

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

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

Энгельсский технологический институт (филиал)



УТВЕРЖДАЮ
Зам. директора по СПДО
О.Г. Коваленко

**Методические указания
к выполнению практических занятий учебной дисциплины
ОП.04 Основы алгоритмизации и программирования**

по специальности:

09.02.07 Информационные системы и программирование

Методические указания
рассмотрены на заседании
предметной (цикловой) методической комиссии
специальности 09.02.07
«25» июня 2024 года, протокол № 11

Председатель ПЦМК  А.А. Сдобнова

Энгельс 2024

ОРГАНИЗАЦИЯ - РАЗРАБОТЧИК:

Энгельсский технологический институт (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Саратовский государственный технический университет имени Гагарина Ю.А.»

РАЗРАБОТЧИК: Андреева М.И. преподаватель спецдисциплин ОСПДО

Пояснительная записка

По учебному плану в соответствии с рабочей программой на изучение ОП.04 «Основы алгоритмизации и программирования» обучающимися предусмотрено аудиторных занятий - 197 часов, из них практических занятий – 115 часов. В методические указания включены 58 практических работ по темам курса. Каждая Практическое занятие содержит сведения о цели ее проведения и практическом использовании результатов исследования, необходимых для проведения работы, включает краткие теоретические сведения, этапы выполнения работы.

Целью практических занятий по ОП.04 «Основы алгоритмизации и программирования» является:

овладение студентами необходимыми знаниями и практическими навыками в области программирования

Планируемые результаты:

В результате освоения учебной дисциплины обучающийся должен уметь:

- Разрабатывать алгоритмы для конкретных задач.
- Использовать программы для графического отображения алгоритмов.
- Определять сложность работы алгоритмов.
- Работать в среде программирования.
- Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.
- Оформлять код программы в соответствии со стандартом кодирования.
- Выполнять проверку, отладку кода программы.

В результате освоения учебной дисциплины обучающийся должен знать:

- Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.
- Эволюцию языков программирования, их классификацию, понятие системы программирования.
- Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.
- Подпрограммы, составление библиотек подпрограмм
- Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения

Практические занятия

Номер и тема раздела	Номер практического занятия	Наименование темы занятия	Кол-во часов (ауд.)
1	2	3	4
Языки программирования	1	Создание первого проекта в среде разработки	2
	2	Организация ввода-вывода данных. Форматирование	2
Операторы языка программирования	3	Операции и выражения	2
	4	Программирование алгоритмов линейной структуры	2
	5	Программирование алгоритмов разветвляющейся структуры	2
	6	Программирование алгоритмов циклической структуры с заданным числом повторений	2
	7	Программирование алгоритмов циклической структуры с внутренней разветвляющейся структурой	2
	8	Обработка одномерных массивов	2
	9	Сумма элементов массива, максимальный и минимальный элемент массива	2
	10	Обработка двумерных массивов	2
	11	Сортировка массива метод «пузырька», вставками	2
	12	Сортировка массива алгоритм Шелла, быстрая сортировка	2
	13	Последовательный и двоичный поиск	2
Процедуры и функции	14	Программирование алгоритмов с использованием функций, не возвращающих значения	2
	15	Программирование алгоритмов с использованием функций, возвращающих значения	2

	16	Программирование рекурсивных алгоритмов	2
Модульное программирование	17	Программирование модуля	2
Символы и строки	18	Работа с символами. Основные методы	2
	19	Работа со строками. Строковый тип string	2
	20	Работа со строками. Строковый тип StringBuilder	2
Множество и записи	21	Работа с данными типа множество	2
	22	Работа с данными типа запись	2
Файлы	23	Составление программ с использованием текстовых файлов. Байтовый поток	2
	24	Составление программ с использованием текстовых файлов. Символьный поток	2
Указатели	25	Использование указателей для организации связанных списков	2
	26	Программирование алгоритмов с использованием стеков	2
	27	Программирование алгоритмов с использованием очередей	2
Основные принципы объектно-ориентированного программирования (ООП)	28	Создание класса. Создание экземпляров класса.	2
	29	Вызов методов класса	2
	30	Конструкторы класса	2
	31	Деструкторы класса	2
	32	Свойства класса	2
	33	Индексаторы класса	2
	34	Унарные операции класса	2
	35	Бинарные операции класса	2
	36	Создание класса со всеми членами класса по индивидуальному заданию	2
	37	Структуры	2
Интегрированная	38	Изучение интерфейса среды	2

среда разработчика		разработчика: характеристика, основные окна, инструменты, объекты.	
Визуальное событийно-управляемое программирование	39	Создание проекта с формой, настройка формы	2
	40	Создание проекта с использованием компонентов для отображения текстовой информации.	2
	41	Создание проекта с использованием компонентов для редактирования текстовой информации.	2
	42	Создание проекта с использованием компонентов для редактирования текстовой информации.	2
	43	Создание проекта с использованием компонентов выбора из списков	2
	44	Создание проекта с использованием компонентов ввода и отображения чисел	2
	45	Создание проекта с использованием компонентов ввода и отображения дат и времени.	2
	46	Создание проекта с использованием кнопочных компонентов.	2
	47	Создание проекта с переключателем.	2
	48	Создание проекта с индикаторами выбора	2
	49	Создание проекта с контейнерными элементами управления	2
	50	Создание проекта с использованием компонентов для отображения графической информации.	2

	51	Создание проекта с использованием окон диалога в формах	2
	52	Создание проекта с использованием компонентов стандартных диалогов и системы меню.	2
	53	Разработка функциональной схемы работы приложения	2
Этапы разработки приложений	54	Разработка многооконного приложения.	2
	55	Создание интерфейса пользователя.	2
	56	Создание процедур обработки событий. Компиляция и запуск приложения.	2
	57	Тестирование, отладка приложения.	2
Иерархия классов	58	Создание класса	1
		Всего	115

Практическое занятие № 1

Тема раздела: Языки программирования

Тема практического занятия: Создание первого проекта в среде разработки

Цель: изучить среду разработки MS Visual Studio

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Вывести на экран сообщение “Hello, world!”

Задание 2. Измените текст кода консольного приложения так, чтобы на экран выводилось сообщение «Привет! Я студент группы ИСП! Меня зовут (ваше имя)»

Задание 3. Изучите, чем метод Wride отличается от WriteLine

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 2

Тема раздела: Языки программирования

Тема практического занятия: Организация ввода-вывода данных. Форматирование

Цель: изучить средства форматирования, способы ввода-вывода данных

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;

- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Вывести на экран целочисленную переменную

Задание 2. Вывести на экран вещественную переменную, с точностью до 6 знаков

Задание 3. С помощью метода ReadLine() ввести данные

Задание 4. Вывести на экран переменные в заданном формате

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 3

Тема раздела: Операторы языка программирования

Тема практического занятия: Операции и выражения

Цель: изучить: операции инкремента и декремента, логические операции, арифметические операции и др.

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Написать программу, которая, реализует диалог с пользователем:

1. запрашивает с клавиатуры два целых числа, и выводит на экран сумму данных чисел
2. запрашивает с клавиатуры три целых числа, и выводит на экран сумму данных чисел в прямом и обратном порядке
3. запрашивает с клавиатуры три целых числа, и выводит на экран сумму данных чисел
4. запрашивает с клавиатуры два вещественных числа, и выводит на экран произведение данных чисел (вещественные числа выводятся с точностью до 1 знака после запятой)
5. запрашивает с клавиатуры два вещественных числа, и выводит на экран результат деления первого числа на второе (вещественные числа выводятся с точностью до 3 знаков после запятой)
6. запрашивает с клавиатуры три вещественных числа, и выводит на следующее сообщение (вещественные числа выводятся с точностью до 2 знаков после запятой)

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 4

Тема раздела: Операторы языка программирования

Тема практического занятия: Программирование алгоритмов линейной структуры

Цель: уметь программировать линейные алгоритмы

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Написать программу, которая подсчитывает:

7. периметр квадрата, площадь которого равна a ;

8. площадь равностороннего треугольника, периметр которого равен p ;
9. среднее арифметическое кубов двух данных чисел;
10. площадь прямоугольного треугольника по двум катетам a , b .
11. периметр прямоугольного треугольника по двум катетам a , b .
12. ребро куба, площадь полной поверхности которого равна s ;
13. ребро куба, объем которого равен v ;
14. периметр треугольника, заданного координатами вершин $x_1, y_1, x_2, y_2, x_3, y_3$;
15. площадь треугольника, заданного координатами вершин $x_1, y_1, x_2, y_2, x_3, y_3$;
16. радиус окружности, длина которой равна l ;
17. площадь равнобедренной трапеции с основаниями a и b и углом α при большем основании;
18. площадь кольца с внутренним радиусом r_1 и внешним r_2 ;
19. радиус окружности, вписанной в равносторонний треугольник со стороной a ;

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 5

Тема раздела: Операторы языка программирования

Тема практического занятия: Программирование алгоритмов разветвляющейся структуры

Цель: программировать алгоритмы разветвляющейся структуры

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;

- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Написать программу, которая определяет

1. максимальное значение для двух различных вещественных чисел;
2. минимальное значение для двух различных вещественных чисел;
3. является ли заданное целое двухзначное число нечетным;
4. является ли заданное целое двухзначное число четным;
5. какая из цифр двухзначного числа меньше: первая или вторая
6. какая из цифр двухзначного числа больше: первая или вторая;
7. оканчивается ли данное целое двухзначное число цифрой 7;
8. какая из цифр двухзначного числа больше: первая или вторая;
9. одинаковы ли цифры данного двухзначного числа;
10. является ли сумма цифр двухзначного числа четной;
11. кратна ли трем сумма цифр двухзначного числа;
12. кратна ли пяти сумма цифр двухзначного числа;
13. больше 30 произведение цифр двухзначного числа

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 6

Тема раздела: Операторы языка программирования

Тема практического занятия: Программирование алгоритмов циклической структуры с заданным числом повторений

Цель: программировать циклические алгоритмы

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Написать программу, которая выводит

1. целые числа 10, 12, 14, ..., 60 в обратном порядке в столбик;
2. таблицу соответствия между весом в фунтах и весом в килограммах для значений 1, 2, 3, ..., 10 фунтов (1 фунтов = 453г);
3. таблицу перевода 5, 10, 15, ..., 120 долларов США в рубли по текущему курсу (значение курса вводится с клавиатуры);
4. таблицу стоимости для 10, 20, 30, ..., 100 штук товара, при условии, что одна штука товара стоит x руб (значение x водится с клавиатуры);
5. таблицу перевода расстояний в дюймах в сантиметры для значений 2, 4, 6, ..., 12 дюймов (1 дюйм = 25.4 мм);
6. кубы всех целых чисел из диапазона от A до B ($A \leq B$) в обратном порядке;

Задание 2. Вывести на экран числа следующим образом:

2) 1 2 3 ... 10	3) -10 -9 -8 ... 12	
1 2 3 ... 10	-10 -9 -8 ... 12	
1 2 3 ... 10	-10 -9 -8 ... 12	
1 2 3 ... 10	-10 -9 -8 ... 12	
	-10 -9 -8 ... 12	
4) 41 42 43 ... 50	5) 5	6) 1 1 1 1 1
51 52 53 ... 60	5 5	1 1 1 1
61 62 63 ... 70	5 5 5	1 1 1
...	5 5 5 5	1 1
71 72 73 ... 80	5 5 5 5 5	1
7) 1	8) 6 6 6 6 6	9) 7
2 2	7 7 7 7	6 6
3 3 3	8 8 8	5 5 5
4 4 4 4	9 9	4 4 4 4
5 5 5 5 5	10	3 3 3 3 3
10) 8 8 8 8 8	11) 1	12) 1
7 7 7 7	1 2	2 1
6 6 6	1 2 3	3 2 1
5 5	1 2 3 4	4 3 2 1
4	1 2 3 4 5	5 4 3 2 1

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны

правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 7

Тема раздела: Операторы языка программирования

Тема практического занятия: Программирование алгоритмов циклической структуры с внутренней разветвляющейся структурой

Цель: программировать циклические алгоритмы

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Написать программу, которая выводит

1. все целые числа из диапазона от А до В ($A \leq B$), оканчивающиеся на цифру Х;
2. все целые числа из диапазона от А до В ($A \leq B$), оканчивающиеся на цифру Х или У;
3. все целые числа из диапазона от А до В ($A \leq B$), оканчивающиеся на любую четную цифру;
4. только положительные целые числа из диапазона от А до В ($A \leq B$);
5. все целые числа из диапазона от А до В, кратные трем ($A \leq B$);
6. все четные числа из диапазона от А до В, кратные трем ($A \leq B$);
7. только отрицательные четные числа из диапазона от А до В ($A \leq B$);
8. все двухзначные числа, в записи которых все цифры разные;
9. все двухзначные числа, в которых старшая цифра отличается от младшей не больше чем на 1;
10. все трехзначные числа, которые начинаются и заканчиваются на одну и ту же цифру;
11. все трехзначные числа, в которых хотя бы две цифры повторяются.

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 8

Тема раздела: Операторы языка программирования

Тема практического занятия: Обработка одномерных массивов

Цель: освоить навыки обработки одномерных массивов

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Написать программу для одномерного массива

1. Заменить все элементы, меньшие заданного числа, этим числом.
2. Заменить все элементы, попадающие в интервал $[a, b]$, нулем.

3. Заменить все отрицательные элементы, не кратные 3, противоположными им числами.
4. Все элементы, меньшие заданного числа, увеличить в два раза.
5. Подсчитать среднее арифметическое элементов.
6. Подсчитать среднее арифметическое отрицательных элементов.
7. Подсчитать количество нечетных элементов.
8. Подсчитать сумму элементов, попадающих в заданный интервал.
9. Подсчитать сумму элементов, кратных 9.
10. Подсчитать количество элементов, не попадающих в заданный интервал.
11. Подсчитать сумму квадратов четных элементов.
12. Вывести на экран номера всех элементов больших заданного числа.
13. Вывести на экран номера всех нечетных элементов.
14. Вывести на экран номера всех элементов, которые не делятся на 7.
15. Вывести на экран номера всех элементов, не попадающих в заданный интервал.
16. Определить, является ли произведение элементов трехзначным числом.
17. Определить, является ли сумма элементов двухзначным числом.
18. Вывести на экран элементы с четными индексами (для двумерного массива – сумма индексов должна быть четной).
19. Вывести на экран положительные элементы с нечетными индексами (для двумерного массива – первый индекс должен быть нечетным).

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 9

Тема раздела: Операторы языка программирования

Тема практического занятия: Сумма элементов массива, максимальный и минимальный элемент массива

Цель: освоить навыки работы с элементами одномерного массива

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Написать программу для одномерного массива

1. Заменить все максимальные элементы нулями.
2. Заменить все минимальные элементы на противоположные.
3. Поменять местами максимальный элемент и первый.
4. Вывести на экран номера всех элементов, не совпадающих с максимальным.
5. Найти номер первого минимального элемента.
6. Найти номер последнего максимального элемента.
7. Подсчитать сумму элементов, расположенных между максимальным и минимальным элементами (минимальный и максимальный элементы в массиве единственные). Если максимальный элемент встречается позже минимального, то выдать сообщение об этом.
8. Найти номер первого максимального элемента.
9. Найти номер последнего минимального элемента.
10. Подсчитать сумму элементов, расположенных между первым максимальным и последним минимальными элементами. Если максимальный элемент встречается позже минимального, то выдать сообщение об этом.
11. Поменять местами первый минимальный и последний максимальный элементы.
12. Найти максимум из отрицательных элементов.
13. Найти минимум из положительных элементов.

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 10

Тема раздела: Операторы языка программирования

Тема практического занятия: Обработка двумерных массивов

Цель: освоить навыки работы с элементами двумерного массива

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Написать программу для двумерного массива

1. Найти минимальный элемент в каждом столбце и записать данные в новый массив.
2. Для каждой строки подсчитать количество положительных элементов и записать данные в новый массив.
3. Для каждого столбца подсчитать сумму отрицательных элементов и записать данные в новый массив.
4. Для каждого столбца подсчитать сумму четных положительных элементов и записать данные в новый массив.
5. Для каждой строки подсчитать количество элементов, больших заданного числа, и записать данные в новый массив.
6. Для каждого столбца найти первый положительный элемент и записать данные в новый массив.
7. Для каждой строки найти последний четный элемент и записать данные в новый массив.
8. Для каждого столбца найти номер последнего нечетного элемента и записать данные в новый массив.
9. Для каждой строки найти номер первого отрицательного элемента и записать данные в новый массив.
10. Для каждой строки найти сумму элементов с номерами от k_1 до k_2 и записать данные в новый массив.
11. Для каждого столбца найти произведение элементов с номерами от k_1 до k_2 и записать данные в новый массив.
12. Для каждой строки подсчитать сумму элементов, не попадающих в заданный интервал, и записать данные в новый массив.
13. Подсчитать сумму элементов каждой строки и записать данные в новый массив. Найти максимальный элемент нового массива.
14. Подсчитать произведение элементов каждого столбца и записать данные в новый массив. Найти минимальный элемент нового массива.

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 11

Тема раздела: Операторы языка программирования

Тема практического занятия: Сортировка массива метод «пузырька», вставками

Цель: изучить сортировку массива различными методами, написать программу для изученных методов сортировки

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Реализовать алгоритм сортировки методом «Пузырька»

Задание 2. Реализовать алгоритм сортировки вставками

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 12

Тема раздела: Операторы языка программирования

Тема практического занятия: Сортировка массива алгоритм Шелла, быстрая сортировка

Цель: изучить сортировку массива различными методами, написать программу для изученных методов сортировки

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;

- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

5. Запустить MS Visual Studio
6. Создать консольное приложение
7. Выполнить задание
8. Сохранить файл

Этапы выполнения работы:

Задание 1. Реализовать алгоритм сортировки Шелла

Задание 2. Реализовать алгоритм быстрой сортировки

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 13

Тема раздела: Операторы языка программирования

Тема практического занятия: Последовательный и двоичный поиск

Цель: изучить методы поиска элемента массива, написать программу

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Реализовать алгоритм последовательного поиска

Задание 2. Реализовать алгоритм двоичного поиска

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 14

Тема раздела: Процедуры и функции

Тема практического занятия: Программирование алгоритмов с использованием функций, не возвращающих значения

Цель: составить программу для функции, не возвращающей значение

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Разработать функцию, которая заполняет массив случайными числами

Задание 2. Разработать функцию, которая выводит массив на экран

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 15

Тема раздела: Процедуры и функции

Тема практического занятия: Программирование алгоритмов с использованием функций, возвращающих значения

Цель: составить программу для функции, возвращающих значение

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:**Задание 1.** Разработать функцию

1. $\min(a,b)$ для нахождения минимального из двух чисел. Вычислить минимальное значение из четырех чисел x, y, z, v .
2. $\max(a,b)$ для нахождения максимального из двух чисел. Вычислить значение выражения $z = \max(x, 2y - x) + \max(5x + 3y, y)$.
3. которая вычисляет значение по следующей формуле: $f(x) = x^3 - \sin x$. Определить, в какой из точек a или b , функция принимает наибольшее значение.
4. которая вычисляет значение по следующей формуле: $f(x) = \cos(2x) + \sin(x - 3)$. Определить, в какой из точек a или b , функция принимает наименьшее значение.
5. которая возвращает младшую цифру натурального числа x .
6. которая возвращает вторую справа цифру натурального числа x .

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2

несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 16

Тема раздела: Процедуры и функции

Тема практического занятия: Программирование рекурсивных алгоритмов

Цель: составить программу для рекурсивных алгоритмов

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Разработать рекурсивную функцию, возвращающую значение

1. для вычисления n -го члена следующей последовательности $b_1 = -10, b_2 = 2, b_{n+2} = |b_n| - 6b_{n+1}$.

2. для вычисления n -го члена следующей последовательности $b_1 = 5, b_{n+1} = \frac{b_n}{n^2 + n + 1}$.

3. для нахождения наибольшего общего делителя методом Евклида:

$$\text{НОД}(a, b) = \begin{cases} a, & \text{если } a = b; \\ \text{НОД}(a - b, b), & \text{если } a > b; \\ \text{НОД}(a, b - a), & \text{если } b > a. \end{cases}$$

4. для вычисления значения функции Аккермана для неотрицательных чисел n и m . Функция Аккермана определяется следующим образом:

$$A(n, m) = \begin{cases} m + 1, & \text{если } n = 0; \\ A(n - 1, 1), & \text{если } n \neq 0, m = 0; \\ A(n - 1, A(n, m - 1)), & \text{если } n > 0, m > 0. \end{cases}$$

5. для вычисления числа сочетаний $C(n, m)$ где $0 \leq m \leq n$, используя следующие свойства $C_n^0 = C_n^n = 1; C_n^m = C_{n-1}^m + C_{n-1}^{m-1}$ при $0 < m < n$.

Задание 2. Разработать рекурсивную функцию, не возвращающую значение

1. Даны первый член и разность арифметической прогрессии. Написать рекурсивный метод для нахождения n -го члена и суммы n первых членов прогрессии.
2. Даны первый член и знаменатель геометрической прогрессии. Написать рекурсивный метод для нахождения n -го члена и суммы n первых членов прогрессии.
3. Разработать рекурсивный метод, который по заданному натуральному числу N ($N \geq 1000$) выведет на экран все натуральные числа не больше N в порядке возрастания. Например, для $N=8$, на экран выводится 1 2 3 4 5 6 7 8.
4. Разработать рекурсивный метод, который по заданному натуральному числу N ($N \geq 1000$) выведет на экран все натуральные числа не больше N в порядке убывания. Например, для $N=8$, на экран выводится 8 7 6 5 4 3 2 1.
5. Разработать рекурсивный метод для вывода на экран цифр натурального числа в прямом порядке. Применить эту процедуру ко всем числам из интервала от A до B .
6. Разработать рекурсивный метод для перевода числа из десятичной системы счисления в двоичную.
7. Разработать рекурсивный метод для перевода числа из двоичной системы счисления в десятичную.
8. Разработать рекурсивный метод для вывода на экран всех делителей заданного натурального числа n .

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 18

Тема раздела: Символы и строки

Тема практического занятия: Работа с символами. Основные методы

Цель: освоить навыки работы с символами. Составить программу для обработки символов

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;

- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. В заданном массиве символов с помощью методов посчитать количество строчных, прописных букв, цифр и знаков пунктуации

Задание 2. Произвести замену символов

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 19

Тема раздела: Символы и строки

Тема практического занятия: Работа со строками. Строковый тип string

Цель: освоить навыки работы со строками. Составить программу для обработки строк

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Разработать программу, которая для заданной строки s:

1. подсчитывает общее число вхождений символов х и у;
2. определяет, какой из двух заданных символов встречается в строке чаще всего;
3. выводит на экран символы, которые наиболее часто встречается в строке;
4. выводит на экран символы, которые встречаются в строке только один раз;
5. определяет, имеются ли в строке два соседствующих одинаковых символа;
6. определяет, является ли строка палиндромом;
7. определяет, упорядочены ли по алфавиту символы строки;
8. подсчитывает количество букв в строке;
9. подсчитывает количество цифр в строке;
10. подсчитывает сумму всех содержащихся в строке цифр;
11. выводит на экран последовательность символов, расположенных до первого двоеточия;
12. выводит на экран последовательность символов, расположенных после последнего двоеточия;
13. выводит на экран последовательность символов, расположенных между круглыми скобками (считается, что в строке ровно одна пара круглых скобок);
14. находит самую длинную подстроку, состоящую только из цифр;
15. находит самую длинную подстроку, состоящую из повторяющегося символа

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 20

Тема раздела: Символы и строки

Тема практического занятия: Работа со строками. Строковый тип StringBuilder

Цель: освоить навыки работы со строками. Составить программу для обработки строк

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Разработать программу, которая

1. вставляет в строку символ x после каждого вхождения символа y;
2. вставляет в строку подстроку x после каждого вхождения подстроки y;
3. удваивает каждое вхождение заданного символа x;
4. удваивает каждое вхождение заданной подстроки x;
5. удаляет среднюю букву, если длина строки нечетная, и две средних, если длина строки четная;
6. удаляет все символы x;
7. удаляет из строки все цифры;
8. удаляет все подстроки substr;
9. заменяет все вхождения подстроки str1 на подстроку str2 (при этом str1 может являться частью str2);
10. заменяет все группы стоящих рядом точек на многоточие;
11. меняет местами первую букву со второй, третью с четвертой и т.д.;
12. меняет местами первую букву с последней, вторую с предпоследней и т.д.;
13. определяет, сколько различных символов встречается в строке;
14. удаляет из строки все подстроки, состоящие из цифр;
15. удаляет из строки самую длинную подстроку, состоящую из повторяющегося символа.

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 21

Тема раздела: Множество и записи

Тема практического занятия: Работа с данными типа множество

Цель: разработать программу для работы с данными типа множества

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Разработать программу, которая

1. Преобразует строку в множество
2. Для двух множеств написать функцию, которая их сравнивает

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 22

Тема раздела: Множество и записи

Тема практического занятия: Работа с данными типа запись

Цель: разработать программу для работы с данными типа запись

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;

- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

5. Запустить MS Visual Studio
6. Создать консольное приложение
7. Выполнить задание
8. Сохранить файл

Этапы выполнения работы:

Задание 1. Заполнить сведения о студентах и вывести их на экран

Задание 2. Вывести сведения на экран о студентах, год рождения которых после, введенного пользователем

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 23

Тема раздела: Организация файлового ввода-вывода

Тема практического занятия: Составление программ с использованием текстовых файлов. Байтовый поток

Цель: разработать программу для работы с байтовым потоком

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;

- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Выполнить

1. Создать файл и записать в него степени числа 3. Вывести на экран все компоненты файла с четным порядковым номером.
2. Создать файл и записать в него обратные натуральные числа $1, \frac{1}{2}, \dots, \frac{1}{n}$. Вывести на экран все компоненты файла с порядковым номером, кратным 3.
3. Создать файл и записать в него n первых членов последовательности Фибоначчи. Вывести на экран все компоненты файла с порядковым номером, не кратным 3.
4. Дана последовательность из n целых чисел. Создать файл и записать в него все четные числа последовательности. Вывести содержимое файла на экран.
5. Дана последовательность из n целых чисел. Создать файл и записать в него все отрицательные числа последовательности. Вывести содержимое файла

- на экран.
6. Дана последовательность из n целых чисел. Создать файл и записать в него числа последовательности, попадающие в заданный интервал. Вывести содержимое файла на экран.
 7. Дана последовательность из n целых чисел. Создать файл и записать в него числа последовательности, не кратные заданному числу. Вывести содержимое файла на экран.
 8. Дана последовательность из n вещественных чисел. Записать все эти числа в файл. Вывести на экран все компоненты, не попадающие в данный диапазон.
 9. Дана последовательность из n вещественных чисел. Записать все эти числа в файл. Вывести на экран все компоненты файла с нечетными номерами, большие заданного числа.
 10. Дана последовательность из n вещественных чисел. Записать все эти числа в файл. Вывести на экран все компоненты файла с четными номерами, меньшие заданного числа.
 11. Дана последовательность из n вещественных чисел. Записать все эти числа в файл. Вывести на экран все положительные компоненты файла.
 12. Дана последовательность из n вещественных чисел. Записать все эти числа в файл. Подсчитать среднее арифметическое компонентов файла, стоящих на четных позициях.
 13. Дана последовательность из n вещественных чисел. Записать все эти числа в файл. Найти максимальное значение среди компонентов файла, стоящих на нечетных позициях.
 14. Дано предложение. Создать файл и записать в него все символы данного предложения, отличные от знаков препинания. Вывести содержимое файла на экран.
 15. Дано предложение. Создать файл и записать в него все символы данного предложения, отличные от цифр. Вывести содержимое файла на экран.
 16. Создать файл, состоящий из слов. Вывести на экран все слова, которые начинаются на заданную букву.
 17. Создать файл, состоящий из слов. Вывести на экран все слова, длина которых равна заданному числу.
 18. Создать файл, состоящий из слов. Вывести на экран все слова, которые начинаются и заканчиваются одной буквой.
 19. Создать файл, состоящий из слов. Вывести на экран все слова, которые начинаются на ту же букву, что и последнее слово.

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на

половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 24

Тема раздела: Организация файлового ввода-вывода

Тема практического занятия: Составление программ с использованием текстовых файлов. Текстовый поток

Цель: разработать программу для работы с текстовым потоком

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio

2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Выполнить

1. Дан текстовый файл. Найти количество строк, которые начинаются и заканчиваются одной буквой.
2. Дан текстовый файл. Найти самую длинную строку и ее длину.
3. Дан текстовый файл. Найти самую короткую строку и ее длину.
4. Дан текстовый файл. Найти номер самой длинной строки.
5. Дан текстовый файл. Найти номер самой короткой строки.
6. Дан текстовый файл. Выяснить, имеется ли в нем строка, которая начинается с данной буквы. Если да, то напечатать ее.
7. Дан текстовый файл. Напечатать первый символ каждой строки.
8. Дан текстовый файл. Напечатать символы с k_1 по k_2 в каждой строке.
9. Дан текстовый файл. Напечатать все нечетные строки.
10. Дан текстовый файл. Напечатать все строки, в которых имеется хотя бы один пробел.
11. Дан текстовый файл. Напечатать все строки, длина которых равна данному числу.
12. Дан текстовый файл. Напечатать все строки, длина которых меньше заданного числа.
13. Дан текстовый файл. Напечатать все строки с номерами от k_1 до k_2 .
14. Дан текстовый файл. Получить слово, образованное k -ыми символами каждой строки.
15. Дан текстовый файл. Переписать в новый файл все его строки, вставив в конец каждой строки ее номер.
16. Дан текстовый файл. Переписать в новый файл все его строки, вставив в конец каждой строки количество символов в ней.
17. Дан текстовый файл. Переписать в новый файл все его строки, длина которых больше заданного числа.
18. Дан текстовый файл. Переписать в новый файл все его строки четной длины.
19. Дан текстовый файл. Переписать в новый файл все его строки, удалив из них символы, стоящие на четных местах.

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2

несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 25

Тема раздела: Указатели

Тема практического занятия: Использование указателей для организации связанных списков

Цель: разработать программу для работы со списками

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

1. Изучить методы для работы со списками разных классов

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 26

Тема раздела: Указатели

Тема практического занятия: Программирование алгоритмов с использованием стеков

Цель: разработать программу для работы со стеком

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;

- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Решить следующие задачи с использованием класса Stack:

1. Дан файл, в котором записан набор чисел. Переписать в другой файл все числа в обратном порядке.
2. Создать текстовый файл. Распечатать гласные буквы этого файла в обратном порядке.
3. Напечатать содержимое текстового файла t, выписывая литеры каждой его строки в обратном порядке.
4. Даны 2 строки s1 и s2. Из каждой можно читать по одному символу. Выяснить, является ли строка s2 обратной s1.
5. Написать программу подсчета выражения в префиксной форме.
6. Написать программу, которая преобразует выражение из инфиксной формы в префиксную.
7. Написать программу преобразования выражения из постфиксной формы в инфиксную.

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 27

Тема раздела: Указатели

Тема практического занятия: Программирование алгоритмов с использованием очередей

Цель: разработать программу для работы сочередью

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Решить следующие задачи с использованием класса Queue:

1. Дан текстовый файл. За один просмотр файла напечатать элементы файла в следующем порядке: сначала все символы, отличные от цифр, а затем все цифры, сохраняя исходный порядок в каждой группе символов.
2. Дан файл, содержащий числа. За один просмотр файла напечатать элементы файла в следующем порядке: сначала все числа, из интервала $[a,b]$, потом все числа, меньшие a , потом все числа, большие b , сохраняя исходный порядок в каждой группе чисел.
3. Дан текстовый файл. За один просмотр файла напечатать элементы файла в следующем порядке: сначала все слова, начинающиеся на гласную букву, потом все слова, начинающиеся на согласную букву, сохраняя исходный порядок в каждой группе слов.
4. Дан файл, содержащий числа. За один просмотр файла напечатать элементы файла в следующем порядке: сначала все положительные числа, потом все отрицательные числа, сохраняя исходный порядок в каждой группе чисел.
5. Дан текстовый файл. За один просмотр файла напечатать элементы файла в следующем порядке: сначала все слова, начинающиеся с прописной буквы, потом все слова, начинающиеся со строчной буквы, сохраняя исходный порядок в каждой группе слов.
6. Дан файл, содержащий информацию о сотрудниках фирмы: фамилия, имя, отчество, пол, возраст, размер зарплаты. За один просмотр файла напечатать элементы файла в следующем порядке: сначала все данные о мужчинах, потом все данные о женщинах, сохраняя исходный порядок в каждой группе сотрудников.
7. Дан файл, содержащий информацию о сотрудниках фирмы: фамилия, имя, отчество, пол, возраст, размер зарплаты. За один просмотр файла напечатать элементы файла в следующем порядке: сначала все данные о сотрудниках, зарплата которых меньше 10000, потом данные об остальных сотрудниках, сохраняя исходный порядок в каждой группе сотрудников.
8. Дан файл, содержащий информацию о сотрудниках фирмы: фамилия, имя, отчество, пол, возраст, размер зарплаты. За один просмотр файла напечатать элементы файла в следующем порядке: сначала все данные о сотрудниках младше 30 лет, потом данные об остальных сотрудниках, сохраняя исходный порядок в каждой группе сотрудников.
9. Дан файл, содержащий информацию о студентах: фамилия, имя, отчество, номер группы, оценки по трем предметам текущей сессии. За один просмотр файла напечатать элементы файла в следующем порядке: сначала все данные о студентах, успешно сдавших сессию, потом данные об остальных студентах, сохраняя исходный порядок в каждой группе сотрудников.
10. Дан файл, содержащий информацию о студентах: фамилия, имя, отчество,

номер группы, оценки по трем предметам текущей сессии. За один просмотр файла напечатать элементы файла в следующем порядке: сначала все данные о студентах, успешно обучающихся на 4 и 5, потом данные об остальных студентах, сохраняя исходный порядок в каждой группе сотрудников.

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 28-32

Тема раздела: Основные принципы объектно-ориентированного программирования (ООП)

Тема практического занятия: Создание класса. Создание экземпляров класса.

Методы класса. Конструкторы класса. Свойства класса

Цель: разработать класс со следующими компонентами: поля, методы, конструкторы, свойства

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;

- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Разработать класс

1. Создать класс Point, разработав следующие элементы класса:

- a. Поля:
 - int x, y;
- b. Конструкторы, позволяющие создать экземпляр класса:
 - с нулевыми координатами;
 - с заданными координатами.
- c. Методы, позволяющие:
 - вывести координаты точки на экран;
 - рассчитать расстояние от начала координат до точки;
 - переместить точку на плоскости на вектор (a, b).
- d. Свойства:
 - получить-установить координаты точки (доступное для чтений и записи);
 - позволяющие умножить координаты точки на скаляр (доступное только для записи).

2. Создать класс Triangle, разработав следующие элементы класса:

- a. Поля:
 - int a, b, c;
- b. Конструктор, позволяющий создать экземпляр класса с заданными длинами сторон.
- c. Методы, позволяющие:
 - вывести длины сторон треугольника на экран;

- рассчитать периметр треугольника;
- рассчитать площадь треугольника.

d. Свойства:

- позволяющее получить-установить длины сторон треугольника (доступное для чтения и записи);
- позволяющее установить, существует ли треугольник с данными длинами сторон (доступное только для чтения).

3. Создать класс Rectangle, разработав следующие элементы класса:

a. Поля:

- int a, b;

b. Конструктор, позволяющий создать экземпляр класса с заданными длинами сторон.

c. Методы, позволяющие:

- вывести длины сторон прямоугольника на экран;
- рассчитать периметр прямоугольника;
- рассчитать площадь прямоугольника.

d. Свойства:

- получить-установить длины сторон прямоугольника (доступное для чтения и записи);
- позволяющее установить, является ли данный прямоугольник квадратом (доступное только для чтения).

4. Создать класс Money, разработав следующие элементы класса:

a. Поля:

- int first;//номинал купюры
- int second; //количество купюр

b. Конструктор, позволяющий создать экземпляр класса с заданными значениям полей.

c. Методы, позволяющие:

- вывести номинал и количество купюр;
- определить, хватит ли денежных средств на покупку товара на сумму N рублей.
- определить, сколько шт товара стоимости n рублей можно купить на имеющиеся денежные средства.

d. Свойства:

- позволяющее получить-установить значение полей (доступное для чтения и записи);
- позволяющее рассчитать сумму денег (доступное только для чтения).

5. Создать класс для работы с одномерным массивом целых чисел. Разработать следующие элементы класса:

a. Поля:

- int [] IntArray;

- `int n`.
 - b. Конструктор, позволяющий создать массив размерности n .
 - c. Методы, позволяющие:
 - ввести элементы массива с клавиатуры;
 - вывести элементы массива на экран;
 - отсортировать элементы массива в порядке возрастания.
 - d. Свойства:
 - возвращающее размерность массива (доступное только для чтения);
 - позволяющее домножить все элементы массива на скаляр (доступное только для записи).
6. Создать класс для работы с двумерным массивом целых чисел. Разработать следующие элементы класса:
- a. Поля:
 - `int [,] IntArray`;
 - `int n`.
 - b. Конструктор, позволяющий создать массив размерности $n \times n$.
 - c. Методы, позволяющие:
 - ввести элементы массива с клавиатуры;
 - вывести элементы массива на экран;
 - вычислить сумму элементов i -того столбца.
 - d. Свойства:
 - позволяющее вычислить количество нулевых элементов в массиве (доступное только для чтения);
 - позволяющее установить значение всех элементов главной диагонали массива равное скаляру (доступное только для записи).
7. Создать класс для работы с двумерным массивом вещественных чисел. Разработать следующие элементы класса:
- a. Поля:
 - `double [][] DoubleArray`;
 - `int n, m`.
 - b. Конструктор, позволяющий создать массив размерности $n \times m$.
 - c. Методы, позволяющие:
 - ввести элементы массива с клавиатуры;
 - вывести элементы массива на экран;
 - отсортировать элементы каждой строки массива в порядке убывания.
 - d. Свойства:
 - возвращающее общее количество элементов в массиве (доступное только для чтения);
 - позволяющее увеличить значение всех элементов массива на скаляр (доступное только для записи).

8. Создать класс для работы со строками. Разработать следующие элементы класса:
- Поля:
 - StringBuilder line;
 - int n.
 - Конструктор, позволяющий создать строку из n символов.
 - Методы, позволяющие:
 - подсчитать количество пробелов в строке;
 - заменить в строке все прописные символы на строчные;
 - удалить из строки все знаки препинания.
 - Свойства:
 - возвращающее общее количество элементов в строке (доступное только для чтения);
 - позволяющее установить значение поля, в соответствии с введенным значением строки с клавиатуры, а также получить значение данного поля (доступно для чтения и записи)
9. Создать класс для работы с регулярными выражениями. Разработать следующие элементы класса:
- Поля:
 - Regex r;
 - string text;
 - Методы, позволяющие:
 - определить, содержит ли текст фрагменты, соответствующие шаблону поля;
 - вывести на экран все фрагменты текста, соответствующие шаблону поля;
 - удалить из текста все фрагменты, соответствующие шаблону поля;
 - Свойства:
 - позволяющее установить или получить строковое поле класса (доступно для чтения и записи)
 - позволяющее установить или получить регулярное выражение, хранящееся в соответствующем поле класса (доступно для чтения и записи)
10. Создать класс для работы с датой. Разработать следующие элементы класса:
- Поле DateTime data.
 - Конструкторы, позволяющие установить:
 - заданную дату
 - дату 1.01.2009
 - Методы, позволяющие:
 - вычислить дату предыдущего дня;
 - вычислить дату следующего дня;
 - определить сколько дней осталось до конца месяца.
 - Свойства:

- позволяющее установить или получить значение поле класса (доступно для чтения и записи)
- позволяющее определить год высокосным (доступно только для чтения)

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 33-36

Тема раздела: Основные принципы объектно-ориентированного программирования (ООП)

Тема практического занятия: Создание класса. Индексаторы класса. Унарные и бинарные операции класса

Цель: разработать класс со следующими компонентами: индексаторы, операции

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;

- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. В класс, разработанный ранее добавить

1. В класс Point добавить:

- е. Индексатор, позволяющий по индексу 0 обращаться к полю x, по индексу 1 – к полю y, при других значениях индекса выдается сообщение об ошибке.
- ф. Перегрузку:
 - операции ++ (--): одновременно увеличивает (уменьшает) значение полей x и y на 1;
 - констант true и false: обращение к экземпляру класса дает значение true, если значение полей x и y совпадает, иначе false;
 - операции бинарный +: одновременно добавляет к полям x и y значение скаляра;
 - преобразования типа Point в string (и наоборот).

2. В класс Triangle добавить:

- е. Индексатор, позволяющий по индексу 0 обращаться к полю a, по индексу 1 – к полю b, по индексу 2 – к полю c, при других значениях индекса выдается сообщение об ошибке.
- ф. Перегрузку:
 - операции ++ (--): одновременно увеличивает (уменьшает) значение полей a, b и c на 1;
 - констант true и false: обращение к экземпляру класса дает значение true, если треугольник с заданными длинами сторон существует, иначе false;
 - операции *: одновременно домножает поля a, b и c на скаляр;

- преобразования типа Triangle в string (и наоборот).

3. В класс Rectangle добавить:

- Индексатор, позволяющий по индексу 0 обращаться к полю a, по индексу 1 – к полю b, при других значениях индекса выдается сообщение об ошибке.
- Перегрузку:
 - операции ++ (--): одновременно увеличивает (уменьшает) значение полей a и b;
 - констант true и false: обращение к экземпляру класса дает значение true, если прямоугольник с заданными длинами сторон является квадратом, иначе false;
 - операции *: одновременно домножает поля a и b на скаляр;
 - преобразования типа Rectangle в string (и наоборот).

4. В класс Money добавить:

- Индексатор, позволяющий по индексу 0 обращаться к полю first, по индексу 1 – к полю second, при других значениях индекса выдается сообщение об ошибке.
- Перегрузку:
 - операции ++ (--): одновременно увеличивает (уменьшает) значение полей first и second;
 - операции !: возвращает значение true, если поле second не нулевое, иначе false;
 - операции бинарный +: добавляет к значению поля second значение скаляра;
 - преобразования типа Money в string (и наоборот).

5. Добавить в класс для работы с одномерным массивом целых чисел:

- Индексатор, позволяющий по индексу обращаться к соответствующему элементу массива.
- Перегрузку:
 - операции ++ (--): одновременно увеличивает (уменьшает) значение всех элементов массива на 1;
 - операции !: возвращает значение true, если элементы массива не упорядочены по возрастанию, иначе false;
 - операции бинарный *: домножить все элементы массива на скаляр;
 - преобразования класса массив в одномерный массив (и наоборот).

6. Добавить в класс для работы с двумерным массивом целых чисел:

- Двумерный индексатор, позволяющий обращаться к соответствующему элементу массива.
- Перегрузку:
 - операции ++ (--): одновременно увеличивает (уменьшает) значение всех элементов массива на 1;

- констант true и false: обращение к экземпляру класса дает значение true, если двумерный массив является квадратным;
- операции бинарный +: позволяющей сложить два массива соответствующих размерностей;
- преобразования класса массив в двумерный массив (и наоборот).

7. Добавить в класс для работы с двумерным массивом вещественных чисел:

- Двумерный индексатор, позволяющий обращаться к соответствующему элементу массива.
- Перегрузку:
 - операции ++ (--): одновременно увеличивает (уменьшает) значение всех элементов массива на 1;
 - констант true и false: обращение к экземпляру класса дает значение true, если каждая строка массива упорядоченна по возрастанию, иначе false.
 - операции *: позволяющей умножить два массива соответствующих размерностей.
 - преобразования класса массив в ступенчатый массив (и наоборот).

8. Добавить в класс для работы со строками:

- Индексатор, позволяющий по индексу обращаться к соответствующему символу строки.
- Перегрузку:
 - операции унарного + (-): преобразующей строку к строчным (прописным) символам;
 - констант true и false: обращение к экземпляру класса дает значение true, если строка не пустая, иначе false.
 - операции &: возвращает значение true, если строковые поля двух объектов посимвольно равны (без учета регистра), иначе false;
 - преобразования класса-строка в тип string (и наоборот).

9. Добавить в класс для работы с регулярными выражениями:

- Индексатор, позволяющий по индексу 0 обращаться к полю r, по индексу 1 – к полю text, при других значениях индекса выдается сообщение об ошибке.
- Перегрузку:
 - операции унарного -: удаляет из поля text все фрагменты, соответствующие регулярному выражению поля r.
 - констант true и false: обращение к экземпляру класса дает значение true, если поле text не пустое, иначе false;
 - операции бинарного +: дописывает в конец поля text строку.
 - преобразования класса Regex в тип string (и наоборот).

10. Добавить в класс для работы с датой:

- a. Индексатор, позволяющий определить дату *i*-того по счету дня относительно установленной даты (при отрицательных значениях индекса отсчет ведется в обратном порядке).
- b. Перегрузку:
 - операции `!`: возвращает значение `true`, если установленная дата не является последним днем месяца, иначе `false`;
 - констант `true` и `false`: обращение к экземпляру класса дает значение `true`, если установленная дата является началом года, иначе `false`;
 - операции `&`: возвращает значение `true`, если поля двух объектов равны, иначе `false`;
 - преобразования класса `DateTime` в тип `string` (и наоборот).

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 37

Тема раздела: Основные принципы объектно-ориентированного программирования (ООП)

Тема практического занятия: Структуры

Цель: разработать структуру по индивидуальному заданию

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать консольное приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Выполнить

1. На основе данных входного файла составить список студентов группы, включив следующие данные: ФИО, год рождения, домашний адрес, какую школу окончил. Вывести в новый файл информацию о студентах, окончивших заданную школу, отсортировав их по году рождения.
2. На основе данных входного файла составить список студентов группы, включив следующие данные: ФИО, номер группы, результаты сдачи трех экзаменов. Вывести в новый файл информацию о студентах, успешно сдавших сессию, отсортировав по номеру группы.
3. На основе данных входного файла составить багажную ведомость камеры хранения, включив следующие данные: ФИО пассажира, количество вещей, общий вес вещей. Вывести в новый файл информацию о тех пассажирах, средний вес багажа которых превышает заданный, отсортировав их по количеству вещей, сданных в камеру хранения.
4. На основе данных входного файла составить автомобильную ведомость, включив следующие данные: марка автомобиля, номер автомобиля, фамилия его владельца, год приобретения, пробег. Вывести в новый файл информацию об автомобилях, выпущенных ранее определенного года, отсортировав их по пробегу.

5. На основе данных входного файла составить список сотрудников учреждения, включив следующие данные: ФИО, год принятия на работу, должность, зарплата, рабочий стаж. Вывести в новый файл информацию о сотрудниках, имеющих зарплату ниже определенного уровня, отсортировав их по рабочему стажу.
6. На основе данных входного файла составить инвентарную ведомость склада, включив следующие данные: вид продукции, стоимость, сорт, количество. Вывести в новый файл информацию о той продукции, количество которой менее заданной величины, отсортировав ее по количеству продукции на складе.
7. На основе данных входного файла составить инвентарную ведомость игрушек, включив следующие данные: название игрушки, ее стоимость (в руб.), возрастные границы детей, для которых предназначена игрушка. Вывести в новый файл информацию о тех игрушках, которые предназначены для детей от N до M лет, отсортировав их по стоимости.
8. На основе данных входного файла составить список вкладчиков банка, включив следующие данные: ФИО, № счета, сумма, год открытия счета. Вывести в новый файл информацию о тех вкладчиках, которые открыли вклад в текущем году, отсортировав их по сумме вклада.
9. На основе данных входного файла составить список студентов, включающий фамилию, факультет, курс, группу, 5 оценок. Вывести в новый файл информацию о тех студентах, которые имеют хотя бы одну двойку, отсортировав их по курсу.
10. На основе данных входного файла составить список студентов, включающий ФИО, курс, группу, результат забега. Вывести в новый файл информацию о студентах, показавших три лучших результата в забеге. Если окажется, что некоторые студенты получили такие же высокие результаты, то добавить их к списку победителей.

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие №38

Тема раздела: Интегрированная среда разработчика

Тема практического занятия: Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.

Цель: овладеть навыками использования среды разработки для создания графических приложений, создать новый проект, изучить основные элементы управления

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать оконное (Windows) приложение
3. Выполнить задание

4. Сохранить файл

Этапы выполнения работы:

Задание 1. Создать оконное приложение

Задание 2. Добавить на форму компонент управления «Кнопка»

Задание 3. Добавить код на кнопку

```
private void button1_Click(object sender, EventArgs e)
{
    MessageBox.Show("Привет");
}
```

Задание 4. Запустите приложение

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 39

Тема раздела: Визуальное событийно-управляемое программирование

Тема практического занятия: Создание проекта с формой, настройка формы

Цель: Изучение методов построения форм Windows и получение навыков по настройке форм, созданию непрямых и наследуемых (производных) форм

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать оконное (Windows) приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Создать оконное приложение

Задание 2. В окне свойств задайте следующие значения

Свойство	Значение
Text	Trey Research
FormBorderStyle	Fixed3D
StartPosition	Manual
Location	100; 200
Opacity	75%

Задание 3. Поместите на форму 3 кнопки, Поочередно выберите каждую кнопку и в окне **Свойства** задайте свойству кнопок **Text** значения **Border Style**, **Resize** и **Opacity**. Для кнопки **Border Style** задайте свойство **Anchor - Top, Left**.

Задание 4. Задайте в окне кода на кнопки

```
this.FormBorderStyle = FormBorderStyle.Sizable;  
this.Size = new Size(300, 500);  
this.Opacity = 1;
```

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 41-52

Тема раздела: Визуальное событийно-управляемое программирование

Тема практического занятия: Создание проекта с использованием компонентов управления

Цель: Изучение способов использования элементов управления и получение навыков по обработке событий.

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;

- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать оконное (Windows) приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Создать оконное приложение

Задание 2. Поместить на форму изучаемые компоненты управления

Задание 3. Задать свойства компонента

Задание 2. Написать программный код к компоненту

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 незначительных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 53

Тема раздела: Разработка оконного приложения

Тема практического занятия: Разработка функциональной схемы работы приложения.

Цель: усвоить знание функциональной схемы приложения; освоить способы создания функциональных схем и их реализации.

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;

- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать оконное (Windows) приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Составить функциональную схемы работы приложения. Описать в ней

- Цель разработки приложение
- Конечные пользователи приложения
- Перечисление форм, их назначение
- Для каждого компонента форм приложения описать реакцию на те или иные действия пользователя

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 54-57

Тема раздела: Этапы разработки приложений

Цель: усвоить знание этапов разработки приложения, разработать приложение по индивидуальному заданию

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 90 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать оконное (Windows) приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Создать интерфейс пользователя

Задание 2. Создать процедуры обработки событий

Задание 3. Тестирование, отладка приложения

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Практическое занятие № 58

Тема раздела: Иерархия классов

Тема практического занятия: Создание класса

Планируемые результаты:

уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;

- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования;
- основные принципы объектно-ориентированного программирования на примере алгоритмического языка;
- понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Материально-техническое и комплексно-методическое обеспечение: Для проведения практической работы используется следующее обеспечение: персональный компьютер, подключённый к Интернету, MS Visual Studio

Время выполнения: 45 минут

Форма отчетности по занятию: файл с программным кодом

Последовательность выполнения работы

1. Запустить MS Visual Studio
2. Создать оконное (Windows) приложение
3. Выполнить задание
4. Сохранить файл

Этапы выполнения работы:

Задание 1. Создать класс по индивидуальному заданию

Критерии оценивания:

Оценка 5 «отлично» работа выполнена полностью и правильно, сделаны правильные выводы

Оценка 4 «хорошо» работа выполнена правильно с учетом 1-2 несущественных ошибок, исправленных самостоятельно по требованию преподавателя

Оценка 3 «удовлетворительно» работа выполнена правильно не менее чем на половину или допущены 3-4 существенные ошибки

Оценка 2 «неудовлетворительно» допущены 5 и более существенные ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя

Информационное обеспечение

Электронные издания (электронные ресурсы):

1. Кудрина, Е. В. Основы алгоритмизации и программирования на языке С#: учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2021. — 322 с. — (Профессиональное образование). — ISBN 978-5-534-10772-2. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/475228>
2. Казанский, А. А. Программирование на Visual С#: учебное пособие для среднего профессионального образования / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2021. — 192 с. — (Профессиональное образование). — ISBN 978-5-534-14130-6. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/471261>
3. Подбельский, В. В. Программирование. Базовый курс С#: учебник для среднего профессионального образования / В. В. Подбельский. — Москва: Издательство Юрайт, 2020. — 369 с. — (Профессиональное образование). — ISBN 978-5-534-11467-6. — Текст: электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/456697>
4. Трофимов, В. В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2021. — 137 с. — (Профессиональное образование). — ISBN 978-5-534-07321-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/473347>

Интернет-ресурсы:

1. Справочник по WPF - <http://msdn.microsoft.com/ru-ru/library/ms754130.aspx>
2. Справочник по базам данных - <http://msdn.microsoft.com/ru-ru/library/h0y4a0f6.aspx>
3. Пошаговые руководства по С# - <http://msdn.microsoft.com/ruru/library/1dbsh6t3.aspx>
4. Учебные руководства - <http://msdn.microsoft.com/ru-ru/library/dd492171.aspx>

Электронно-библиотечные системы:

1. «ЭБС IPRbooks», ООО «Ай Пи Эр Медиа»
2. ЭБС «Электронная библиотека технического вуза», ООО «Политехресурс»
3. ЭБС «Лань», ООО «Издательство Лань»
4. «ЭБС elibrary», ООО «РУНЭБ»
5. ЭБС «ЮРАЙТ»
6. ЭБС «Book.ru»

