседовав с Петром, Дмитрий понял, что необходимо согласовать логику реализации еще одного варианта использования «Выполнить поставку заказа». Стало ясно, что здесь возможны несколько альтернативных потоков управления. Для таких ситуаций более удобно использовать не диаграммы последовательности и кооперации, приспособленные для единственного потока, а диаграмму активности.

Описание варианта использования.

При оформлении заказа проверяют каждую содержащуюся в нем позицию, чтобы убедиться в наличии соответствующих товаров на складе. После этого выписываются товары для реализации заказа. Во время выполнения этих процедур одновременно проверяется прохождение платежа. Если платеж прошел и товары имеются на складе, то осуществляется их поставка. Если платеж прошел, но товары на складе отсутствуют, то заказ ставится в ожидание. Если платеж не прошел, то заказ аннулируется.

Порядок выполнения задания:

1. Найдите в браузере вариант использования «Выполнить поставку заказа»
2. Щелкните на классе правой кнопкой мыши и в открывшемся меню укажите пункт *New – Activity Diagram* (Создать диаграмму активности).
3. Назовите диаграмму «Выполнить поставку» и откройте ее двукратным щелчком мыши
4. На панели инструментов щелкните мышкой на элементе *Swimlane*, затем на поле диаграммы. На диаграмме появится разделительная линия («водная дорожка»).
5. Установите курсор на заголовок *NewSwimlane* и нажмите правую клавишу мыши. В выпадающем списке нажмите *Select in browser*. В браузере выделится этот объект. Нажав правую клавишу мыши в выпадающем списке выберете *Open Specification* и откройте спецификацию. Измените поле *Name* на *Клерк*. Выберите в поле *Class* *Клерк в магазине*.
6. Выполните заново пункты 5-6 и присвойте полю *Name* *Система*, *Class-Бухгалтерская система*.
7. Найдите в браузере сплошной черный кружок (начальное состояние). Перенесите его на дорожку *Клерк*.
8. Выберите из панели инструментов объект *Activity* и поместите его на диаграмму в дорожку *Клерк*. Измените имя объекта на *Получить заказ*.
9. Повторите предыдущий этап, создайте на «дорожке» *Клерк* 4 новых *Activity* и присвойте им имена *Проверить позицию заказа, закрепить позицию за заказом, Поставить заказ в ожидание, Скомплектовать заказ*
10. Поместите на «дорожку» 2 новых объекта *End State* (конечное состояние). Одному из них измените поле *Name* на «Выполнить поставку»
11. На дорожку *Система* поместите новый объект *Activity* и присвойте полю *Name* —*Проверить платеж*. На эту же дорожку поместите новый объект *End State* и измените в его спецификации поле *Name* на «Отменить заказ».
12. Поместить на «дорожку» *Клерк* 2 объекта *Horisontal Sinhronization* (горизонтальная синхронизация). Присвойте полю *Name* спецификации одного объекта «1», другого- «2».
13. Поместить на «дорожку» *Клерк* объект *Dicision* (выбор). Через спецификацию присвойте полю *Name* «Позиция имеется?».
14. Поместить на «дорожку» *Система* объект *Dicision*. Присвойте полю *Name* «Деньги поступили?».
15. Щелкните мышкой на панели инструментов объекте - стрелке *State Transition* (состояние перехода). Затем щелкните мышкой на диаграмме объекта начальное состояние. Удерживая кнопку мыши перенесите курсор на активность *Получить заказ* и лишь затем отпустить курсор. В результате два объекта будут соединены стрелкой.
16. Выполните этап 14, соединив стрелкой объект Активность «Получить заказ» с объектом *Horisontal Sinhronization* 1.
17. Соедините этими же стрелками объекты 1 и «Проверить платеж», 1 и «Проверить позицию заказа», «Проверить заказ» и «Деньги поступили?», «Деньги поступили?» и «Отменить заказ», «Проверить позицию заказа» и

«Позиция имеется», «Позиция имеется» и «Закрепить имеется?» и «Поставить заказ в ожидание», 2 и «Скомплектовать заказ», «Скомплектовать заказ» и «Выполнить поставку», «Поставить заказ в ожидание» и объект Конечное состояние (без имени).

18. Присвоим некоторым стрелкам наименование полю *Event* (условие перехода). Для этого, установим курсор на стрелке, соединяющей «Денги получены?» и «Отменить заказ». Двукратным щелчком мыши откроем окно спецификации. В поле *Event* введем «Нет».

19. Выполним пункт 18 для стрелки, соединяющей «Денги получены?» и 2 и присвойте *Event* «Да». Аналогично для стрелки соединяющей «Позиция имеется?» и «Закрепить позицию за заказом» присвоить *Event* «Да». Стрелке, соединяющей «Позиция имеется?» и «Поставить заказ в ожидание» - «Нет».

20. Добавим элементарные действия (Actions) к активности - Проверить позицию заказа». Установим курсор на «Проверить позицию заказа» и двукратным щелчком мыши откроем окно спецификации. Откроем закладку Actions. Установим курсор на свободное поле и нажмем правую клавиши мыши. В выпадающем меню нажмем *Insert*. В появившейся заставке в поле *When* выберем *Entry* (на входе в активность), В поле *Name* введем «Просмотреть спецификацию к заказу». Нажать Ок. Вновь нажмем курсор правой мыши и введем новое действие. Полю *When* присвоим *Do* (промежуток между входом и выходом), а полю *Name* «Найти новую позицию». При вводе третьей активности полю *When* присвойте *Exit* (выход), а полю *Name* «Передать результаты поиска».

21. Путем перемещения объектов (установить курсор мыши - нажать-тащить - отпустить) привести диаграмму к виду, показанному на рис.2

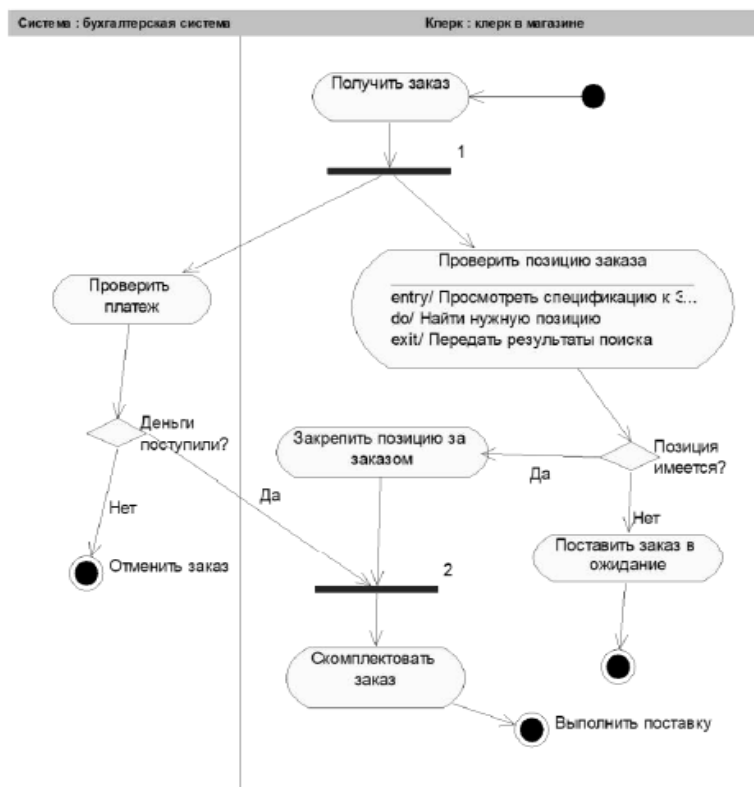


Рис.2. Диаграмма активности для варианта использования

Задание 3. Пакеты и классы.

В этом упражнении необходимо сгруппировать в пакеты классы, созданные при выполнении предыдущих работ. Затем нужно будет построить несколько диаграмм классов и показать на них классы и пакеты системы.

Создание диаграммы классов

Объедините обнаруженные классы в пакеты. Создайте диаграмму классов для отображения пакетов, диаграммы классов, для представления классов в каждом пакете и диаграмму Классов для представления всех классов варианта использования "Ввести новый заказ".

Порядок выполнения задания:

Создание пакетов

1. Щелкните правой кнопкой мыши на Логическом представлении браузера.
2. В открывшемся меню выберите пункт *New - Package* (Создать - Пакет).
3. Назовите новый пакет Сущности.
4. Повторив шаги 1—3, создайте пакеты Границы и Управление.

Создание Главной диаграммы Классов

1. Дважды щелкнув мышью на Главной диаграмме Классов, находящейся под Логическим представлением браузера, откройте ее.
 2. Перетащите пакет *Сущности* из браузера на диаграмму.
 3. Перетащите пакеты *Границы* и *Управление* из браузера на диаграмму.
- Главная диаграмма Классов должна выглядеть, как показано на рис. 3

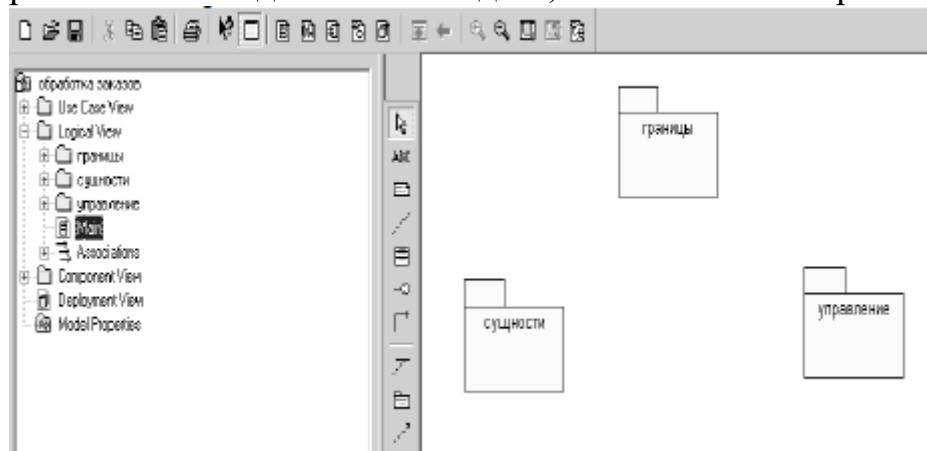


Рис. 3 Главная диаграмма классов в логическом представлении браузера

Создание диаграммы Классов для сценария "Ввести новый заказ" с отображением всех классов

1. Щелкните правой кнопкой мыши на Логическом представлении браузера.
2. В открывшемся меню выберите пункт *New - Class Diagram* (Создать – Диаграмма классов).
3. Назовите новую диаграмму Классов: Ввод нового заказа.
4. Дважды щелкнув мышью на этой диаграмме в браузере, откройте ее.
5. Перетащите из браузера все классы (*Выбор_заказа*, *Заказ_деталей*, *упр_заказами*, *Заказ*, *Упр_транзакциями*).

Объединение классов в пакеты

1. В браузере перетащите класс *Выбор_заказа* на пакет Границы.
2. Перетащите класс *Заказ_деталей* на пакет Границы.
3. Перетащите классы *Упр_заказами* и *Упр-транзакциями* на пакет Управление.
4. Перетащите класс *Заказ* на пакет Сущности.

Классы и пакеты в браузере показаны на рис. 4.



Рис.4. Представление пакетов и классов

Добавление диаграмм Классов к каждому пакету.

1. В браузере щелкните правой кнопкой мыши на пакете *Границы*.
2. В открывшемся меню выберите пункт *New - Class Diagram* (Создать - Диаграмма классов).
3. Введите имя новой диаграммы — *Main* (Главная).
4. Дважды щелкнув мышью на этой диаграмме, откройте ее.
5. Перетащите на нее из браузера классы *выбор_заказа* и *заказ_деталей*.
6. Закройте диаграмму.
7. В браузере щелкните правой кнопкой мыши на пакете *Сущности*.
8. В открывшемся меню выберите пункт *New - Class Diagram* (Создать - Диаграмма классов).
9. Введите имя новой диаграммы — *Main* (Главная).
10. Дважды щелкнув мышью на этой диаграмме, откройте ее.
11. Перетащите на нее из браузера класс *Заказ*.
12. Закройте диаграмму
13. В браузере щелкните правой кнопкой мыши на пакете *Управление*
14. В открывшемся меню выберите пункт *New - Class Diagram* (Создать - Диаграмма классов).
15. Введите имя новой диаграммы — *Main* (Главная).
16. Дважды щелкнув мышью на этой диаграмме, откройте ее.
17. Перетащите на нее из браузера классы *Упр_заказами* и *Упр_транзакциями*
18. Закройте диаграмму.

Задание 4. Уточнение методов и свойств классов.

В этом упражнении к описаниям операций будут добавлены детали, включая параметры и типы возвращаемых значений, и определены атрибуты классов.

Постановка проблемы

Для определения атрибутов классов был проанализирован поток событий. В результате к классу *Заказ* диаграммы Классов были добавлены атрибуты *Номер заказа* и *Имя клиента*. Так как в одном заказе можно указать большое количество товаров и у каждого из них имеются свои собственные данные и поведение, было решено моделировать товары как самостоятельные классы, а не как атрибуты класса *Заказ*.

Добавление атрибутов и операций

Добавим атрибуты и операции к классам диаграммы классов "Ввод нового заказа". При этом используем специфические для языка особенности. Установим параметры так, чтобы показывать все атрибуты, все операции и их сигнатуры. Применим нотацию UML.

Порядок выполнения задания:

Настройка

1. В меню модели выберите пункт *Tools - Options* (Инструменты - Параметры).
2. Перейдите на вкладку *Diagram*.
3. Убедитесь, что флажок *Show visibility* (Показать видимость) установлен.
4. Убедитесь, что флажок *Show stereotypes* (Показать стереотипы) установлен.
5. Убедитесь, что флажок *Show operation signatures* (Показать сигнатуры операций) установлен.
6. Убедитесь, что флажки *Show all attributes* (Показать все атрибуты) и *Show all operations* (Показать все операции) установлены.
7. Убедитесь, что флажки *Suppress attributes* (Подавить атрибуты) и *Suppress operations* (Подавить операции) сброшены.
8. Перейдите на вкладку *Notation* (Нотация).
9. Убедитесь, что флажок *Visibility as icons* (Отображать пиктограммы) сброшен.

Добавление нового класса

1. Найдите в браузере диаграмму классов варианта использования "Ввести новый заказ".
2. Дважды щелкнув мышью на диаграмме, откройте ее.
3. Нажмите кнопку *Class* панели инструментов.
4. Щелкните мышью внутри диаграммы, чтобы поместить туда новый класс.
5. Назовите его *Позиц_заказа*.
6. Назначьте этому классу стереотип *Entity*.
7. В браузере перетащите класс в пакет *Сущности*.

Добавление атрибутов

1. Щелкните правой кнопкой мыши на классе *Заказ*.
2. В открывшемся меню выберите пункт *New Attribute* (Создать атрибут).
3. Введите новый атрибут: *OrderNumber: Integer*
4. Нажмите клавишу *Enter*

5. Введите следующий атрибут: *CustomerName: String*.
6. Повторив шаги 4 и 5, добавьте атрибуты: *OrderDate: Date* и *OrderFillDate: Date*

Если тип атрибута не появляется в выпадающем списке, то введите его от руки и он далее будет появляться.

7. Щелкните правой кнопкой мыши на классе *Позиц_заказа*.
8. В открывшемся меню выберите пункт *New Attribute* (Создать атрибут).
9. Введите новый атрибут: *ItemID : Integer*.
10. Нажмите клавишу *Enter*.
11. Введите следующий атрибут: *ItemDescription: String*.

Добавление операций к классу *Позиц_заказа*

1. Щелкните правой кнопкой мыши на классе *Позиц_заказа*.
2. В открывшемся меню выберите пункт *New Opration* (Создать операцию).
3. Введите новую операцию: *Создать()*
4. Нажмите клавишу *Enter*.
5. Введите следующую операцию: *Взять_информацию()*
6. Нажмите клавишу *Enter*.
7. Введите операцию: *Дать_информацию()*

Подробное описание операций с помощью диаграммы классов

1. Щелкнув мышью на классе *Заказ*, выделите его.
2. Щелкните на этом классе еще раз, чтобы переместить курсор внутрь.
3. Отредактируйте операцию *Создать()*, чтобы она выглядела следующим образом: *Создать(): Boolean*
4. Отредактируйте операцию *Взять_информацию*:
Взять_информацию (OrderNum: Integer, Customer: String, OrderDate: Date, FillDate: Date): Boolean
5. Отредактируйте операцию *Дать_информацию*; *Дать_информацию(): String*

Подробное описание операций с помощью браузера

1. Найдите в браузере класс *Позиц_заказа*.
2. Раскройте этот класс, щелкнув на значке "+" рядом с ним. В браузере появятся атрибуты и операции класса.
3. Дважды щелкнув мышью на операции *Дать_информацию()*, откройте окно ее спецификации
4. В раскрывающемся списке *Return class* (Возвращаемый класс) укажите *String*.
5. Щелкнув мышью на кнопке ОК, закройте окно спецификации операции.
6. Дважды щелкните в браузере на операции *Дать_информацию ()* класса *Позиц_заказа*, чтобы открыть окно ее спецификации.
7. В раскрывающемся списке *Return class* укажите *Boolean*.
8. Перейдите на вкладку *Detail* (Подробно).
9. Щелкните правой кнопкой мыши в области аргументов, чтобы добавить туда новый параметр
10. В открывшемся меню выберите пункт *Insert* (Вставить). Rose добавит аргумент под названием *argname*.

11. Щелкнув один раз на этом слове, выделите его и измените имя аргумента на *ID*.
12. Щелкните на колонке *Type* (Тип). В раскрывающемся списке типов выберите *Integer* (Если этого либо иного необходимого типа не будет-введите его вручную).
13. Щелкните на колонке *Default* (По умолчанию), чтобы добавить значение аргумента по умолчанию. Введите число 0.
14. Нажав на кнопку ОК, закройте окно спецификации операции.
15. Дважды щелкните на операции *Создать()* класса *Позиц_заказа*, чтобы открыть окно ее спецификации.
16. В раскрывающемся списке *Return class* укажите *Boolean*.
17. Нажав на кнопку ОК, закройте окно спецификации операции.

Подробное описание операций

1. Используя браузер или диаграмму Классов, введите следующие сигнатуры операций класса *Заказ_деталей*:

Открыть(): Boolean

Сохранить заказ(): Boolean

2. Используя браузер или диаграмму Классов, введите сигнатуру операций класса *Выбор_заказа*:

Создать(): Boolean

3. Используя браузер или диаграмму Классов, введите сигнатуру операций класса *Упр_заказами*:

Сохранить заказ(OrderID: Integer): Boolean

4. Используя браузер или диаграмму Классов, введите сигнатуры операций класса *Упр_транзакциями*:

Сохранить заказ(OrderID: Integer): Boolean

Сохранить информацию(): Integer

Задание 5. Описание связей между классами

В этом упражнении определяются связи между классами, участвующими в варианте использования "Ввести новый заказ".

Постановка задачи

Чтобы найти связи, были просмотрены диаграммы Последовательности. Все взаимодействующие там классы нуждались в определении соответствующих связей на диаграммах Классов. После обнаружения связи были добавлены на диаграммы классов.

Добавим связи к классам, принимающим участие в варианте использования "Ввести новый заказ".

Порядок выполнения задания:

1. Найдите в браузере диаграмму Классов "Ввод нового заказа"
2. Дважды щелкнув на диаграмме, откройте ее.
3. Проверьте, имеется ли в панели инструментов диаграммы кнопка *Unidirectional Association* (Однонаправленная ассоциация). Если ее нет, продолжите настройку, выполнив шаги 4 и 5. Если есть, приступайте к выполнению самого упражнения.

4. Щелкните правой кнопкой мыши на панели инструментов диаграммы и в открывшемся меню выберите пункт *Customize* (Настроить),
5. Добавьте на панель кнопку *Creates A Unidirectional Association* (Создать однонаправленную ассоциацию).

Добавление ассоциаций

1. Нажмите кнопку *Unidirectional Association* панели инструментов.
2. Проведите ассоциацию от класса *выбор_заказа* к классу *заказ_деталей*.
3. Повторите шаги 1 и 2, создав ассоциации:
 - От класса *заказ_деталей* к классу *упр_заказами*
 - От класса *упр_заказами* к классу *Заказ*
 - От класса *упр_заказами* к классу *упр_транзакциями*
 - От класса *упр_транзакциями* к классу *Заказ*
 - От класса *упр_транзакциями* к классу *Позиц_заказа*
 - От класса *Заказ* к классу *Позиц_заказа*
4. Щелкните правой кнопкой мыши на однонаправленной ассоциации между классами *выбор_заказа* и *заказ_деталей* класса *выбор_заказа*.
5. В открывшемся меню выберите пункт *Multiplicity - Zero or One* (Множественность - Нуль или один). Щелкните правой кнопкой мыши на другом конце однонаправленной ассоциации.
6. В открывшемся меню выберите пункт *Multiplicity - Zero or One* (Множественность - Нуль или один)
7. Повторите шаги 4—6, добавив на диаграмму значения множественности для остальных ассоциаций, как показано на рис. 5

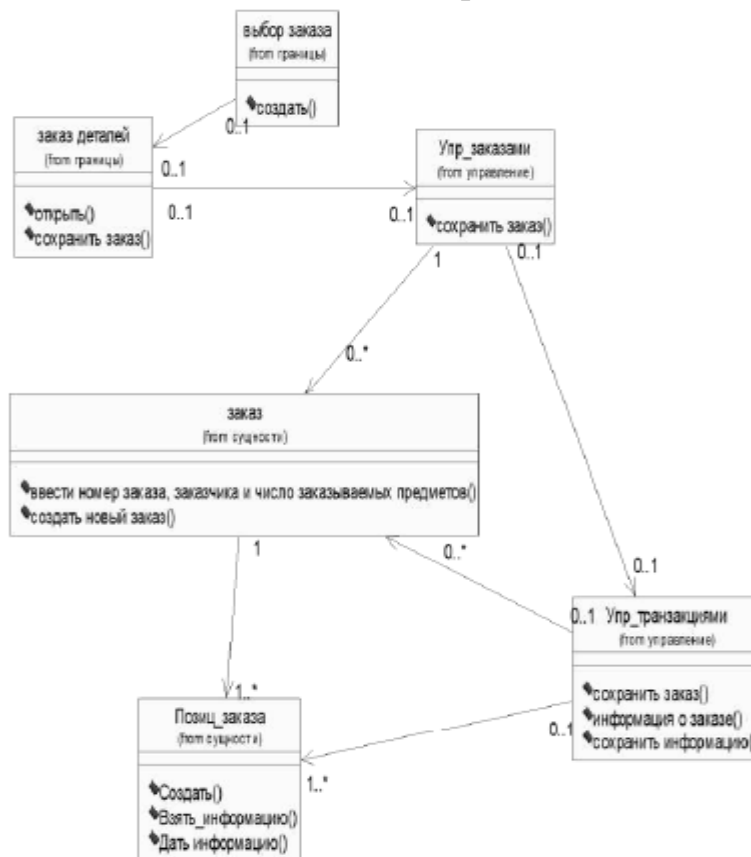


Рис.5. Ассоциации сценария «Ввести новый заказ»

Задание 6. Исключение кириллизированного текста в информации классов.

Разработанные ранее модели, предназначенные для описания предметной области используют кириллизированную символику, недопустимую для большинства языков программирования. Выполните замену русского текста на латинский шрифт. Для этой цели сохраните предыдущую модель под другим именем и далее работайте с новым файлом (что бы при необходимости можно было бы вернуться к бизнес- процессам, описанным русским шрифтом).

Порядок выполнения задания:

Этап 1. Используя меню (*Файл - Сохранить как*) сохраните данную модель под другим именем (например Заказ1) в той же папке, что и исходная модель. Работайте далее с копией модели (то есть Заказ1).

Этап 2. Переименуйте классы и их спецификации таким образом, чтобы использовался только латинский шрифт. Замените имя класса

Заказ_деталей на *OrderDetail*

Выбор_заказа на *OrderOptions*

Заказ на *Order*

Упр_заказами на *OrderMgr*

Позиц_заказа на *OrderItem*

Упр_транзакциями на *TransactionMgr*

Измените имена операций таким образом, чтобы рис.5 преобразовался в рис. 6. Для этого, измените операцию класса *OrderOptions*

Открыть() на *Open()*

Класса *OrderDetail*

Открыть() на *Open()*

Сохранить заказ() на *Save()*

Класса *Order*

Ввести номер заказа, заказчика и число заказываемых предметов() на *SetInfo()*

Сохранить_заказ() на *Save()*

Класса *OrderMgr*

Сохранить заказ() на *SaveOrder()*

Класса *TransactionMgr*

Сохранить заказ() на *SaveOrder()*

Сохранить информацию о заказе() на *Commit()*

Создать_заказ() на *SubmitInfo()*

Класса *OrderItem*

Создать() на *Create()*

Взять_информацию() на *GetInfo()*

Дать_информацию на *SetInfo()*

Переименуйте имена пакетов

Границы на *Boundaries*

Сущности на *Entity*

Контроль на *Control*

Добавление стереотипов к классам

1. Щелкните правой кнопкой мыши на классе *OrderOptions* диаграммы.
2. В открывшемся меню выберите пункт *Open Specification* (Открыть спецификацию).
3. В поле стереотипа выберите из выпадающего списка слово *Boundary* (если его нет, то введите).
4. Нажмите на кнопку ОК.
5. Повторив шаги 1—4, свяжите классы *OrderDetail* со стереотипом *Boundary*, *OrderMgr* и *TransactionMgr* со стереотипом *Control*, а класс *Order* и *OrderItem* со стереотипом *Entity*.

Теперь диаграмма Классов должна иметь вид, показанный на рис. 6.

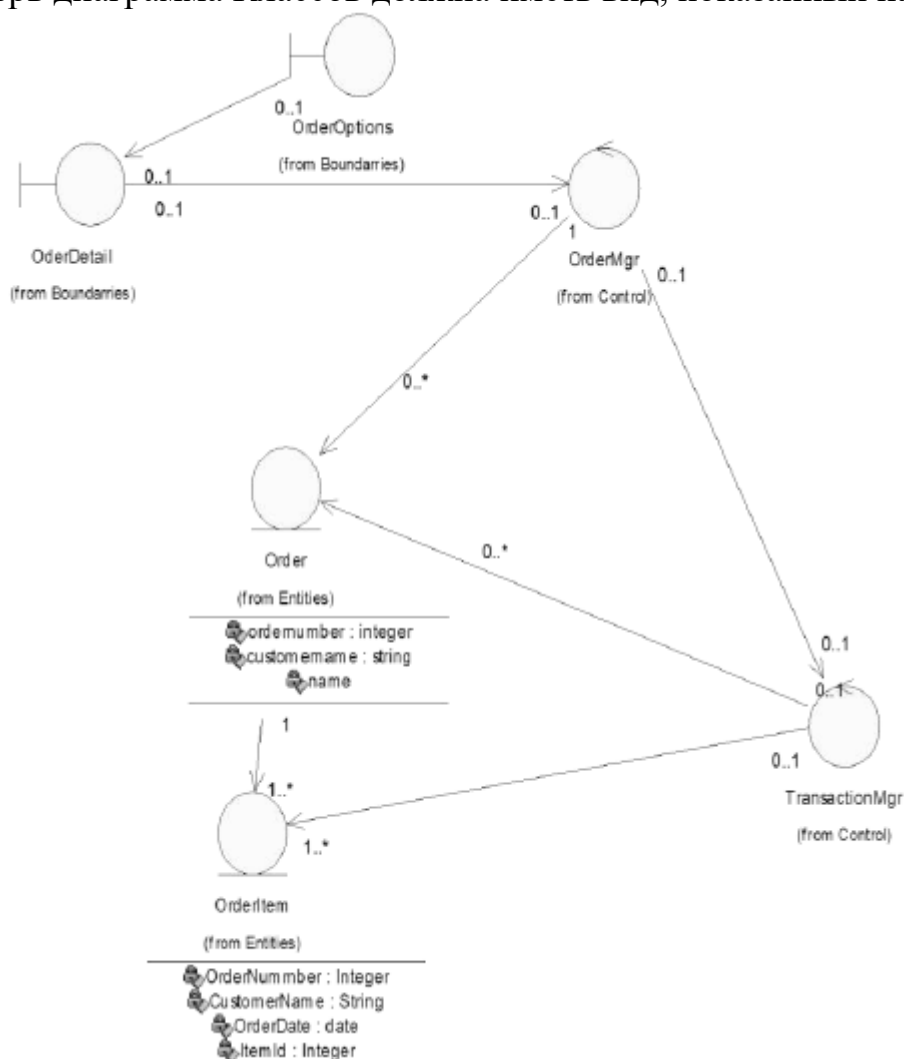


Рис.6. Основная диаграмма классов

Замечание. На диаграмме рис. 6 возможно визуальное представление классов не в виде иконок, а в виде дополнительной строки текста с именем стереотипа. За этот вид отвечает метка установленная либо на *icon* либо на *label* (Class – Open Specification – Options – Label).

ПРАКТИЧЕСКАЯ РАБОТА № 6

Тема: Построение диаграммы компонентов

Цель работы: исследование процесса построения диаграммы компонентов и в заданной предметной области.

Оборудование: ПК, программное обеспечение – Model Maker, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Построение диаграммы компонентов

Порядок выполнения задания:

Так как эта модель связана с конкретным языком программирования, то в настройках это необходимо отметить. Выполнить *Tools – Options – Notations – Default Language* и из выпадающего списка языков программирования выбрать *Delphi*.

Создание пакетов компонентов

1. Щелкните правой кнопкой мыши на представлении компонентов в браузере.
2. В открывшемся меню выберите пункт *New - Package* (Создать – Пакет).
3. Назовите пакет *Entities* (Сущности).
4. Повторив шаги с первого по третий, создайте пакеты *Boundaries* (Границы) и *Control* (Управление).

Добавление пакетов на Главную диаграмму компонентов

1. Откройте Главную диаграмму компонентов, дважды щелкнув на ней мышью
2. Перетащите пакеты *Entities*, *Boundary* и *Control* из браузера на Главную диаграмму.

Отображение зависимостей между пакетами

1. Нажмите кнопку *Dependency* (Зависимость) панели инструментов.
2. Щелкните мышью на пакете *Boundary* Главной диаграммы компонентов.
3. Проведите линию зависимости к пакету *Control*.
4. Повторив шаги 1 — 3, проведите зависимость от пакета *Control* к пакету *Entities*.

В результате диаграмма примет вид рис. 1

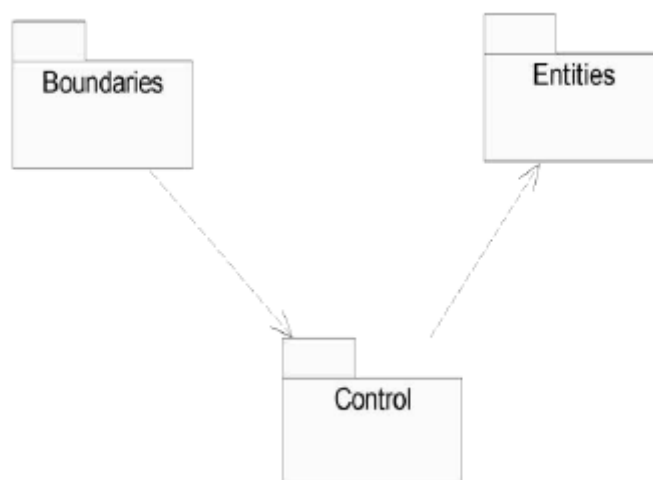


Рис. 1. Зависимости между пакетами

Добавление компонентов к пакетам и отображение зависимостей.

1. Дважды щелкнув мышью на пакете *Entities* Главной диаграммы компонентов, откройте Главную диаграмму компонентов этого пакета.
2. Нажмите кнопку *Package Specification* (Спецификация пакета) панели инструментов.
3. Поместите спецификацию пакета на диаграмму.
4. Введите имя спецификации пакета — *OrderItem_*.
5. Повторив шаги 2—4, добавьте спецификацию пакета *Order_*.
6. Нажмите кнопку *Dependency* (Зависимость) панели инструментов.
7. Щелкните мышью на спецификации пакета *OrderItem_*.
8. Проведите линию зависимости к спецификации пакета *OrderItem_*.
9. С помощью описанного метода создайте следующие компоненты и зависимости:

Для пакета *Boundaries*:

- Спецификацию пакета *Orderoptions_*
- Спецификацию пакета *OrderDetail_*

Зависимости в пакете *Boundaries*:

- От спецификации пакета *Orderoptions_* к спецификации пакета *OrderDetail_*

Для пакета *Control*:

- Спецификацию пакета *OrderMgr_*
- Спецификацию пакета *TransactionMgr_*

Зависимости в пакете *Control*:

- От спецификации пакета *OrderMgr_* к спецификации пакета *TransactionMgr_*

Создание диаграммы компонентов системы

1. Щелкните правой кнопкой мыши на представлении компонентов в браузере.
2. В открывшемся меню выберите пункт *New - Component Diagram* (Создать – Диаграмма компонентов).
3. Назовите новую диаграмму *System*.
4. Дважды щелкните на этой диаграмме мышью.

Размещение компонентов на диаграмме компонентов системы

1. Разверните в браузере пакет компонентов *Entities*, чтобы открыть его.
2. Щелкните мышью на спецификации пакета *Order_* в пакете компонентов *Entities*.
3. Перетащите эту спецификацию на диаграмму.
4. Повторив шаги 2 и 3, поместите на диаграмму спецификацию пакета *OrderItem_*.
5. С помощью этого метода поместите на диаграмму следующие компоненты:

Из пакета компонентов *Boundaries*:

- Спецификацию пакета *Orderoptions_*
- Спецификацию пакета *OrderDetail_*

Из пакета компонентов *Control*:

- Спецификацию пакета *OrderMgr_*
 - Спецификацию пакета *TransactionMgr_*
6. Нажмите кнопку *Task Specification* (Спецификация задачи) панели инструментов.
 7. Поместите на диаграмму спецификацию задачи и назовите ее *OrderClientExe*.
 8. Повторите шаги 6 и 7 для спецификации задачи *OrderServerExe*.

Уже существующие зависимости будут автоматически показаны на диаграмме компонентов системы после добавления туда соответствующих компонентов. Теперь нужно добавить остальные зависимости.

1. Нажмите кнопку *Dependency* (Зависимость) панели инструментов.
2. Щелкните мышью на спецификации пакета *OrderDetail_*
3. Проведите линию зависимости к спецификации пакета *OrderDetail_*
4. Повторив шаги 1 — 3, создайте следующие зависимости:
 - От спецификации пакета *OrderMgr_* к спецификации пакета *Order_*
 - От спецификации пакета *TransactionMgr_* к спецификации пакета *OrderItem_*
 - От спецификации пакета *TransactionMgr_* к спецификации пакета *Order_*
 - От спецификации задачи *OrderClientExe* к спецификации пакета *Orderoptions_*
 - От спецификации задачи *OrderServerExe* к спецификации пакета *OrderMgr_r*

Соотнесение классов с компонентами

1. В Логическом представлении браузера найдите класс *Order* пакета *Entities*.
2. Перетащите этот класс на спецификацию пакета компонента *Order_* в представлении компонентов браузера. В результате класс *Order* будет соотнесен со спецификацией пакета компонента *Order_*.
3. Повторив шаги 1 — 2, соотнесите классы со следующими компонентами:
 - Класс *OrderItem* со спецификацией пакета *OrderItem_*
 - Класс *Orderoptions* со спецификацией пакета *Orderoptions_*
 - Класс *OrderDetail* со спецификацией пакета *OrderDetail_*
 - Класс *OrderMgr* со спецификацией пакета *OrderMgr_*
 - Класс *TransactionMgr* со спецификацией пакета *TransactionMgr_*

В результате в браузере после имени класса, в скобках появятся имена компонентов, с которыми этот класс связан (рис. 2)

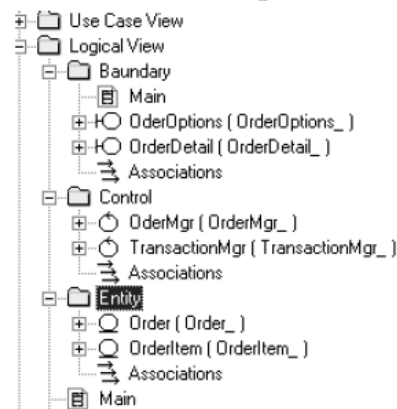


Рис.2. Представление компонентов и классов в браузере

ПРАКТИЧЕСКАЯ РАБОТА № 7

Тема: Построение диаграмм потоков данных

Цель работы: исследование процесса построения диаграммы потоков данных и диаграммы размещения в заданной предметной области с помощью Model Maker и Delphi.

Оборудование: ПК, программное обеспечение – Model Maker, MS Word, инструкции по выполнению работы.

Содержание работы:

Задание 1. Кодогенерация проекта в Delphi

Теперь вся информация подготовлена к тому, чтобы запрограммировать классы с их методами и операциями.

Для выполнения кодогенерации в среде Delphi необходимо выполнить следующую последовательность действий:

- протестировать модель на логические непротиворечия;
- настроить (или проверить настройки) среду на законы кодогенерации (соответствие элемента модели Rose элементу кода Delphi);
- создать имя проекта Delphi и выполнить кодогенерацию.

Порядок выполнения задания:

1) Протестируйте модель *Tools - Ceck Model*. Просмотрите log-файл на наличие ошибок. Если файл не виден, то выполните команду *file - Save Log As* и введите имя файла (по умолчанию error.log). Затем его просмотрите и, при необходимости, исправьте ошибки. К наиболее распространенным ошибкам относятся такие, как неотображение сообщений на операции или несоотнесение объектов с классами на диаграммах взаимодействия. С помощью пункта меню *Check Model* можно выявить большую часть неточностей и ошибок в модели.

2) Пункт меню *Access Violations* позволяет обнаружить нарушения правил доступа, возникающие тогда, когда существует связь между двумя классами разных пакетов. При этом связи между самими пакетами нет. Например, если существует связь между классами *Order* пакета *Entities* и *OrderManager* пакета *Control*, то обязательно должна существовать и связь между пакетами *Entities* и *Control*. Если последняя связь не установлена, Rose выявит нарушение правил доступа. Чтобы обнаружить нарушение правил доступа, выберите в меню *Report - Show Access Violations*.

Проанализируйте все нарушения правил доступа.

3) Выполните *Tools – Options – Notation – Default Language* и из выпадающего списка выберите язык программирования Delphi.

4) Проверьте правильность установок кодогенерации по умолчанию (default). Для этого выполните *Tools – Options – Delphi* и последовательно переберите из выпадающего списка поля *Type* все элементы. Сравните установки в поле *Model Propities* с данными (default) из таблицы Приложения А. В случае несоответствия, исправьте.

5) Выполните *Tools – Ensemble Tools – Rose Delphi Link* (рис.1)

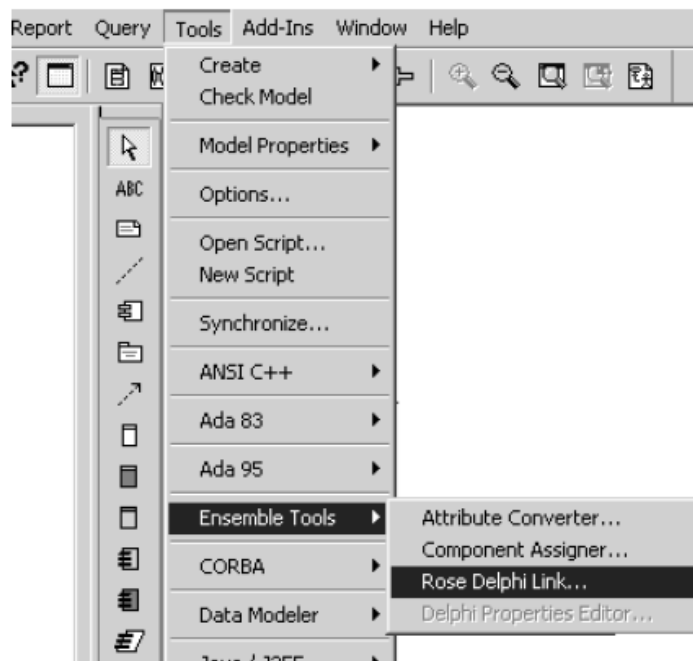


Рис. 1. Меню для выбора процесса кодогенерации

В результате появится соответствующая экранная форма. Выполните на этой форме *File – New Proect*. Появится форма с браузером. Введите имя файла и место на диске, куда будет сохранено имя сгенерированного проекта в Delphi. Например, *NewProect.dpr* и нажмите *Открыть*.

В результате форма примет вид (рис. 2)

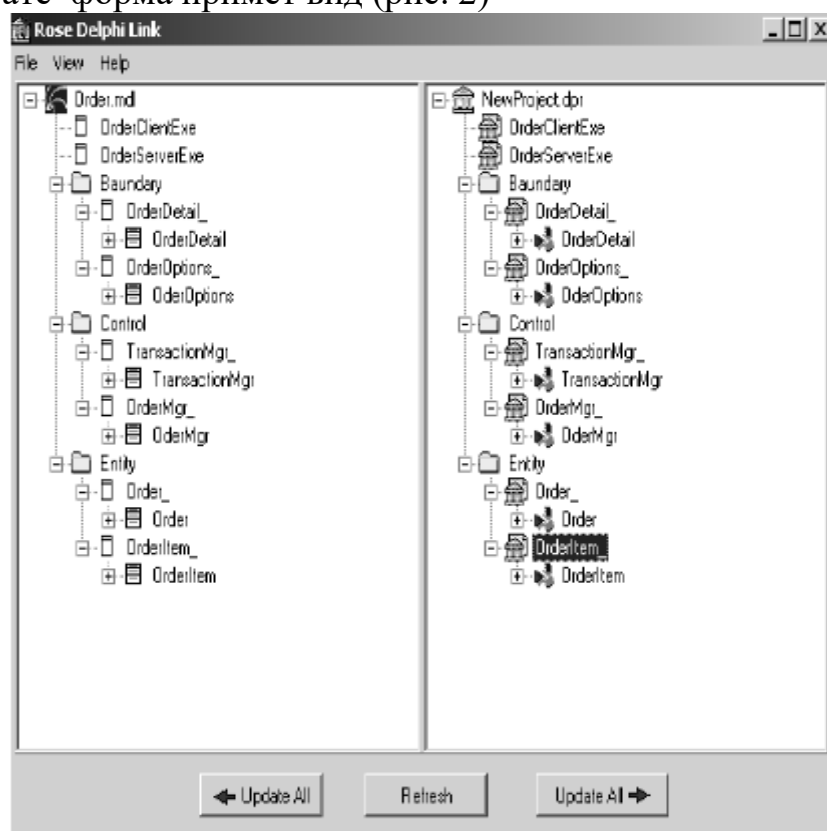


Рис. 2. Представление результатов кодогенерации

5. Через проводник Windows найдите папку проекта Delphi. С помощью программы Блокнот просмотрите содержимое всех файлов. В приложении В к руководству приведено содержимое всех файлов проекта.

Задание 2. Анализ Delphi проекта, добавление визуальных объектов, реинжиниринг в Rose

- 1) Запустите на выполнение программу Delphi и загрузите сгенерированный проект (Proect1.dpr). Проверьте, что проект содержит все модули и присмотрите их содержимое через редактор Delphi.
- 2) Создайте в проекте Delphi новую форму с *Name Form1*. Поместите на форму компонент *MainMenu* (главное меню)
- 3) С помощью *Menu Disigner* введите две позиции горизонтального меню с названиями (полями Caption) *Oder* и *OderItem*.
- 4) Для *Oder* введите две строки вертикального меню с *Caption Create* и *SubmitInfo*. Для *OderItem* введите одну строку вертикального меню с названием *GetInfo*.
- 5) Сохраните проект в Delphi.

Реинжиниринг Delphi проекта в модель Rose

- 1) Вернитесь в проект Rose и откройте окно проектов Rose Delphi Link. Проверьте, что открыт именно тот проект, для которого выполнялась кодогенерация.
- 2) Курсором мыши нажмите клавишу Update ALL со стрелкой влево (обновление модели Rose на основе изменений проекта Delphi). В результате в модели Rose должны произойти определенные изменения (рис. 3)

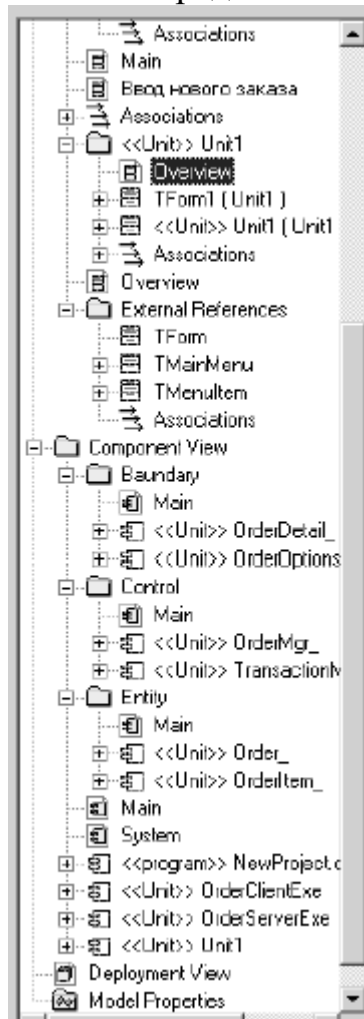


Рис.3. Окно Rose Delphi Link после кодогенерации

- в представление *Logic View* созданся новый пакет *Unit1* и *External References* (Внешние ссылки). Внутри второго пакета созданы три класса *TForm*, *TMainMenu* и *TMenuItem*, которые использовались при развитии проекта Delphi. Отметим, что эта папка не создалась бы, если бы мы при первоначальном создании проекта включили в него пакет классов Delphi FreimWork.
- в этом же представлении в пакете *Unit1* создан класс *TForm1* и *Unit1* оба соотнесенные с вновь созданным компонентом Unit1. Кроме того, в этом же пакете создалась диаграмма классов Overview, содержимое которой показано на рис. 4

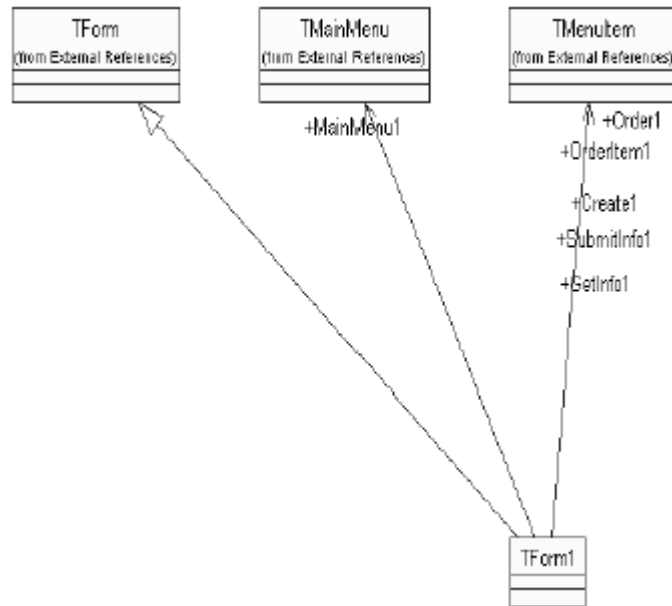


Рис.4. Результаты реинжиниринга проекта Delphi в Rose

Задание 3. Построение диаграммы размещения

В этом задании создается диаграмма размещения для системы обработки заказов.

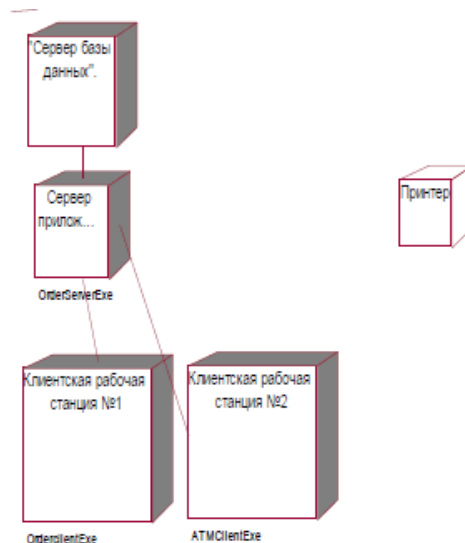


Рис. 5.Диаграмма размещения для модельной задачи

Порядок выполнения задания:

Добавление узлов к диаграмме размещения

1. Дважды щелкнув мышью на представлении размещения в браузере, откройте диаграмму размещения.
2. Нажмите кнопку *Processor* (Процессор) панели инструментов.
3. Щелкнув мышью на диаграмме, поместите туда процессор.
4. Введите имя процессора "Сервер базы данных".
5. Повторив шаги 2—4, добавьте следующие процессоры:
 - Сервер приложения
 - Клиентская рабочая станция №1
 - Клиентская рабочая станция №2
6. На панели инструментов нажмите кнопку *Devices* (Устройство).
7. Щелкнув мышью на диаграмме, поместите туда устройство.
8. Назовите его "Принтер".

Добавление связей

1. Нажмите кнопку *Connection* (Связь) панели инструментов.
2. Щелкните мышью на процессоре "Сервер базы данных".
3. Проведите линию связи к процессору "Сервер приложения".
4. Повторив шаги 1 — 3, добавьте следующие связи;
 - От процессора "Сервер приложения" к процессору "Клиентская рабочая станция № 1"
 - От процессора "Сервер приложения" к процессору "Клиентская рабочая станция № 2"
 - От процессора "Сервер приложения" к устройству "Принтер"

Добавление процессов

1. Щелкните правой кнопкой мыши на процессоре "Сервер приложения" в браузере.
2. В открывшемся меню выберите пункт *New - Process* (Создать - Процесс),
3. Введите имя процесса — *OrderServerExe*.
4. Повторив шаги 1 - 3, добавьте процессы:
 - Процесс *OrderclientExe* на процессоре "Клиентская рабочая станция №1"
 - Процесс *ATMClientExe* на процессоре "Клиентская рабочая станция №2"

Показ процессов на диаграмме

1. Щелкните правой кнопкой мыши на процессоре "Сервер приложения".
2. В открывшемся меню выберите пункт *Show Process* (Показать процессы).
3. Повторив шаги 1 и 2, покажите процессы на следующих процессорах:
 - Клиентская рабочая станция №1
 - Клиентская рабочая станция №2

ПРАКТИЧЕСКАЯ РАБОТА № 8

Тема: Разработка тестового сценария. Оценка необходимого количества тестов

Цели работы: усвоить знание о видах тестирования; освоить способы обнаружения и фиксирования ошибок; получить навыки разработки тестовых сценариев.

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

Содержание работы:

Общепринятая практика состоит в том, что после завершения продукта и до передачи его заказчику независимой группой тестировщиков проводится тестирование ПО.

Уровни тестирования:

Модульное тестирование. Тестируется минимально возможный для тестирования компонент, например отдельный класс или функция;

Интеграционное тестирование. Проверяется, есть ли какие-либо проблемы в интерфейсах и взаимодействии между интегрируемыми компонентами, например, не передается информация, передается некорректная информация;

Системное тестирование. Тестируется интегрированная система на ее соответствие исходным требованиям.

Таблица 1. Виды некоторых ошибок и способы их обнаружения

Виды программных ошибок	Способы их обнаружения
Ошибки выполнения, выявляемые автоматически: а) переполнение, защита памяти; б) несоответствие типов; в) заикливание	Динамически й контроль: Аппаратурой процессора; run-time системы программирования; операционной системой – по превышению лимита времени

Тест – это набор контрольных входных данных совместно с ожидаемыми результатами. Тесты должны обладать определенными свойствами.

Детективность: тест должен с большой вероятностью обнаруживать возможные ошибки.

Покрывающая способность: один тест должен выявлять как можно больше ошибок.

Воспроизводимость: ошибка должна выявляться независимо от изменяющихся условий.

С помощью тестирования разных видов обнаруживаются ошибки в разрабатываемом программном обеспечении. После обнаружения ошибок проводится их устранение.

- Оценка стоимости и причины ошибок в программном обеспечении.
- Виды и методы тестирования.
- Понятие теста.
- Требования к разработке тестовых сценариев.

- Правила разработки тестовых сценариев.

Содержание работы:

Задание 1. Создать приложение Простой калькулятор, в котором реализовать выполнение простых операций с вводимыми двумя операндами. Выполнить тестирование приложения на различных данных, отличающихся по типу и значению.

Порядок выполнения задания:

1. Разработать интерфейс приложения и написать программные коды для событий кнопок.
2. Сохранить проект в отдельной папке, скопировать исполняемый файл на рабочий стол.
3. Составить тесты для проверки работы приложения.
4. Провести тестирование исполняемого файла

Составить отчет по итогам тестирования и рекомендации по устранению выявленных ошибок

Задание 2. Написать программу решения квадратного уравнения $ax^2 + bx + c = 0$.

Задание 3. Найти минимальный набор тестов для программы нахождения вещественных корней квадратного уравнения $ax^2 + bx + c = 0$. Решение представлено в таблице.

Но- мер теста	a	b	c	Ожидаемый результат	Что проверяется
1	2	-5	2	$x_1=2, x_2=0,5$	Случай вещественных корней
2	3	2	5	Сообщение	Случай комплексных корней
3	3	-12	0	$x_1=4, x_2=0$	Нулевой корень
4	0	0	10	Сообщение	Неразрешимое уравнение
5	0	0	0	Сообщение	Неразрешимое уравнение
6	0	5	17	Сообщение	Неквадратное уравнение
7	9	0	0	$x_1=x_2=0$	Нулевые корни

Таким образом, для этой программы предлагается минимальный набор функциональных тестов, исходя из 7 классов выходных данных.

Заповеди по отладки программного средства, предложенные Г. Майерсом.

Заповедь 1. Считайте тестирование ключевой задачей разработки ПС, поручайте его самым квалифицированным и одаренным программистам, нежелательно тестировать свою собственную программу.

Заповедь 2. Хорош тот тест, для которого высока вероятность обнаружить ошибку, а не тот, который демонстрирует правильную работу программы.

Заповедь 3. Готовьте тесты как для правильных, так и для неправильных данных.

Заповедь 4. Документируйте пропуск тестов через компьютер, детально изучайте результаты каждого теста, избегайте тестов, пропуск которых нельзя повторить.

Заповедь 5. Каждый модуль подключайте к программе только один раз, никогда не изменяйте программу, чтобы облегчить ее тестирование.

Заповедь 6. Пропускайте заново все тесты, связанные с проверкой работы какой-либо программы ПС или ее взаимодействия с другими программами, если в нее были внесены изменения (например, в результате устранения ошибки).

Задание 4. Разработайте набор тестовых сценариев (как позитивных, так и негативных) для следующей программы:

Имеется консольное приложение (разработайте самостоятельно). Ему на вход подается 2 строки. На выходе приложение выдает число вхождений второй строки в первую. Например:

Строка 1	Строка 2	Вывод
абвгабвг	аб	2
стстсап	стс	2

Набор тестовых сценариев запишите в виде таблицы, приведенной выше

Задание 5. Оформить отчет.

ПРАКТИЧЕСКАЯ РАБОТА № 9

Тема: Разработка тестовых пакетов

Цель работы: получить навыки разработки тестовых пакетов.

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

Содержание работы:

- Системные основы разработки требований к сложным комплексам программ.
- Формализация эталонов требований и характеристик комплекса программ.
- Формирование требований компонентов и модулей путем декомпозиции функций комплексов программ.
- Тестирование по принципу «белого ящика».

Содержание работы:

Задание 1. В Древней Греции (II в. до н.э.) был известен шифр, называемый "квадрат Полибия". Шифровальная таблица представляла собой квадрат с пятью столбцами и пятью строками, которые нумеровались цифрами от 1 до 5. В каждую клетку такого квадрата записывалась одна буква. В результате каждой букве соответствовала пара чисел, и шифрование сводилось к замене буквы парой чисел. Для латинского алфавита квадрат Полибия имеет вид:

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I, J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Пользуясь изложенным способом создать программу, которая:

- а) зашифрует введенный текст и сохранит его в файл;
- б) считает зашифрованный текст из файла и расшифрует данный текст.

Задание 2. Спроектировать тесты по принципу «белого ящика» для программы, разработанной в задании № 1. Выбрать несколько алгоритмов для тестирования и обозначить буквами или цифрами ветви этих алгоритмов. Выписать пути алгоритма, которые должны быть проверены тестами для выбранного метода тестирования. Записать тесты, которые позволят пройти по путям алгоритма. Протестировать разработанную вами программу. Результаты оформить в виде таблицы:

Тест	Ожидаемый результат	Фактический результат	Результат тестирования
...

Задание 3. Проверить все виды тестов и сделать выводы об их эффективности

Задание 4. Оформить отчет.

Информационное обеспечение обучения

Печатные и электронные издания:

Основные учебные издания:

1. Зубкова, Т. М. Технология разработки программного обеспечения: учебное пособие для СПО / Т. М. Зубкова. — Саратов: Профобразование, 2019. — 468 с. — ISBN 978-5-4488-0354-3. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/86208>

Дополнительные учебные издания:

2. Долженко, А. И. Технологии командной разработки программного обеспечения информационных систем: курс лекций / А. И. Долженко. — 3-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019. — 300 с. — ISBN 978-5-4486-0525-3. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/79723>
3. Сеницын, С. В. Основы разработки программного обеспечения на примере языка С: учебное пособие для СПО / С. В. Сеницын, О. И. Хлытчиев. — Саратов: Профобразование, 2019. — 212 с. — ISBN 978-5-4488-0362-8. — Текст: электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование: [сайт]. — URL: <https://profspo.ru/books/86201>

Интернет-ресурсы:

4. Учебники по программированию <http://programm.ws/index.php>

Электронно-библиотечная система:

5. ЭБС «IPRbooks», ООО «Ай Пи Ар Медиа»
6. ЭБС «Znanium»
7. ЭБС «PROФобразование»
8. ЭБС «Book.ru»