

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Федорова Марина Владимировна
Должность: Директор филиала
Дата подписания: 29.09.2023 10:20:48
Уникальный программный ключ:
e766def0e2eb455f02135d659e45051ac23041da

Приложение №5
к ППССЗ по специальности 09.02.03
Программирование в компьютерных
системах

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
ОП.05 ОСНОВЫ ПРОГРАММИРОВАНИЯ
для специальности**

09.02.03 Программирование в компьютерных системах

Уровень подготовки - базовый

Год начала подготовки 2020

Фонд оценочных средств разработан в соответствии с Федеральным государственным образовательным стандартом по специальности СПО 09.02.03 Программирование в компьютерных системах, Рабочей программы учебной дисциплины ОП.05 Основы программирования и Положения о формировании фонда оценочных средств для проведения текущего контроля, промежуточной и государственной итоговой аттестации лиц, обучающихся по программам среднего профессионального образования в СамГУПС (приказ №196 от 18.03.2020г.)

Разработчики:

филиал СамГУПС в г.Алатыре преподаватель И.И.Скворцова
(место работы) (занимаемая должность) (инициалы, фамилия)

Рассмотрено на заседании предметной (цикловой) комиссией
математических и общих естественнонаучных дисциплин
Протокол № 6 25 мая от 2021г.

Председатель предметной (цикловой)комиссии математических
и общих естественнонаучных дисциплин Пас - /Р.В.Пасюнина/

СОГЛАСОВАНО:

Зам. директора по УР
Базилевич /Базилевич Т.Ю./
« 25 » мая 2021

Одобрено Методическим советом филиала
Протокол № 6 от « 25 » мая 2021г

СОДЕРЖАНИЕ

- 1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ**
- 2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ**
- 3. ОЦЕНКА ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ**
- 4. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ИТОГОВОЙ АТТЕСТАЦИИ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ**

ПРИЛОЖЕНИЕ

ЛИТЕРАТУРА

1. Паспорт комплекта контрольно-оценочных средств

Контрольно-оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины Основы программирования.

КОС включают контрольные материалы для проведения текущего контроля и промежуточной аттестации в форме экзамена и дифференцированного зачёта.

КОС разработаны на основании положений:

программы подготовки специалистов среднего звена по специальности СПО 09.02.03 Программирование в компьютерных системах;

программы учебной дисциплины Основы программирования;

учебного плана по специальности СПО 09.02.03 Программирование в компьютерных системах;

положения «О фонде оценочных средств для проведения текущего контроля успеваемости промежуточной и итоговой аттестации студентов и обучающихся АТЖТ – филиала СамГУПС».

2. Результаты освоения дисциплины, подлежащие проверке

2.1 Перечень умений, знаний, общих компетенций

В результате освоения учебной дисциплины Теория вероятностей и математическая статистика обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.03 Программирование в компьютерных системах (базовый уровень) следующими умениями, знаниями, которые формируют профессиональную компетенцию, и общими компетенциями:

У1. - работать в средах программирования;

У2. - разрабатывать алгоритмы;

У3. - реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;

У4. - описывать основные структуры данных;

У5. - реализовывать методы обработки данных.

З1. - этапы решения задачи на компьютере;

З2. - типы данных;

З3. - базовые конструкции изучаемых языков программирования;

З4. - принципы структурного и модульного программирования;

З5. - принципы объектно-ориентированного программирования;

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

- ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.
- ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.
- ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.
- ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.
- ОК 6. Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями.
- ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за результат выполнения заданий.
- ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.
- ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.
- ПК 1.1. Выполнять разработку спецификаций отдельных компонент.
- ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.
- ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств
- ПК 1.4. Выполнять тестирование программных модулей
- ПК 1.5. Осуществлять оптимизацию программного кода модуля
- ПК 3.1. Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.

2.2. Форма аттестации

Формой аттестации по учебной дисциплине является экзамен и дифференцированный зачет

2.3. Результаты освоения дисциплины, подлежащие проверке

- У1. – уметь работать в среде программирования;
- У2. – уметь реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.
- З1. – знать этапы решения задач на компьютере;
- З2. – знать типы данных;
- З3. – знать базовые конструкции изучаемых языков программирования;
- З4. – знать принципы структурного и модульного программирования;
- З5. – знать принципы объектно-ориентированного программирования.

Таблица 1.1

Результаты обучения: умения, знания и общие компетенции	Показатели оценки результата	Форма контроля и оценивания
Уметь:		
<p><i>У 1 - работать в средах программирования;</i> ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество. ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития. ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.</p>	<p>-работать в средах программирования Pascal ABC, Visual Studio.NET, NetBeans;</p>	<p>Устный опрос Самостоятельные работы; практические работы контрольные работы;</p>
<p><i>У 2 - разрабатывать алгоритмы;</i> ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество. ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.</p>	<p>-составлять блок-схемы алгоритмов; -разрабатывать линейные, разветвляющиеся и циклические алгоритмы.</p>	<p>Устный опрос Самостоятельные работы; практические работы контрольные работы;</p>
<p><i>У 3 -- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;</i> ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество. ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития. ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.</p>	<p>-писать и отлаживать программу на языках программирования Pascal, C++, Java</p>	<p>Устный опрос Самостоятельные работы; практические работы контрольные работы;</p>
<p><i>У 4 - описывать основные структуры данных;</i> ОК 2. Организовывать собственную деятельность, выбирать типовые</p>	<p>-объявлять данные простого и составного типа, а так же использовать эти данные в программе.</p>	<p>Устный опрос Самостоятельные работы; практические</p>

<p>методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.</p> <p>ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.</p> <p>ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.</p>		<p>работы контрольные работы;</p>
<p><i>У 5 - реализовывать методы обработки данных;</i></p> <p>ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.</p> <p>ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.</p> <p>ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.</p>	<p>-обработать одномерные и двумерные числовые массивы;</p> <p>-обработать символьные массивы;</p> <p>-обработать строки;</p> <p>-работать со структурами (записями);</p> <p>-работать с файлами.</p>	<p>Устный опрос Самостоятельные работы; практические работы контрольные работы;</p>
<p>Знать:</p>		
<p><i>З1. - этапы решения задачи на компьютере;</i></p> <p>ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.</p> <p>ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.</p> <p>ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.</p> <p>ОК 6. Работать в коллективе и в</p>	<p>-этапы решения задач на ПК.</p>	

<p>32 - типы данных; ОК 4. Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития. ОК 5. Использовать</p>	<p>-основные типы данных (простые, структурированные)</p>	
<p>33. - базовые конструкции изучаемых языков программирования; ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество. ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность. ОК 4. Осуществлять поиск и использование информации,</p>	<p>-базовые конструкции языков Pascal, C++, Java</p>	
<p>34. - принципы структурного и модульного программирования; ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество. ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность. ОК 4. Осуществлять поиск и</p>	<p>-основы структурного программирования; -основы модульного программирования.</p>	
<p>35. - принципы объектно-ориентированного программирования; ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество. ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность. ОК 4. Осуществлять поиск и использование информации,</p>	<p>-базовые понятия и основные принципы ООП.</p>	

3. Оценка освоения учебной дисциплины

3.1. Формы и методы оценивания

Предметом оценки служат умения и знания, предусмотренные ФГОС по дисциплине Основы программирования, направленные на формирование общих и профессиональных компетенций.

Оценка освоения дисциплины Основы программирования включает текущий контроль успеваемости, аттестацию в виде экзамена. Проведение текущего контроля успеваемости осуществляется в форме устных опросов, письменных заданий, практических занятий, контрольных работ. Для этих целей формируются фонды оценочных средств, включающие типовые задания, контрольные работы, тесты и методы контроля, позволяющие оценить знания, умения и уровень приобретенных компетенций.

Контроль и оценка освоения учебной дисциплины по темам (разделам)

Таблица 2.2

Элемент учебной дисциплины	Формы и методы контроля			
	Контроль в ходе изучения дисциплины		Промежуточная аттестация	
	Форма контроля	Проверяемые ОК, У, З	Форма контроля	Проверяемые ОК, У, З
Раздел 1 Введение в программирование			<i>Дифференцированный зачет</i>	<i>У2, У3, У5 З 1,32 ОК 4, ОК 5, ОК2</i>
Тема 1.1 Основы алгоритмизации	<i>Практическое занятие №1-2 Лабораторные работы № 1-3 Тестирование Самостоятельные работы</i>	<i>У2, З 1,33 ОК 4, ОК 5, ОК 2</i>		
Тема 1.2 Языки программирования	<i>Тестирование Самостоятельные работы</i>	<i>У1, У3 З 2,33 ОК 4, ОК 5, ОК 2</i>		
Тема 1.3 Типы данных	<i>Тестирование Самостоятельные работы</i>	<i>У1, У3 З 2,33 ОК 4, ОК 5, ОК 2</i>		
Раздел 2. Основные конструкции языков программирования			<i>Экзамен, Дифференцированный зачет</i>	<i>У2, У3, У4 З3 ОК 2, ОК4, ОК5, ОК3</i>
Тема. 2.1 Операторы языка	<i>Практическое занятие №3-5</i>	<i>У2 З3</i>		

программирования	<i>Лабораторные работы №4-6 Контрольная работа № 1 Самостоятельные работы</i>	<i>ОК 2, ОК4, ОК5, ОК3</i>		
Раздел 3. Структурное и модульное программирование			<i>Экзамен, Дифференцированный зачет</i>	<i>У2, У3, У4, У5 31-35 ОК 2, ОК4, ОК5, ОК3</i>
Тема 3.1. Процедуры и функции	<i>Практическое занятие №6-7 Лабораторные работы №7-9 Тестирование Самостоятельные работы</i>	<i>У1, У2, У3, У4, У5 31-33 ОК 4, ОК 5, ОК 2</i>		
Тема 3.2. Структуризация в программировании	<i>Тестирование Самостоятельные работы</i>	<i>31-35 ОК 2, ОК4, ОК5, ОК3</i>		
Тема 3.3. Модульное программирование	<i>Практическое занятие №8-9 Лабораторные работы №10-11 Контрольная работа № 2 Самостоятельные работы</i>	<i>У1-У5 31-35 ОК 2, ОК4, ОК5, ОК3</i>		
Раздел 4. Структуры данных			<i>Экзамен, Дифференцированный зачет</i>	<i>У2, У3, У4, У5 31-35 ОК 2, ОК4, ОК5, ОК3</i>
Тема 4.1. Массивы	<i>Практическое занятие №10-11 Лабораторные работы №12-13 Самостоятельные работы</i>	<i>У1-У5 31-35 ОК 4, ОК 5, ОК 2</i>		
Тема 4.2. Строки	<i>Практическое занятие №12 Лабораторные работы №14-15 Самостоятельные работы</i>	<i>У1-У5 31-35 ОК 4, ОК 5, ОК 2</i>		
Тема 4.3. Множества	<i>Практическое занятие №13 Лабораторная работа № 16 Самостоятельные работы</i>	<i>У2 33ОК 2, ОК4, ОК5, ОК3</i>		
Тема 4.4. Записи	<i>Практическое занятие №14 Самостоятельные работы</i>	<i>У2 33ОК 2, ОК4, ОК5, ОК3</i>		

Тема 4.5. Файлы	<i>Практическое занятие №15-17 Лабораторная работа №17-20 Самостоятельные работы</i>	У1-У5 31-35 ОК 4, ОК 5, ОК 2		
Тема 4.6. Указатели	<i>Практическое занятие №18-19 Лабораторная работа №21-22 Самостоятельные работы</i>	У1-У5 31-35 ОК 4, ОК 5, ОК 2		
Раздел 5. Объектно – ориентированное программирование			<i>Экзамен, Дифференцированный зачет</i>	У1-У5 31-35 ОК2, ОК3, ОК4, ОК5
Тема 5.1 Основные принципы объектно-ориентированного программирования (ООП)	<i>Самостоятельные работы</i>	У1-У5 31-35 ОК 4, ОК 5, ОК 2		
Тема 5.2 Интегрированная среда разработчика	<i>Практическое занятие №20-21 Лабораторная работа №23 Самостоятельные работы</i>	У1-У5 31-35 ОК 4, ОК 5, ОК 2		
Тема 5.3 Этапы разработки приложения	<i>Практическое занятие №22 Самостоятельные работы</i>	У1-У5 31-35 ОК 4, ОК 5, ОК 2		
Тема 5.4 Иерархия классов	<i>Практическое занятие №23 Лабораторная работа №24-26 Контрольная работа №3 Самостоятельные работы</i>	У1-У5 31-35 ОК 4, ОК 5, ОК 2		
Тема 5.5. Визуальное событийно-управляемое программирование	<i>Практическое занятие №24-25 Лабораторная работа №27-30 Самостоятельные работы</i>	У1-У5 31-35 ОК 4, ОК 5, ОК 2		
Тема 5.6. Разработка оконного приложения	<i>Практическое занятие №26 Лабораторная работа №31-32 Самостоятельные работы</i>	У1-У5 31-35 ОК 4, ОК 5, ОК 2		

3.2. Типовые задания для оценки освоения учебной дисциплины

Перечень вопросов и источников для подготовки к аттестации

1. Этапы решения задач с использованием ЭВМ.
2. Понятие алгоритма. Подходы к определению алгоритма. Свойства алгоритма. Способы записи алгоритма.
3. Понятие алгоритма. Понятие исполнителя. Система команд исполнителя.
4. Понятие величины. Типы величин. Присваивание величин. Совместимость по присваиванию.
5. Понятие о структурном программировании. Другие парадигмы программирования: сравнительная характеристика.
6. Языки программирования. Алгоритмические языки (алфавит, синтаксис, семантика). Способы описания синтаксиса (язык металингвистических формул, синтаксические диаграммы).
7. Система программирования PascalABC.
8. Структура программы, элементы языка (алфавит). Понятие типа данных.
9. Операции (арифметические, логические) на типах. Стандартные функции. Выражения.
10. Процедуры консольного ввода и вывода, управление вводом-выводом. Оператор присваивания. Совместимость по присваиванию.
11. Условный оператор. Оператор множественного ветвления (выбора).
12. Циклы в Pascal: с предусловием, с постусловием. Связь с другими циклами.
13. Циклы в Pascal: с параметром. Связь с другими циклами.
14. Структурированные типы данных. Линейные массивы. Примеры задач.
15. Структурированные типы данных. Двумерные массивы. Примеры задач.
16. Сортировка массивов. Метод выбора. Двоичный поиск в массиве.
17. Сортировка массивов. Метод обмена.
18. Сортировка массивов. Метод вставок.
19. Подпрограммы в Pascal. Основные способы передачи параметров в подпрограмму, их сравнение.
20. Подпрограммы в Pascal. Область видимости. Локальные и глобальные идентификаторы.
21. Процедуры. Организация и вызов. Примеры.
22. Функции. Организация и вызов. Примеры.
23. Простые типы данных в Pascal.
24. Структурированные типы данных. Строковый тип данных в Pascal: основные процедуры и функции, примеры.
25. Рекурсия. Механизм рекурсии. Примеры.
26. Сортировка массивов. Метод быстрой сортировки.

27. Множества в Pascal. Примеры.
28. Комбинированный тип данных (записи). Оператор присоединения. Записи с вариантами. Программирование типовых алгоритмов обработки записей.
29. Файловые типы в Pascal. Общие процедуры для работы с файлами. Компонентные (типизированные) файлы.
30. Текстовые файлы. Текст-ориентированные процедуры и функции. Типовые задачи.
31. Прямой и последовательный доступ к компонентам файла. Процедуры и функции, ориентированные на прямой доступ к компонентам файла.
32. Поиск в типизированных файлах. Сортировка файлов (на примере одного из методов).
33. Типизированные файлы. Файлы записей. Типовые алгоритмы обработки.
34. Модуль. Общая структура модуля. Компиляция и подключение модуля.
35. Объектно-ориентированное программирование. Изучение интегрированной среды разработчика

Комплект заданий для выполнения лабораторных работ

Лабораторная работа №1.

Тема: Составление блок-схем линейных алгоритмов.

Цель: изучить правила разработки алгоритмов программ.

Задание:

Задание 1. Записать в виде инструкции присваивания формулу вычисления площади прямоугольника.

Задание 2. Записать в виде инструкции присваивания формулу вычисления сопротивления электрической цепи по известным значениям напряжения и силы тока.

Лабораторная работа №2.

Тема: Составление блок-схем ветвящихся алгоритмов.

Цель: изучить правила разработки алгоритмов программ.

Задание:

Задание 1. Составить алгоритм для вычисления площади треугольника со сторонами A , B , C и определения того, является ли треугольник равнобедренным.

Задание 2. Составил алгоритм для определения того, выполнен ли план по продаже товаров за день, если известны: план продажи (в рублях), количество проданного товара и цена товара.

Лабораторная работа №3.

Тема: Составление блок-схем циклических алгоритмов.

Цель: изучить правила разработки алгоритмов программ.

Задание:

Задание 1. Написать алгоритм программы, которая выдает на экран таблицу умножения на 7.

Задание 2. Написать алгоритм табулирования функции $y = -x^2 - x + 8$ на отрезке $[-1; 2]$ с шагом 0.2

Задание 3. Написать алгоритм программы для вычисления суммы n членов последовательности.

Лабораторная работа №4.

Тема: Составление программ линейных алгоритмов.

Цель: изучить основные операторы и функции языка Паскаль.

Задание: (вариант 2).

Составить программу.

Задание 1. Присвоить целой переменной d вторую из дробной части цифру в записи положительного вещественного числа x (например, если $x = 32,345$, то $d = 4$)

Задание 2. Определить, сколько полных часов (h) и полных минут (m) прошло от начала суток до того момента (в первой половине дня), когда часовая стрелка повернулась на f градусов ($0 \leq f \leq 360$, f – вещественное число).

Задание 3. Пусть k – целое число от 1 до 365. Присвоить целой переменной n значение $1, 2, \dots, 7$ в зависимости от того, на какой день недели (понедельник, вторник, ..., субботу или воскресенье) приходится k -ый день невисокосного года, котором 1 января понедельник.

Задание 4. Составить диалоговую программу, которая спрашивает, сколько товара и по какой цене берет покупатель, сообщает величину сдачи.

Лабораторная работа №5.

Тема: Программирование алгоритмов с ветвлением.

Цель: научиться составлять программы алгоритмов ветвящихся структур.

Задание:

Задание 1. Написать программу вычисления значения функции, заданной формулами.

$$F = \begin{cases} 2|x|, & -1 < x < 0 \\ -x, & \text{если } x = -1 \\ 0 & \text{в остальных случаях} \end{cases}$$

Задание 2. Дано двузначное число. Определить, одинаковы ли его цифры.

Задание 3. Написать программу, которая запрашивает коэффициенты квадратного уравнения и выдает сообщение, есть ли у него действительные корни и почему.

Лабораторная работа №6.

Тема: Составление программ циклической структур.

Цель: научиться составлять программы алгоритмов циклических структур.

Задание:

Задание 1. Ввести количество посадочных мест, вывести соответствующий этому количеству вид транспорта «велосипед», «мотоцикл», «легковой автомобиль», «микроавтобус», «автобус».

Задание 2. Написать программу табулирования функции $y = x^2 - 3x - 4$ от -2 до 2 с шагом $0,5$.

Задание 3. Напечатать столбиком таблицу чисел от 1 до 81 и их квадратных корней (число и корень из него в одной строке).

Лабораторная работа №7.

Тема: Организация и использование процедур.

Цель: закрепить умение работать с процедурами – объявление процедур, создание процедур и вызов процедур из программ.

Задание:

Задание 1. Составить программу вычисления значений функции $F(x) = \cos(x)$ на отрезке $[\pi/3, 2\pi/3]$ в точках $x_i = A + iH$, где $H = (B - A)/M$, $M = 20$ – заданное целое число.

Лабораторная работа №8.

Тема: Организация и использование функции.

Цель: закрепить умение работать с функциями – объявление функций, создание функций и вызов функций из программ.

Задание:

Задание 1. Составить программу вычисления значения функции функции $F(x)=\cos(x)$ на отрезке $[\pi/3, 2\pi/3]$ в точках $x_i=A+iH$, где $H=(B-A)/M, M=20$ – заданное целое число.

Лабораторная работа №9.

Тема: Применение рекурсивных функций.

Цель: научиться составлять программы для вычисления рекурсивных функций с использованием подпрограмм.

Задание:

Задание 1. Написать программу, которая проверяет, является ли целое число, введенное пользователем, простым.

Задание 2. (Старинная задача). Сколько можно купить быков, коров и телят, если плата за быка 10 рублей, за корову – 5 рублей, за теленка – полтинник (0,5 рубля), на 100 рублей надо купить 100 голов скота.

Задание 3. Найти сумму ряда с точностью $E=10^{-3}$, общий член которого $a_n=(2n+1)/2^n$.

Лабораторная работа №10.

Тема: Программирование модуля.

Цель: изучить структуру модуль, научиться составлять и использовать простейшие модули

Задание:

Задание 1. Написать модуль, который вычисляет функцию $y=x^2+x*(a-b)+c$ на интервале $x \in [2:7]$ при $a=17; b=4; c=3$. Сохранит его.

Задание 2. Для $x \in [1:4]$ и, принимающем только целые значения, вычислить: $S=24(x^5-17)+x^4$; $V=53(x^5-17)+x^4$; $W=(24(x^5-17)+x^4)+(x^5-17)+x^4$

Лабораторная работа №11.

Тема: Использование библиотек подпрограмм.

Цель: научиться использовать и подключать различные библиотеки подпрограмм.

Задание:

Задание 1. Написать программу, которая рисует на экране смайлик желтого цвета.

Задание 2. Написать программу, которая рисует на экране флажок красного цвета.

Задание 3. Написать программу, которая вычеркивает на экране домик.

Лабораторная работа №12.

Тема: Обработка одномерных массивов.

Цель: Изучить методы обработки одномерных массивов.

Задание:

Написать, отладить и протестировать программу, обеспечивающую выполнение следующих действий:

- Ввод с клавиатуры одномерного массива $A[N]$ целых чисел, $1 \leq N \leq 20$.
- Вывод исходного, т.е. только что введенного, массива.
- Обработка массива в соответствии с заданием.
- Вывод параметров массива, которые требуется найти по заданию.

Задание 1. Подсчитать количество нечетных элементов массива.

Пример: массив 7 2 6 5 4 4 10 3 9

нечетные элементы 7 5 3 9

их количество 4

Задание 2. Удалить последний четный элемент массива.

Пример: массив 7 2 6 5 4 4 10 3 9

четные элементы 2 6 4 4 10

последний из четных 10

массив после удаления 7 2 6 5 4 4 3 9

Задание 3. Создать массив В из нечетных элементов массива А.

Массив А не изменять. (Другая формулировка этой же задачи: скопировать нечетные элементы из массива А в В).

Пример: массив А 7 2 6 5 4 4 10 3 9

нечетные элементы 7 5 3 9

массив В 7 5 3 9

массив А (после копирования) 7 2 6 5 4 4 10 3 9

Лабораторная работа №13.

Тема: Обработка двумерных массивов.

Цель: Изучить методы обработки двумерных массивов.

Задание:

Написать, отладить и протестировать программу, обеспечивающую выполнение следующих действий:

- Ввод массива целых чисел с клавиатуры $A[N][M]$, $1 \leq N \leq 10$, $1 \leq M \leq 10$.
- Вывод исходного массива.
- Обработка массива в соответствии с заданием.
- Вывод параметров массива, которые требуется найти по заданию.

Задание 1: В массиве все элементы, стоящие выше максимального элемента, заменить на минимальный элемент.

Задание 2: Обнулить элементы в тех столбцах, в которых встречается хотя бы два

одинаковых элемента.

Задание 3: В массиве все элементы, имеющие четное значение суммы индексов,

заменить на минимальный элемент массива.

Лабораторная работа №14.

Тема: Работа со строковыми переменными.

Цель: Ознакомиться с методами обработки строк.

Задание:

Разработать программу, обеспечивающую обработку массива строк.

Программа после запуска выводит на экран заставку (с описанием реализованных функций и ФИО автора) и просит ввести строки. После ввода строк производится их обработка. Результат обработки выводится на экран. Вводимые строки содержат 0, 1, 2 или больше слов, разделенных любым количеством разделителей – пробелов и знаков препинания.

Задание 1: Все гласные большие буквы заменить символом '\$'.

Задание 2: За исключением русских и латинских букв, все символы заменить символом '_' (подчеркивание).

Задание 3: Сделать большими все буквы в тех словах, в которых согласных букв меньше гласных.

Лабораторная работа №15.

Тема: Использование стандартных функций и процедур для работы со строками.

Цель: Изучить функции, обрабатывающие строки.

Задание: Разработать программы с использованием стандартных процедур и функций для работы со строками.

Задание 1: Заменить маленькие буквы большими в словах, содержащих две и более подряд идущих одинаковых букв.

Задание 2: Удалить в самом длинном слове массива все символы, отличные от заглавных латинских букв.

Лабораторная работа №16.

Тема: Работа с данными типа множество.

Цель: Изучить способы обработки множеств.

Задание: Разработать программу, обеспечивающую проверку знаний шестнадцатеричных цифр.

Лабораторная работа №17.

Тема: Работа с файлом последовательного доступа.

Цель: Изучить функции для работы с файлами.

Задание:

Задание 1: Разработать программу, которая создает на диске A: файл и записывает в него 5 целых чисел, введенных пользователем

Задание 2: Разработать программу, которая дописывает в файл a:\numbers.txt пять введенных пользователем целых чисел

Задание 3: Разработать программу, которая выводит на экран содержимое файла a:\numbers.txt

Лабораторная работа №18.

Тема: Работа с файлом произвольного доступа.

Цель: Изучить функции для работы с файлами.

Задание:

Задание 1: Разработать программу, которая вычисляет среднее арифметическое чисел, находящихся в файле

Задание 2: Разработать программу, которая выводит на экран содержимое файла, имя которого указано пользователем

Задание 3: Разработать программу, которая дописывает в файл a:\phone.txt фамилию, имя и номер телефона. Если файла на диске нет, то создает его

Лабораторная работа №19 (4 ч.)

Тема: Разработка программ с чтением и записью файлов разных типов.

Цель: Изучить функции для работы с файлами.

Задание: Разработать программу, обеспечивающую обработку файлов.

Программа после запуска выводит на экран заставку (с описанием реализованных функций и ФИО автора) и просит ввести файл. После ввода файла производится его обработка. Результат обработки выводится на экран и в файл.

Задание 1: Файл input.dat содержит несколько строк текста. Слова в тексте могут разделяться пробелами и знаками препинания. Переписать текст в файл output.dat, удалив, если необходимо, какое-то количество слов, так чтобы не осталось слов, имеющих одинаковое сочетание первых двух букв.

Задание 2: Файл input.dat содержит несколько строк текста. Слова в тексте могут разделяться пробелами и знаками препинания. Переписать текст в файл output.dat, удалив из него все слова, начинающиеся с той же буквы, что и последнее слово последней строки.

Лабораторная работа №20 (4 ч.)

Тема: Использование указателей для организации связанных списков.

Цель: Ознакомиться с использованием указателей для организации связанных списков.

Задание: Разработать две программы, выполняющие следующие действия:

1. Ввод размера массива (или двух – в зависимости от задания)
2. Выделения памяти под исходный массив (массивы)
3. Ввод исходного массива (массивов)
4. Вывод введенных массивов
5. Обработка массива (массивов) в соответствии с вариантом
6. Вывод получившихся массивов
7. Освобождение динамической памяти

Ввести массив А. В массив В скопировать все элементы массива А, имеющие

четный индекс и четное значение. Массив В отсортировать по убыванию, используя модифицированный метод простого выбора.

Лабораторная работа №21.

Тема: Изучение интегрированной среды разработчика.

Цель: Ознакомиться с принципами работы в среде Visual Studio 2017.

Задание №1. Изучение интерфейса среды программирования Visual Studio 2017. Изучение свойств формы.

Задание 1. Создание приложения

Запускаем Microsoft Visual C++ .

После запуска системы мы увидим начальный пользовательский интерфейс, показанный на рис. 1.1.

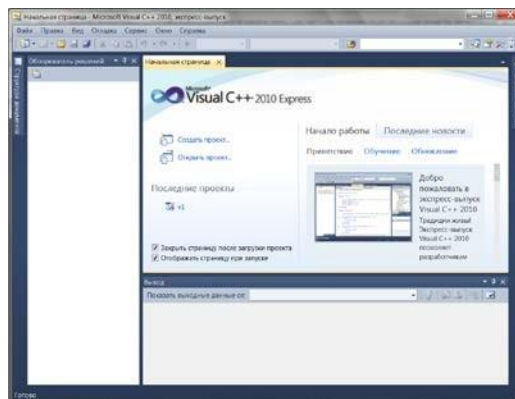


Рис. 1.1. Фрагмент стартовой страницы системы Visual Studio

Для создания приложения, необходимо в пункте меню File выполнить команду New Project (Новый проект). В появившемся окне New Project в левой колонке находится список установленных шаблонов (Installed Templates). Среди них — шаблоны языков программирования, встроенных в Visual Studio, в том числе Visual Basic, Visual C#, Visual C++, Visual F# и др. Нам нужен язык Visual C++. В узле Visual C++ области типов проектов выберем среду CLR, а затем в области шаблонов (в средней колонке) выберем шаблон (Templates) Windows Forms Application Visual C++.

Теперь введем имя проекта (Name) v4 и щелкнем на кнопке ОК, в результате увидим окно, представленное на рис. 1.2.

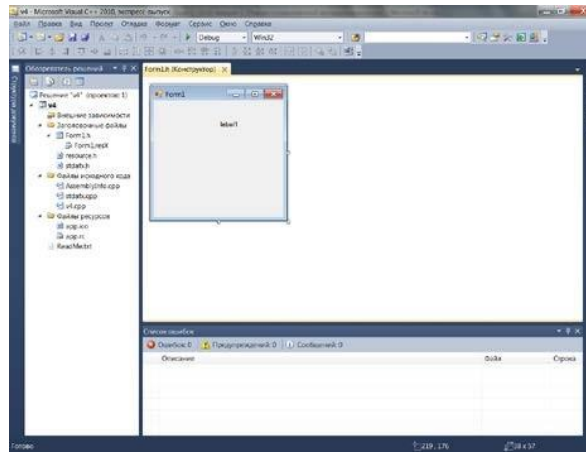


Рис. 1.2. Окно для проектирования пользовательского интерфейса

В этом окне изображена экранная форма — Form1. Первая программа будет отображать такую экранную форму, в которой будет что-либо написано, например «Microsoft Visual C++ 2010», также в форме будет расположена командная кнопка с надписью

«Нажми меня». При нажатии кнопки будет появляться диалоговое окно с сообщением «Всем привет!» В программе четыре объекта: форму Form, надпись на форме Label, кнопка Button и диалоговое окно MessageBox с текстом «Всем привет!» (окно с приветом).

Добавить в форму названные элементы управления. Для этого понадобится панель элементов управления Toolbox (Панель управления), ее можно добавить, например, с помощью комбинации клавиш Ctrl+Alt+x или ViewToolbox. Итак, добавьте метку Label и кнопку Button в форму, дважды щелкая на этих элементах на панели Toolbox. А затем следует расположить их примерно так, как показано на рис. 1.3.

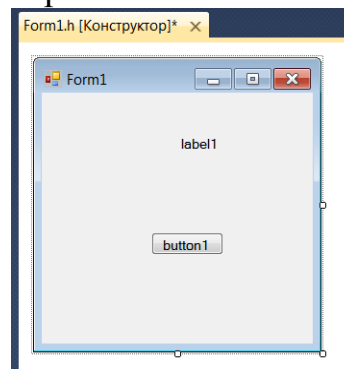


Рис. 1.3. Форма первого проекта

Каждый объект имеет свойства (properties- Свойства) . Свойств много, их можно увидеть, если щелкнуть правой кнопкой мыши в пределах формы и выбрать в контекстном меню команду Properties-Свойства, при этом появится панель свойств .

Для объекта label1 выбрать свойство Text и написать напротив этого поля «Microsoft Visual C++ 2010» (вместо текста label1). Для объекта button1 также в свойстве Text написать «Нажми меня».

Объекты не только имеют свойства, но и обрабатываются событиями. В задаче событием, которым управляем, является щелчок на командной кнопке. Для получения пустого обработчика этого события следует в свойствах кнопки button1 щелкнуть на значке молнии Events (события) и в списке всех возможных событий кнопки button1 выбрать двойным щелчком событие Click. После этого попадаем на вкладку программного кода Form1.h (см. рис. 1.4).

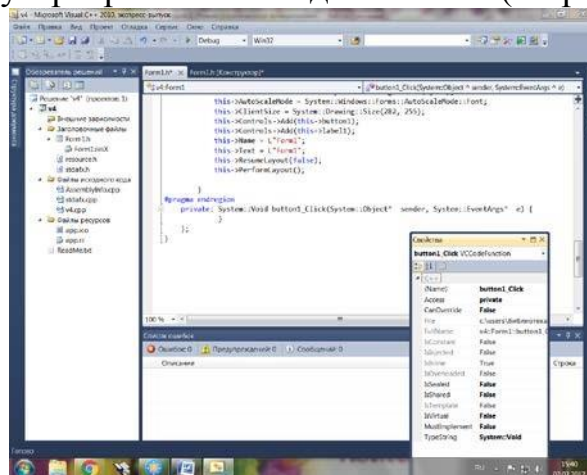


Рис 1.4 Вкладка программного кода

На вкладке Form1.h видно, что управляющая среда Visual C++ сгенерировала довольно таки много строк программного кода. В этом тексте уже можно найти те присваивания, которые сделали в панели свойств Properties. Например, для свойства Text кнопки Button управляющая среда назначила строку «Нажми меня»:

this->button1->Text = L"Нажми меня";

Пустой обработчик события button1_Click:

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) { }
```

Здесь в фигурных скобках пишутся команды, подлежащие выполнению после щелчка на кнопке. В фигурных скобках обработчика события напишите:

```
MessageBox::Show("Всем привет!");
```

Теперь нажмите клавишу F5 и проверьте работоспособность программы (рис. 1.5).

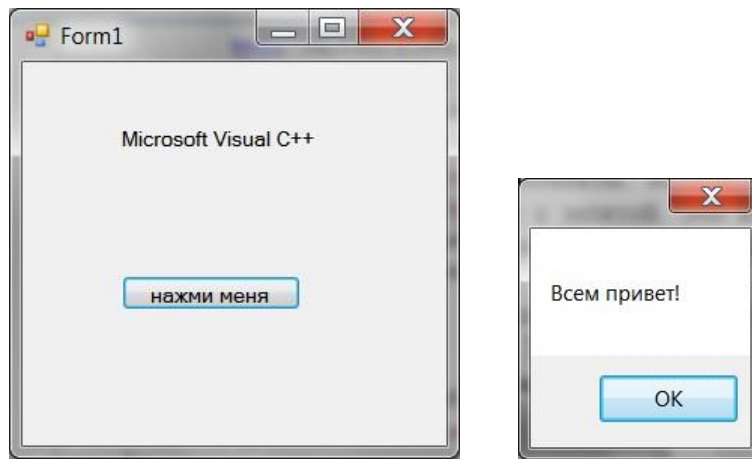


Рис. 1.5. Фрагмент работы программы

Задание 2. Обработка события MouseHover мыши

Событие MouseHover наступает тогда, когда пользователь указателем мыши «зависает» над каким-либо объектом, событие MouseHover происходит, когда указатель мыши наведен на элемент.

Таким образом, программа в данном примере должна содержать на экранной форме текстовую метку Label и кнопку Button. Метка должна отображать текст «Microsoft Visual C++ 2010»; при щелчке на командной кнопке, на которой попрежнему будет написано «Нажми меня», появится диалоговое окно с сообщением «Всем привет!». Кроме того, когда указатель мыши наведен на текстовую метку (то самое событие MouseHover), должно появиться диалоговое окно с текстом «Событие Hover».

Для решения этой задачи запустим Visual Studio 2010, щелкнем на пункте меню New Project. В появившемся окне New Project в левой колонке в узле Visual C++ выберем среду CLR, а затем в области шаблоны (в средней колонке) выберем шаблон (Templates) Windows Forms Application Visual C++. В качестве имени проекта введем имя Hover и щелкнем на кнопке ОК.

В дизайнера формы из панели Toolbox перетащим на форму метку Label и кнопку Button, а затем немного уменьшим размеры формы на свое усмотрение. Теперь добавим три обработчика событий в программный код. Для этого в панели Properties следует щелкнуть на значке молнии (Events) и двойным щелчком последовательно выбрать событие загрузки формы Form_Load, событие «щелчок на кнопке button1_Click» и событие label1_MouseHover.

При этом осуществится переход на вкладку программного кода Form1.h, и среда Visual Studio 2010 сгенерирует три пустых обработчика события (рис.1.6). Например, обработчик последнего события будет иметь вид:

```
private: System::Void label1_MouseHover(System::Object^ sender,
System::EventArgs^ e) {}
```

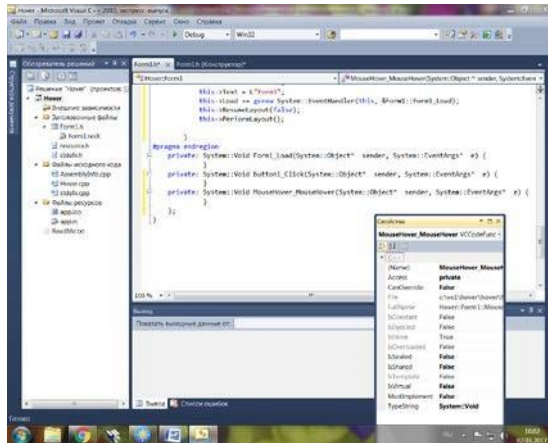
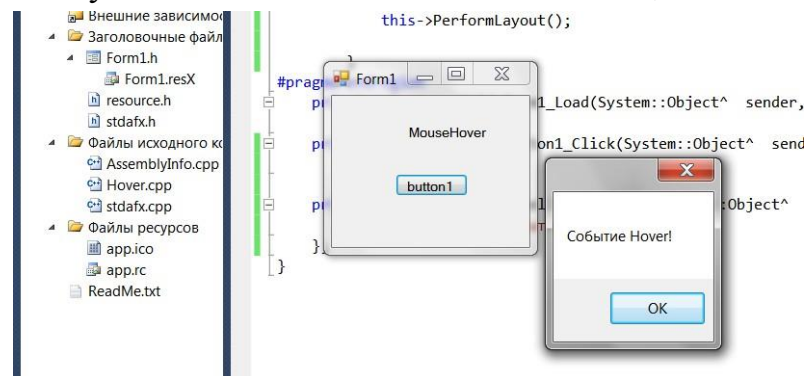


Рис. 1.6 Вкладка программного кода

Между фигурными скобками вставим вызов диалогового окна: `MessageBox::Show("Событие Hover!");`

Теперь проверим возможности программы: нажимаем клавишу F5, «зависаем» указателем мыши над `label1`, щелкаем на



кнопке `button1`. Все работает! (рис.1.7)

Рис.1.7 Работа приложения

Листинг. Фрагмент файла `Form1.h`, содержащего программный код тремя обработчиками событий

```
// .....
// Программный код, расположенный выше, создан средой
VisualStudio
// автоматически, поэтому автором не
приводится this->ResumeLayout(false);
this->PerformLayout();
}
#pragma endregion
// Данная программа управляется тремя событиями. Событие
загрузки формы
// Form1_Load инициализирует надписи заголовка формы,
текстовойметки
```



```

// и кнопки. Событие щелчок на кнопке button1_Click
вызывает появление
// диалогового окна с текстом "Всем привет!". Событие,
когда указатель
// мыши наведен на метку, вызывает появление диалогового окна
с текстом
// "Событие Hover".
private: System::Void Form1_Load(System::Object^
sender, System::EventArgs^ e)
{ // Обработка события загрузки формы:
this->Text = "Приветствие";
// или Form1::Text = "Приветствие";
label1->Text = "Microsoft Visual C++
2010"; button1->Text = "Нажми меня";
}
private: System::Void button1_Click(System::Object^
sender, System::EventArgs^ e)
{ // Обработка события щелчок на
кнопке: MessageBox::Show("Всем
привет!");
}
private: System::Void label1_MouseHover(System::Object^
sender, System::EventArgs^ e)
{ // Обработка события, когда указатель мыши наведен на
метку: MessageBox::Show("Событие Hover!");
}
};
}

```

Контрольные вопросы

1. Из каких двух этапов состоит процесс проектирования программы Visual C++.
2. Что такое программа, основанная на диалоге.
3. Что такое Windows Forms Application.

Лабораторная работа №22

Тема: Объявление класса, создание экземпляра класса.

Цель: научиться объявлять класс, описывать его поля и методы, создавать экземпляр класса с помощью конструктора.

Задание 1: Необходимо написать программу для учета успеваемости студентов

Группу студентов можно представить, как класс языка C++. Классы в программировании состоят из свойств и методов. Свойства — это любые данные-характеристики объекта класса. В нашем примере студент — объект класса, а его свойства — имя, фамилия, оценки и средний балл. Методы — это функции, которые могут выполнять какие-либо действия над данными

(свойствами) класса.

Модификаторы доступа **public** и **private**

Все свойства и методы классов имеют права доступа. По умолчанию в созданном классе все его содержимое является доступным для чтения и записи только для него самого. Модификатор доступа **public** необходим для того, чтобы разрешить доступ к данным класса извне. Все функции и переменные, которые находятся после модификатора **public**, становятся доступными из всех областей программы.

Недоступные (приватные) для общего пользования данные класса размещаются после модификатора доступа **private**. Если отсутствует модификатор **public**, то все функции и переменные, по умолчанию являются закрытыми.

Обычно, закрытыми делают все свойства класса, а публичными — его методы. Все действия с закрытыми свойствами класса реализуются через его методы.

Выполнение:

1. Создайте заголовочный файл *students.h*, в котором будет находиться класс *Students*: меню *Проект* → *Заголовочный файл (.h)*.

Описание будущего кода:

В нашем случае, объектом класса является студент, а его свойствами — имя, фамилия, оценки и средний балл.

У каждого студента есть имя — *name* и фамилия - *last_name*. Также, у него есть промежуточные оценки за весь семестр. Эти оценки мы будем записывать в целочисленный массив из пяти элементов. После того, как все пять оценок будут проставлены, определим средний балл успеваемости студента за весь семестр — свойство *sred_ball*.

Добавим в наш класс функцию *calculate_sred_ball()*, которая будет определять средний балл успеваемости ученика.

2. Добавьте в созданный файл код:

```
/* students.h */  
  
#include  
using namespace std;  
  
class Students{  
public:  
// Установка имени студента
```

```
void set_name(string student_name)
{
    name = student_name;
}

// Получение имени студента
string get_name()
{
    return name;
}

// Установка фамилии студента
void set_last_name(string
student_last_name)
{
    last_name = student_last_name;
}

// Получение фамилии студента
string get_last_name()
{
    return last_name;
}

// Установка промежуточных оценок
void set_scores(int student_scores[])
{
    for (int i = 0; i < 5; i++){
        scores[i] = student_scores[i];
    }
}

// Установка среднего балла
void set_sred_ball(float ball)
{
    sred_ball = ball;
}

// Получение среднего балла
float get_sred_ball()
{
    return sred_ball;
}

private:
```

```

// Промежуточные оценки
int scores[5];
// Средний балл
float sred_ball;
// Имя
string name;
// Фамилия
string last_name;
};

```

Таким образом, нельзя напрямую обращаться к закрытым данным класса. Осуществлять работу с этими данными можно только посредством методов (функций) этого класса. В приведенном выше примере используется функция **get_sred_ball()** для получения средней оценки студента, и **set_sred_ball()** для выставления этой оценки.

Функция **set_sred_ball()** воспринимает средний балл в качестве параметра и присваивает его значение закрытой переменной **sred_ball**.

Функция **get_sred_ball()** просто возвращает значение этой переменной.

3. Теперь в главный файл проекта добавьте следующее содержимое:

```

/* main.cpp */
#include "stdafx.h"
#include "students.h"
using namespace std;

bool score_is_valid (int test_score)
{
if (test_score <= 5 && test_score > 0)           // оценка от
1 до 5
    return true;
else
    return false;
}

int main()
{
SetConsoleCP(1251); // использование кириллицы
SetConsoleOutputCP(1251); // использование кириллицы

// Создание объекта класса Student
Students student;

```

```

string name;
string last_name;

// Ввод имени с клавиатуры
cout << "Name: ";
getline (cin, name);

// Ввод фамилии
cout << "Last name: ";
getline (cin, last_name);

// Сохранение имени и фамилии в объект класса Students
student.set_name(name);
student.set_last_name(last_name);
int scores[5]; // Оценки
int sum = 0; // Сумма всех оценок

// Ввод промежуточных оценок
for (int i = 0; i < 5; i++){
    cout << "Grade " << i+1 << ": "; do { cin >> scores[i];
    if (score_is_valid(scores[i])) {
        sum += scores[i];
        break;
    }
    else {
        cout << "Make correct input. Grade " << i+1 << ": ";
    }
} while (true);
}

// Сохраняем промежуточные оценки в объект класса Student
student.set_scores(scores);

// Считаем средний балл
float sred_ball = sum / 5.0;

// Сохраняем средний балл в объект класса Students
student.set_sred_ball(sred_ball);

// Выводим данные по студенту
cout << "Average grade of student " << student.get_name()
<< " "
<< student.get_last_name() << " equal: "
<< student.get_sred_ball() << endl;
system("pause");

```

```
return 0;
}
```

В начале программы определяется объект класса Students. Нужно понимать, что сам класс является только описанием его объекта. Так, класс Students является описанием любого из студентов, у которого есть имя, фамилия и возможность получения оценок.

Теперь об объекте созданного класса. Объект класса Students характеризует уже конкретного студента. Если стоит задача выставить оценки всем ученикам в группе, то необходимо создавать новый объект для каждого из них. Использование классов очень хорошо подходит для описания объектов реального мира.

После того, как объект student создан, вводится с клавиатуры фамилия, имя и промежуточные оценки для конкретного ученика. Пусть это будет Иван Иванов, у которого есть пять оценок за семестр — две тройки, две четверки и одна пятерка.

Введенные данные передаются **set**-функциям, которые присваивают их закрытым переменным класса. После введения промежуточных оценок высчитывается средний балл на основе этих оценок, а затем сохраняется это значение в закрытом свойстве sred_ball, с помощью функции set_sred_ball().

Задание 2 Создать класс Книга

с полями: название книги, количество страниц, год издания;

методы: вычислить сколько лет книге и количество дней прошедших после года издания книги.

Указания к выполнению:

Создать проект Книга, классическое пустое консольное приложение.

В проекте создать три файла: **book.h**, **book.cpp** и **main.cpp** со следующими кодами

book.h

```
typedef unsigned int uint;

class Book
{
public:

    Book(char*, uint, uint);

    ~Book();
```

```

class Error {};

uint book_years();
uint book_days();

char* get_name() const {return name;}
uint get_pages() const {return pages;}
uint get_year() const {return year;}

void set_name(char*);
void set_pages(uint);
void set_year(uint);

private:

static const uint year_min = 1900;
static const uint year_max = 2016;

char* name;
uint pages;
uint year;
bool is_year_valid();

};

```

book.cpp

```

Book::Book(char * n, uint p, uint y)
    :name{n}, pages{p}, year{y}
{
    if(!is_year_valid()) throw Error{};
}

Book::~Book()
{
}

uint Book::book_years()
{
    return year_max - year;
}

uint Book::book_days()

```

```

{
    return book_years()*365;
}

void Book::set_name(char* n)
{
    name = n;
}

void Book::set_pages(uint p)
{
    pages = p;
}

void Book::set_year(uint y)
{
    year = y;
}

bool Book::is_year_valid()
{
    if(year < year_min || year > year_max )
    {
        return false;
    }
    else
    {
        return true;
    }
}

```

main.cpp

```

int main()
{

    Book bk{"First book", 450, 1983};

    Book bk1{"Last book", 317, 2000};

    std::cout << bk.get_name() << " " << bk.book_years()
<< " " << bk.book_days() << std::endl;

    std::cout << bk1.get_name() << " " <<
bk1.book_years() << " " << bk1.book_days();
}

```



```
    return 0;
}
```

Соберите и откомпилируйте проект. Запустите на выполнение. Сделайте скриншоты с результатами работы программы.

Лабораторная работа №23.

Тема: Создание наследованного класса.

Цель: научиться создавать наследованные классы и экземпляры наследованных классов.

Задание 1. Создать базовый класс `human`, который будет описывать модель человека. В нем будут храниться имя, фамилия и отчество.

Задание 2. Создать два наследованных класса `student`, который хранит данные о студенте (список оценок и средний балл), и `teacher`, который будет хранить данные о преподавателе (количество учебных часов).

Выполнение

Создание базового класса

Для решения этой задачи создадим базовый класс `human`, который будет описывать модель человека. В нем будут храниться имя, фамилия и отчество.

Создайте файл `human.h`:

```
// human.h
#ifndef HUMAN_H_INCLUDED
#define HUMAN_H_INCLUDED

#include <string>
#include <sstream>

class human {
public:
    // Конструктор класса human
    human(std::string last_name, std::string name, std::string second_name)
    {
        this->last_name = last_name;
        this->name = name;
        this->second_name = second_name;
    }
};
```

```

// Получение ФИО человека
std::string get_full_name()
{
    std::ostringstream full_name;
    full_name << this->last_name << " "
        << this->name << " "
        << this->second_name;
    return full_name.str();
}

```

```

private:

```

```

    std::string name; // имя
    std::string last_name; // фамилия
    std::string second_name; // отчество
};

```

```

#endif // HUMAN_H_INCLUDED

```

Наследование от базового класса

Теперь создайте новый класс `student`, который будет наследником класса `human`. Поместите его в файл `student.h`.

```

// student.h

```

```

#ifndef STUDENT_H_INCLUDED
#define STUDENT_H_INCLUDED

```

```

#include "human.h"

```

```

#include <string>

```

```

#include <vector>

```

```

class student : public human {

```

```

    public:

```

```

        // Конструктор класса Student

```

```

        student(
            std::string last_name,
            std::string name,
            std::string second_name,
            std::vector<int> scores

```

```

        ) : human(
            last_name,
            name,
            second_name

```

```

        ) {
            this->scores = scores;

```

```
}
```

```
// Получение среднего балла студента
```

```
float get_average_score()
```

```
{
```

```
// Общее количество оценок
```

```
unsigned int count_scores = this->scores.size();
```

```
// Сумма всех оценок студента
```

```
unsigned int sum_scores = 0;
```

```
// Средний балл
```

```
float average_score;
```

```
for (unsigned int i = 0; i < count_scores; ++i) {
```

```
    sum_scores += this->scores[i];
```

```
}
```

```
average_score = (float) sum_scores / (float) count_scores;
```

```
return average_score;
```

```
}
```

```
private:
```

```
// Оценки студента
```

```
std::vector<int> scores;
```

```
};
```

```
#endif // STUDENT_H_INCLUDED
```

Функция `get_average_score` вычисляет среднее арифметическое всех оценок студента. Все [публичные свойства и методы](#) класса `human` будут доступны в классе `student`.

Конструктор базового класса

Для того, чтобы инициализировать [конструктор](#) родительского класса (в нашем случае — это сохранение имени, фамилии и отчества ученика), используется следующий синтаксис:

```
// Конструктор класса Student
```

```
student(
```

```
// аргументы конструктора текущего класса
```

```
): human(
```

```
// инициализация конструктора родительского класса
```

```
) {
```

```
// инициализация конструктора текущего класса
```

```
}
```

В конструктор класса `human` мы передаем инициалы человека, которые сохраняются в экземпляре класса. Для класса `students`, нам необходимо задать еще и список оценок студента. Поэтому конструктор `students` принимает все аргументы конструктора базового класса, а также дополнительные аргументы для расширения функционала:

// Конструктор класса Student

```
student(
    std::string last_name,
    std::string name,
    std::string second_name,
    std::vector<int> scores
) : human(
    last_name,
    name,
    second_name
) {
    this->scores = scores;
}
```

Список оценок студента хранится в [векторе](#).

Создание объекта класса `student`

Реализуем пользовательский интерфейс для работы с классом `student`.

// main.cpp

```
#include <iostream>
```

```
#include <vector>
```

```
#include "human.h"
```

```
#include "student.h"
```

```
int main(int argc, char* argv[])
```

```
{
```

```
    // Оценки студента
```

```
    std::vector<int> scores;
```

```
    // Добавление оценок студента в вектор
```

```
    scores.push_back(5);
```

```
    scores.push_back(3);
```

```
    scores.push_back(2);
```

```
scores.push_back(2);
scores.push_back(5);
scores.push_back(3);
scores.push_back(3);
scores.push_back(3);
scores.push_back(3);
```

```
// Создание объекта класса student
```

```
student *stud = new student("Петров", "Иван", "Алексеевич", scores);
```

```
// Вывод полного имени студента (используется унаследованный метод
класса human)
```

```
std::cout << stud->get_full_name() << std::endl;
```

```
// Вывод среднего балла студента
```

```
std::cout << "Средний балл: " << stud->get_average_score() << std::endl;
```

```
return 0;
```

```
}
```

В этом примере мы написали программу, которая создает объект класса `student`, сохраняя в нем его имя, фамилию, отчество и список оценок.

После инициализации объекта, происходит вывод полного имени студента с помощью функции `get_full_name`. Эта функция была унаследована от базового класса `human`.

Затем программа вычисляет средний балл студента и выводит его на экран. Этим занимается функция `get_average_score`, которую мы описали внутри

```
selevit@gentoo codelive_class_inheritance % ./human
Петров Иван Алексеевич
Средний балл: 3.22222
selevit@gentoo codelive_class_inheritance %
```

класса `student`.

Мы реализовали часть функционала для нашей базы данных института (я конечно утрирую, когда оперирую столь серьезными высказываниями про настоящую базу данных :)

Создание класса-наследника `teacher`

Нужно создать еще один класс, в котором будут храниться данные преподавателей. Дадим ему название — `teacher`. Как вы уже поняли, мы не

будем описывать все методы этого класса с нуля, а просто унаследуем его от класса `human`. Тогда, не нужно будет реализовывать хранение имени, фамилии и отчества препода. Это уже есть в базовом классе `human`.

Создайте файл `teacher.h`:

```
// teacher.h
```

```
#ifndef TEACHER_H_INCLUDED  
#define TEACHER_H_INCLUDED
```

```
#include "human.h"
```

```
#include <string>
```

```
class teacher : public human {
```

```
    // Конструктор класса teacher
```

```
    public:
```

```
        teacher(
```

```
            std::string last_name,
```

```
            std::string name,
```

```
            std::string second_name,
```

```
            // Количество учебных часов за семестр у преподавателя
```

```
            unsigned int work_time
```

```
        ) : human(
```

```
            last_name,
```

```
            name,
```

```
            second_name
```

```
        ) {
```

```
            this->work_time = work_time;
```

```
        }
```

```
    // Получение количества учебных часов
```

```
    unsigned int get_work_time()
```

```
    {
```

```
        return this->work_time;
```

```
    }
```

```
    private:
```

```
        // Учебные часы
```

```
        unsigned int work_time;
```

```
};
```

```
#endif // TEACHER_H_INCLUDED
```

У класса `teacher` появилось новое свойство — количество учебных часов, отведенное преподавателю на единицу времени (семестр). Весь остальной функционал наследуется от базового класса `human`. Если бы мы писали все с нуля, то одинакового кода бы получилось в разы больше, и его поддержка усложнилась бы на порядок.

Создание объекта класса `teacher`

Изменим содержимое файла `main.cpp`, чтобы проверить работу класса `teacher`.

```
#include <iostream>
```

```
#include "human.h"  
#include "teacher.h"
```

```
int main(int argc, char* argv[])  
{
```

```
// Количество учебных часов преподавателя  
unsigned int teacher_work_time = 40;
```

```
teacher *tch = new teacher("Васильков", "Петр", "Сергеевич",  
teacher_work_time);
```

```
std::cout << tch->get_full_name() << std::endl;  
std::cout << "Количество часов: " << tch->get_work_time() << std::endl;
```

```
return 0;  
}
```

Если сборка программы прошла без ошибок, то результат работы программы будет таким:

```
selevit@gentoo codelive_class_inheritance % ./human  
Васильков Петр Сергеевич  
Количество часов: 40  
selevit@gentoo codelive_class_inheritance %
```

Можно таким же образом создать класс, в котором будут храниться данные обслуживающего персонала или руководящего состава. Наследование используют, когда у каждой группы объектов есть общие параметры, но для каждой из этих групп нужно хранить более кастомные данные.

Также, мы можем создать класс, который будет описывать студента заочной формы обучения. Его мы унаследовали бы от класса `student`, добавив какие-либо дополнительные данные.

В класс `human` можно добавить еще больше свойств, которые будут описывать данные, имеющиеся у любого человека. Например, номер паспорта, дату рождения, прописку и место проживания.

Подобный подход позволяет в разы уменьшить дублирование кода в реальных проектах, и упростить его поддержку.

Когда нужно использовать конструктор

Если у класса много свойств — их совсем не обязательно задавать в конструкторе. Для сохранения отдельных свойств класса используют `set`-функции. Например, для сохранения номера паспорта, можно создать публичный метод `set_passport_number(std::string number)`, который будет принимать значение свойства и сохранять его в объекте, через переменную `this`.

Лабораторная работа №24.

Тема: Перегрузка методов.

Задание: Ознакомится с теоретическим материалом по теме «Полиморфизм». Рассмотреть применение этого принципа ООП на примерах перегрузки функций, перегрузки методов класса и перегрузки операторов.

Выполнение

Перегрузка функций в C++

Перегрузка функций в C++ используется, когда нужно сделать одно и то же действие с разными типами данных. Для примера, создадим простую функцию `max`, которая будет определять максимальное из двух целых чисел.

/ Функция max для целых чисел */*

```
int max(int num1, int num2)
```

```
{
```

```
    if (num1 > num2)
```

```
        return num1;
```

```
    return num2;
```



```
}
```

В эту функцию мы можем передавать только целочисленные параметры. Для того, чтобы сделать аналог этой функции для чисел с плавающей запятой, выполним перегрузку этой функции:

```
/* Функция max для чисел с плавающей запятой */
```

```
double max(double num1, double num2)
```

```
{
```

```
    if (num1 > num2)
```

```
        return num1;
```

```
    return num2;
```

```
}
```

Теперь, когда мы будем вызывать функцию **max** с целыми параметрами, то вызовется первая функция. А если с дробными — то вторая. Например:

```
// Здесь будет использоваться первый вариант функции max
```

```
int imax = max(1, 10);
```

```
// А здесь - второй
```

```
double dmax = max(1.0, 20.0);
```

Задание: попробуйте написать функцию сортировки массива пузырьком для целочисленных массивов. А затем перегрузить эту же функцию для массивов типа **double**.

Исходный код примера из урока.

```
#include <iostream>
```

```
/* Функция max для целых чисел */
```

```
int max(int num1, int num2)
```

```
{
```

```
    if (num1 > num2)
```

```
        return num1;
```

```
    return num2;
```

```
}
```

```
/* Функция max для чисел с плавающей запятой */
```

```
double max(double num1, double num2)
```

```
{
```

```
    if (num1 > num2)
```

```
        return num1;
```

```
    return num2;
```

```
}
```

```
int main()
```

```
{
```

```
    // Здесь будет использоваться первый вариант функции max
```

```
    int imax = max(1, 10);
```

```
    // А здесь - второй
```

```

double dmax = max(1.0, 20.0);
std::cout << "imax = " << imax << std::endl;
std::cout << "dmax = " << dmax << std::endl;
std::cin.get();
return 0;
}

```

Перегрузка методов класса в C++

Методы класса можно перегружать так же, как и обычные функции. Особенно это удобно, когда нужно сделать несколько конструкторов, которые будут принимать разные параметры.

Например, попробуем создать основу класса **decimal**, который реализует длинную арифметику для чисел произвольной точности. В таких случаях, обычно хранят число внутри строки, а логика математических операций реализуется через написание соответствующих операторов класса.

Сделаем так, чтобы в конструктор этого класса можно было передавать и строку и число типа **double**.

```

// Передача в конструктор строки
decimal num1("1000000.999999");
// Передача числа
decimal num2(10000.0);

```

Для того, чтобы класс поддерживал такую универсальность, мы сделаем два разных конструктора для строки и числа:

```

/**
 * Представим, что этот класс реализует длинную арифметику для чисел
 * любой
 * точности
 */
class decimal
{
public:
    /**
     * Конструктор, принимающий в качестве аргумента строку, содержащую
     * число
     */
    decimal(string number)
    {
        clog << "First constructor called\n";
    }
}

```

```

/**
 * Конструктор принимает число типа double
 */
decimal(double number)
{
    clog << "Second constructor called\n";
}

```

```

private:
    string number;
};

```

При передаче строки будет вызван первый конструктор, а при передаче числа — второй.

Полный текст программы:

```

#include <iostream>
#include <string>

```

```

using namespace std;

```

```

/**
 * Представим, что этот класс реализует длинную арифметику для чисел
 * любой точности
 */
class decimal
{
public:
    /**
     * Конструктор, принимающий в качестве аргумента строку,
     * содержащее число
     */
    decimal(string number)
    {
        clog << "First constructor called\n";
        this->number = number;
    }

    /**
     * Конструктор принимает число типа double
     */
    decimal(double number)
    {
        /**
         * преобразуем double в строку с максимально возможной точностью
         */
    }
}

```

```

        * и записываем полученное значение в this->number
        */
        clog << "Second constructor called\n";
    }

```

private:

```

    string number;
};

```

int main()

```

{
    // Будет вызван первый конструктор
    decimal num1("10000000.999999");
    // Будет вызван второй конструктор
    decimal num2(10000.0);
    cin.get();
    return 0;
}

```

Конечно, наш класс пока ничего не делает, потому что реализация длинной арифметики выходит за рамки данной статьи. Но на этом примере можно понять, когда может быть полезно использовать перегрузку методов класса.

Задание: попробуйте написать класс **student**, в конструктор которого можно будет передавать либо его имя и фамилию, либо имя и год рождения. При передаче года рождения, должен считаться примерный возраст студента. Пример использования класса:

```

student stud1("Иван", "Иванов");
student stud2("Иван", 1990);

```

Определение и перегрузка операторов класса в C++

В C++ можно определять пользовательские операторы для собственных типов данных. Оператор определяется, как обычная функция-член класса, только после определения возвращаемого типа ставится ключевое слово **operator**.

Пример определения оператора сложения:

```

int operator+ (int value) { return number + value; }

```

Оператор может быть унарным или бинарным. Унарный оператор не принимает никаких аргументов. Например, оператор отрицания — «!». Бинарный оператор принимает дополнительный параметр. Например, в случае со сложением, принимается второе слагаемое.

Чтобы прояснить картину, попробуем написать класс `simple_fraction`, который будет описывать простую дробь с целыми числителем и знаменателем. И определим операторы сложения, вычитания, умножения и деления для этого класса.

```
/*
 * Класс, описывающий простую дробь
 */
class simple_fraction
{
public:
    simple_fraction(int numerator, int denominator)
    {
        if (denominator == 0) // Ошибка деления на ноль
            throw std::runtime_error("zero division error");
        this->numerator = numerator;
        this->denominator = denominator;
    }

    // Определение основных математических операций для
    // простой дроби
    double operator+ (int val) { return number() + val; } //
    // Сложение
    double operator- (int val) { return number() - val; } //
    // Вычитание
    double operator* (int val) { return number() * val; } //
    // Умножение
    double operator/ (int val) // Деление
    {
        if (val == 0) {
            throw std::runtime_error("zero division error");
        }
        return number() / val;
    }

    // Получение значения дроби в виде обычного double-числа
    double number() { return numerator / (double) denominator;
    }
private:
    int numerator; // Числитель
    int denominator; // Знаменатель
};
```

Для операции деления, мы также сделали проверку деления на ноль.

Пример использования класса `simple_fraction`:

```
// Простая дробь 2/3
simple_fraction fr(2, 3);

double sum = fr + 10; // сумма
double diff = fr - 10; // разность
double factor = fr * 10; // произведение
double div = fr / 10; // частное
```

Операторы можно перегружать так же, как и обычные функции-члены класса. Например, можно перегрузить оператор сложения для двух простых дробей, который будет возвращать новую простую дробь. Тогда, нам придется привести дроби к общему знаменателю и вернуть другую простую дробь.

Задание: усовершенствуйте класс `simple_fraction`. Перегрузите операторы сложения, вычитания, умножения и деления так, чтобы можно было производить операции над двумя простыми дробями и получать новую простую дробь. Реализуйте приведение двух дробей к общему знаменателю.

Пример использования будущего класса:

```
simple_fraction fr1(2, 3);
simple_fraction fr2(3, 4);

// 2/3 + 3/4 – это 17/12
simple_fraction sum = fr1 + fr2;
```

Лабораторная работа №25.

Тема: Создание проекта с использованием кнопочных компонентов.

Цель работы: сформировать навыки разработки приложений с использованием кнопочных компонентов в среде визуального программирования Visual Studio, изучить особенности их использования

Задание 1. Используя кнопочные компоненты Button, разработать программу – калькулятор, выполняющий простейшие действия.



Методические указания по выполнению задания:

1. Запустите интегрированную среду разработчика Visual Studio. Создайте новый проект и назовите его Калькулятор.
2. Разместите на форме одно поле ввода, дайте ему имя Disp и очистите свойство Text.
3. Добавьте на форму 12 кнопок для цифр, арифметических действий, знака «равно» и операции «сброс». Разместите их так, как на рисунке.
4. Дайте кнопкам-действиям имена Plus, Minus, Mul, Div (сложение, вычитание, умножение и деление), а кнопке «равно» — имя Calc.
5. Для действий и кнопки C установите жирный шрифт, а для кнопки C дополнительно — красный цвет шрифта.
6. Кнопка C должна просто стирать содержимое поля ввода Disp. То есть, нужно вызвать метод Clear. Добавьте обработчик события OnClick для кнопки C.
7. Выделите все кнопки-цифры и определите для них общий обработчик события Click.
8. Запустите программу.
9. При работе программы вы увидели, что с клавиатуры можно ввести буквы, которые нам совсем не нужны. Когда пользователь нажмет клавишу в поле ввода, возникает событие KeyPress, которое можно перехватить, установив соответствующий обработчик. В нем все «ненужные» символы заменяются символом с кодом 0, который не изменяет содержимое поля. Создайте обработчик события KeyPress для поля ввода Disp.

10. Проверьте работу программы.
11. Далее необходимо организовать вычисления. Нам нужны две переменных для хранения чисел и одна символьная переменная, в которую будем записывать тип операции. Объявите в начале программы две вещественных переменные `x1` и `x2` типа `Double` и одну символьную строку `oper`.
12. Когда мы нажимаем на одну из кнопок-операций, нужно запомнить введенное число в переменной `x1` и тип операции в переменной `oper`. Тип операции (надпись на кнопке) легко узнать, обратившись к свойству `text`. Поэтому можно установить для всех кнопок-операций один обработчик. Выделите все кнопки-операции и создайте для них один обработчик события `Click`.
13. При нажатии на кнопку «Равно» нужно прочитать из поля ввода второе число и выполнить операцию. При этом первое число и тип операции уже должны находиться в переменных `x1` и `oper`. Поскольку в результате деления может получиться число с дробной частью, переменная для хранения результата (назовем ее `res`) тоже должна быть вещественной.
14. Введите обработчик события `Click` для кнопки «Равно».
15. Запустите программу и проверьте ее работу. Учтите, что для ввода второго числа нужно сначала очистить экран кнопкой «С».
16. Закройте окно программы и затем запустите ее заново. Введите какое-нибудь число, а затем сразу щелкните по кнопке «равно». Что получилось? Попробуйте объяснить этот эффект, учитывая, что при создании глобальной символьной переменной `oper` в нее записана пустая строка. Проблема вызвана тем, что мы еще не задали операцию, а уже попытались что-то вычислить. При этом ни один условный оператор не сработал, и значение переменной `res` не изменялось. Переменная `res` объявлена в процедуре, то есть, она локальная. Память для локальных переменных выделяется в стеке при каждом новом вызове процедуры, причем эти ячейки не обнуляются. Поэтому в переменной `res` в нашем последнем эксперименте осталось постороннее значение, которое программа и вывела на экран. Если вы были очень внимательны, можно было заметить, что при трансляции программы в окне `Message` (в нижнем левом углу) было выдано предупреждение «Переменной `res`, возможно, не будет присвоено никакого значения». Формально это не ошибка, и программа может запуститься, однако к предупреждениям нужно относиться внимательно, потому что они могут указать на скрытые логические ошибки (ошибки в алгоритме), как в нашем случае. Какой же выход? Проще всего сделать так: если в переменной `oper` записана пустая строка, мы просто выйдем из процедуры с помощью оператора `Exit`.
17. Доведите работу до конца.
18. Сделайте отчет.

Лабораторная работа №26.

Тема: Создание проекта с использованием компонентов для работы с текстом.

Цель: Научиться правильно использовать в проекте компоненты для работы с текстом.

Ход занятия:

1. Изучите в учебнике Н.Кульпина «Основы программирования в Visual C++2010» свойства компонентов Label (стр.61-64), TextBox (стр.64-69).
2. Изучите свойства компонента RichTextBox, информацию о нем можно найти на сайтах:

<https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/controls/richtextbox-control-windows-forms>

3. Ответьте на вопросы:
 - a. Для чего можно использовать каждый из них?
 - b. Чем отличается TextBox от RichTextBox?
4. Выполните задания:
 - a. Создайте проект с формой, на которой имеется два текстовых поля (TextBox) и кнопка. В первое поле вводится текст, второе поле закрыто для редактирования. По щелчку на кнопке текст, введенный в первое текстовое поле, переносится во второе текстовое поле (при этом исчезает из первого).
 - b. Модифицируйте программу так, чтобы текст можно было вводить в оба текстовых окна и по щелчку на кнопке их содержимое менялось бы местами.
 - c. Создайте проект с формой. На форме разместите компонент RichTextBox. Растяните его на всю форму. Текст, введенный в текстовое поле отформатируйте так, чтобы разные части текста имели разный размер, начертание и цвет. Для этого используйте стандартное диалоговое окно FontDialog. Для вызова этого компонента при работе программы опишите процедуру нажатия на кнопку (кнопка должна тоже быть на форме).

```
private void Button1_Click(object sender, EventArgs e) {
    fontDialog1->Font = richTextBox1->SelectionFont;
    fontDialog1->Color = richTextBox1->SelectionColor;
    fontDialog1->ShowColor = true;
    if(fontDialog1->ShowDialog() != System::Windows::Forms::DialogResult::Cancel) {
        richTextBox1->SelectionFont = fontDialog1->Font;
        richTextBox1->SelectionColor = fontDialog1->Color;
    }
}
```

Лабораторная работа №27.

Тема: Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.

Цель: Научиться правильно использовать в проекте компоненты ввода и отображения чисел, дат и времени.

Ход занятия:

1. Изучите материал учебника Зиборова «MS Visual c++2010 в среде NET» стр.24-26. по работе с датами. Создайте приложение, добавьте на форму элемент DateTimePicker и добавьте код из учебника. Проверьте работу приложения.
2. Изучите материал учебника Б.Пахомова «С С++ и MS Visual С++ 2010 для начинающих» стр.457-464 (446-475).
3. Создайте приложение, выполните все действия по учебнику, проверьте работу приложения.
4. Самостоятельно изучите элемент MonthCalender и все его свойства.

Лабораторная работа №28.

Тема: Создание проекта с использованием компонентов стандартных диалогов и системы меню.

Цель: Научиться правильно использовать в проекте компоненты стандартных диалогов и системы меню.

Ход занятия:

1. Изучите по учебнику Н.Культина компоненты ToolStrip (стр.130) и MenuStrip(стр.133).
2. Создайте проект «TextEdit». Добавьте строку меню MenuStrip1 со стандартными компонентами. (щелкните по черному треугольнику и выберите «Добавить стандартные компоненты меню»).
3. Добавьте на форму панель инструментов ToolStrip со стандартными компонентами.
4. Добавьте компонент RichTextBox1 для работы с текстом. Привяжите его к родительскому окну, чтобы он изменял свои размеры одновременно с формой.
5. Изучите по учебнику Н.Культина компоненты OpenFileDialog стр.136), SaveFileDialog(стр.139). Описание процедуры вызова этого окна есть в предыдущей лабораторной работе. Опишите процедуру сохранения текста в формате txt и rtf. Для этого используется маска.
6. С пунктами 4-5 вы уже знакомы по предыдущей лабораторной работе.
7. Открытие и сохранение файла необходимо привязать и к соответствующему пункту меню, и к кнопке на панели инструментов.
8. Чтобы ознакомиться с FontDialog, ColorDialog, рассмотрите материалы сайта <https://docs.microsoft.com/ru-ru/dotnet/framework/winforms/controls/dialog-box-controls-and-components-windows-forms>
9. Для выбора шрифта добавьте на панель инструментов кнопку «Аа» и в меню «Правка» пункт «Шрифт». Кнопку «Аа» создаем так: сначала

добавляем в панель инструментов кнопку, потом устанавливаем ее свойство `toolStripButton1.Image` картинку, иконку, которую можно скачать из интернета.

10. Соответственно привяжите к кнопке панели инструментов и пункту меню процедуру вызова диалогового окна `FontDialog` и установки атрибутов текста.
11. Добавьте на форму элемент `ContextMenuStrip`. О нем можно прочитать в учебнике *MS Visual C++ 2010 в среде .NET. Библиотека программиста*. Автор Зиборов Виктор Владимирович (стр.288). Этот элемент обязательно нужно связать с элементом, по которому будем щелкать правой кнопкой мыши, чтобы вызвать меню. Выделите компонент `RichTextBox1` и в свойствах найдите `ContextMenuStrip` и справа выберите `ContextMenuStrip1`.
12. Добавьте пункты меню для работы с текстом, такие как «Создать», «Выделить все», «Копировать», «Вставить», «Вырезать». В контекстном меню обычно указываются наиболее часто применяемые команды.
13. Добавьте в панель инструментов кнопки выравнивания по левому, правому краю и по центру. Иконки для них тоже скачайте в интернете. Опишите процедуры нажатия на эти кнопки. Например, для выравнивания по правому краю используйте строку **`richTextBox1->SelectionAlignment = HorizontalAlignment::Right;`**
14. Опишите и процедуру очистки текстового поля. Эта процедура подойдет и для пункта меню «Файл»-> «Создать», и для кнопки Создать на панели инструментов, и для пункта контекстного меню «Очистить выделенное» при условии, что выбран пункт «Выделить все».
15. Примените полученные знания и умения и разработайте программу для простейшего текстового редактора.

Принципы работы программы таковы:

1. В текстовое поле можно вводить текст.
2. Набранный текст можно сохранять с помощью диалогового окна `SaveFileDialog`.
3. В текстовом поле можно открыть любой текстовый документ в формате `txt` или `rtf`, для этого используется диалоговое окно `OpenFileDialog`.
4. Можно создавать новый документ.
5. Можно редактировать текст с помощью пунктов меню Копировать, Вырезать, Вставить и соответствующих им кнопок на панели инструментов.
6. Любой выделенный фрагмент текста можно форматировать, то есть изменять его цвет, размер и начертание. Для этого необходимо вызвать диалоговое окно `FontDialog` и выбрать нужные атрибуты шрифта.

Лабораторная работа №29.

Тема: Разработка оконного приложения.

Цель: Научиться использовать при разработке полнофункционального приложения различные компоненты.

Задание: Разработать на основе компонента RichTextVox1 (улучшенного текстового окна) программу текстового редактора, позволяющего пользователю набирать, сохранять и загружать текст в область редактора, изменять шрифт и цвет области с текстом (цвет фона), осуществлять поиск и замену текста, настраивать принтер и задавать параметры вывода документа с помощью обычного и контекстного меню и стандартных диалоговых окон.

Порядок выполнения лабораторной работы

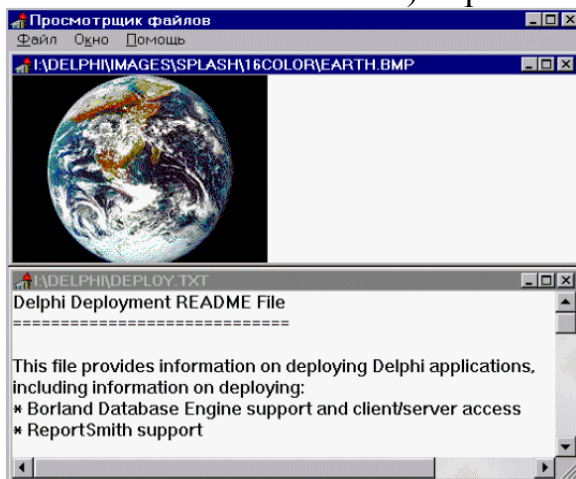
1. Ознакомиться с теоретическими сведениями о компонентах, реализующих пользовательский интерфейс.
2. Создать проект, разместив на форме, компоненты обычного и контекстного меню, а также компоненты-диалоги, представленные в теоретических сведениях.
3. Создать меню программы.

Лабораторная работа №30.

Тема: Разработка оконного приложения с несколькими формами.

Цель: Научиться использовать при разработке полнофункционального приложения различные компоненты.

Задание: Создать приложение с многодокументным интерфейсом (Multiple Document Interface - MDI). Приложение должно иметь следующий интерфейс:



Критерии оценки:

- оценка «зачтено» выставляется студенту, если правильно выполнены все задания лабораторной работы;

- оценка «не зачтено» выставляется студенту, если неправильно выполнены некоторые задания или выполнены не все задания лабораторной работы.

Комплект заданий для выполнения
практических работ

Общее задание:

1. Составить блок-схему алгоритма программы
2. Написать и отладить программу
3. Составить отчет по практической работе

Практическая работа №1 "Составление программ линейной структуры"

Задания к практической №1 по вариантам	
Вариант 1:	Ввести два ненулевых числа. Найти их сумму, разность, произведение и частное. Вывести полученные значения.
Вариант 2:	Найти периметр и площадь прямоугольного треугольника. Ввести длины его катетов a и b . Вывести полученные значения.
Вариант 3:	Ввести длину ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба. Вывести полученные значения.
Вариант 4:	Найти длину окружности и площадь круга заданного радиуса R . В качестве значения P_i использовать 3.14. Вывести полученные значения.
Вариант 5:	Найти площадь кольца, внутренний радиус которого равен R_1 , а внешний радиус равен R_2 ($R_1 < R_2$). В качестве значения P_i использовать 3.14. Ввести радиусы R_1 и R_2 . Вывести полученное значение.
Вариант 6:	Ввести длину окружности. Найти площадь круга, ограниченного этой окружностью. В качестве значения P_i использовать 3.14. Вывести полученное значение.
Вариант 7:	Ввести площадь круга. Найти длину окружности, ограничивающей этот круг. В качестве значения P_i использовать 3.14. Вывести полученное значение.
Вариант 8:	Ввести длину и ширину прямоугольника. Найти его площадь и периметр. Вывести полученные значения.
Вариант 9:	Ввести два положительных числа a и b ($a > b$). Определить на сколько первое число больше второго и во сколько раз первое число больше второго. Результаты вывести на экран.
Вариант 10:	Из пункта А в пункт Б вышел пешеход. Написать программу, вычисляющую и выводящую на экран Время, затраченное пешеходом на весь путь, если известны расстояние от пункта А до пункта Б и скорость пешехода.

Соответствие номеров по списку в журнале вариантам																										
Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
Вариант	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6
г	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6

Практическая работа №2 "Составление программ математических вычислений"

Задания к практической №2 по вариантам	
Вариант 1:	Вычислить: $\frac{b + \sqrt{b^2 + 4ac}}{2a} - a^3c + b^2$. Тест: a=3; b=2; c=1. Результат: -22.
Вариант 2:	Вычислить: $3^a - 4b + (c - \sqrt{ab})$. Тест: a=2; b=-8; c=10. Результат: 47.
Вариант 3:	Вычислить: $\cos(a) - \frac{5\sqrt{b^3 + ac }}{2c}$. Тест: a=0; b=4; c=-2. Результат: 11.
Вариант 4:	Вычислить: $\frac{b^a + \operatorname{tg}(c)}{a \cos(c)} - 2a^3$. Тест: a=2; b=4; c=0. Результат: -8
Вариант 5:	Вычислить: $\frac{2ac - \sqrt{\sin(2b)}}{4a - 2bc} c - b$. Тест: a=1; b=0; c=2. Результат: 2.
Вариант 6:	Вычислить: $\frac{2b + \sqrt{\cos(a) + bc}}{b - \log_{10}(2c)}$. Тест: a=0; b=3; c=5. Результат: 2.
Вариант 7:	Вычислить: $\frac{x + y}{y + 1} - \frac{xy - 12}{6 + x}$. Тест: x=6; y=4. Результат: 1.
Вариант 8:	Вычислить: $\frac{a}{c} * \frac{b}{d} - \frac{ab - c}{cd + 1}$. Тест: a=6; b=4; c=3; d=2. Результат: 1.
Вариант 9:	Вычислить: $\frac{a^2 + c}{b} * \left(4 - \frac{a + b}{c}\right)$. Тест: a=4; b=2; c=6. Результат: 33.
Вариант 10:	Вычислить: $\arcsin\left(\frac{a + b}{c} - 2\right) + \frac{2^a - c}{b}$. Тест: a=4; b=2; c=3. Результат: 6,5.
Соответствие номеров по списку в журнале вариантам	
Номер по списку	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Вариант	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Практическая работа №3 "Составление программ разветвляющейся структуры с использованием инструкции if"

Задания к практической №3 по вариантам	
Вариант 1:	Ввести три действительных числа от -10 до 10. Возвести в квадрат те из них, значения которых неотрицательны, и в третью степень – отрицательные. Вывести полученные значения.
Вариант 2:	Даны два угла треугольника (в градусах). Определить существует ли такой треугольник, и если да, то будет ли он прямоугольным. Результаты вывести на экран.
Вариант 3:	Из трех введенных чисел выбрать наименьшее. Результат вывести на экран.
Вариант 4:	Ввести вещественные координаты точки, не лежащей на координатных осях ОХ и ОУ. Вывести номер координатной четверти, в которой находится данная точка.
Вариант 5:	Дано целое число, лежащее в диапазоне от -99 до 99. Вывести строку — словесное описание данного числа вида "отрицательное двузначное число", "нулевое число", "положительное однозначное число" и т.д.
Вариант 6:	Даны две переменные целого типа: А и В. Если их значения не равны, то присвоить каждой переменной максимальное из этих значений, а если равны, то присвоить переменным нулевые значения. Вывести полученные значения.
Вариант 7:	Подсчитать количество отрицательных среди чисел a,b,c. Результаты вывести на экран.
Вариант 8:	Дан номер некоторого года (положительное целое число). Вывести число дней в этом году, учитывая, что обычный год насчитывает 365 дней, а високосный — 366 дней. Високосным считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400 (например, годы 300, 1300 и 1900 не являются високосными, а 1200 и 2000 — являются). Результат вывести на экран.
Вариант 9:	Дано целое число. Определить, является ли оно четным или нет. Результат вывести на экран.
Вариант 10:	Даны две точки А(x1,y1) и В(x2,y2). Написать программу, которая определяет какая из точек расположена ближе к началу координат.

Соответствие номеров по списку в журнале вариантам																														
Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Вариант	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10

Практическая работа №4 "Составление программ разветвляющейся структуры с использованием инструкции switch"

Задания к практической №4 по вариантам																														
Вариант 1:	Дан номер месяца (1 — январь, 2 — февраль, ...). Вывести название соответствующего времени года ("зима", "весна" и т.д.).																													
Вариант 2:	Дан номер месяца (1 — январь, 2 — февраль, ...). Вывести число дней в этом месяце для невисокосного года.																													
Вариант 3:	Дано целое число в диапазоне 0 – 9. Вывести строку — название соответствующей цифры на русском языке (0 — "ноль", 1 — "один", 2 — "два", ...).																													
Вариант 4:	Дано целое число в диапазоне 1 – 5. Вывести строку — словесное описание соответствующей оценки (1 — "плохо", 2 — "неудовлетворительно", 3 — "удовлетворительно", 4 — "хорошо", 5 — "отлично").																													
Вариант 5:	Написать программу, которая по номеру дня недели (целому числу от 1 до 7) выдает в качестве результата название дня недели.																													
Вариант 6:	Единицы длины пронумерованы следующим образом: 1 — дециметр, 2 — километр, 3 — метр, 4 — миллиметр, 5 — сантиметр. Дан номер единицы длины и длина отрезка L в этих единицах (вещественное число). Вывести длину данного отрезка в метрах.																													
Вариант 7:	Арифметические действия над числами пронумерованы следующим образом: 1 — сложение, 2 — вычитание, 3 — умножение, 4 — деление. Дан номер действия и два числа A и B (B не равно нулю). Выполнить над числами указанное действие и вывести результат.																													
Вариант 8:	Единицы массы пронумерованы следующим образом: 1 — килограмм, 2 — миллиграмм, 3 — грамм, 4 — тонна, 5 — центнер. Дан номер единицы массы и масса тела M в этих единицах (вещественное число). Вывести массу данного тела в килограммах.																													
Вариант 9:	Написать программу, которая по последней цифре числа позволяет определить последнюю цифру его квадрата.																													
Вариант 10:	В старояпонском календаре был принят 12-летний цикл. Годы внутри цикла носили названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи. Написать программу, которая вводит номер некоторого года и печатает его название по старояпонскому календарю. 1996 г. — год Крысы — начало очередного цикла.																													
Соответствие номеров по списку в журнале вариантам																														
Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Вариант	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10

Практическая работа №5 "Составление программ циклической структуры с использованием инструкции for"

Задания к практической №5 по вариантам																														
Вариант 1:	Даны два целых числа A и B ($A < B$). Вывести все целые числа, расположенные между данными числами (включая сами эти числа), в порядке их возрастания. Использовать цикл for.																													
Вариант 2:	Вывести на экран квадраты всех целых чисел от 0 до 10. Использовать цикл for.																													
Вариант 3:	Дано вещественное число A и целое число N (> 0). Вывести все целые степени числа A от 1 до N. Использовать цикл for.																													
Вариант 4:	10 раз вывести на экран (каждый раз с новой строки) слово «Привет». Использовать цикл for.																													
Вариант 5:	Дано вещественное число A и целое число N ($N > 0$). Вывести A в степени N: $A^N = A \cdot A \cdot \dots \cdot A$ (числа A перемножаются N раз). Использовать цикл for.																													
Вариант 6:	Ввести с клавиатуры N целых чисел ($N > 0$) и вывести на экран квадраты этих чисел. Использовать цикл for.																													
Вариант 7:	Дано число N ($N > 0$). Вывести на экран все четные числа от 0 до N. (не используя логические операторы). Использовать цикл for.																													
Вариант 8:	Даны два целых числа A и B ($A < B$). Вывести все целые числа, расположенные между данными числами (включая сами эти числа), в порядке их убывания. Использовать цикл for.																													
Вариант 9:	Дано число N ($N > 0$). Вывести на экран N первых нечетных положительных чисел (не используя логические операторы). Использовать цикл for.																													
Вариант 10:	Составить программу, определяющую является ли введенное число простым. (Простое число делится нацело только на само себя и на 1). Использовать цикл for.																													
Соответствие номеров по списку в журнале вариантам																														
Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Вариант	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10

Практическая работа №6 "Составление программ циклической структуры с использованием инструкции while"

Задания к практической №6 по вариантам																														
Вариант 1:	<p style="text-align: center;">$p(n) = \frac{1}{1}; \frac{1}{4}; \dots; \frac{1}{n^2} \dots$</p> <p>Дана последовательность: $p(n) = \frac{1}{1}; \frac{1}{4}; \dots; \frac{1}{n^2} \dots$. Составить программу с использованием цикла while, выводящую на экран значение и номер члена последовательности, меньшего 0,1. Тест: n=4, p(n)=0,0625. Использовать цикл while.</p>																													
Вариант 2:	<p>Дано вещественное число A и целое число N (> 0). Вывести все целые степени числа A от 1 до N. Использовать цикл while.</p>																													
Вариант 3:	<p style="text-align: center;">$p(n) = \frac{1}{1}; \frac{1}{2}; \dots; \frac{1}{n} \dots$</p> <p>Дана последовательность: $p(n) = \frac{1}{1}; \frac{1}{2}; \dots; \frac{1}{n} \dots$. Составить программу с использованием цикла while, выводящую на экран значение и номер члена последовательности, меньшего 0,5. Тест: n=3, p(n)=0,3333. Использовать цикл while.</p>																													
Вариант 4:	<p>Дано вещественное число A и целое число N (N > 0). Вывести A в степени N: AN = A•A•...•A (числа A перемножаются N раз). Использовать цикл while.</p>																													
Вариант 5:	<p style="text-align: center;">$p(n) = \frac{1}{2}; \frac{1}{4}; \dots; \frac{1}{2^n} \dots$</p> <p>Дана последовательность: $p(n) = \frac{1}{2}; \frac{1}{4}; \dots; \frac{1}{2^n} \dots$. Составить программу с использованием цикла while, выводящую на экран значение и номер члена последовательности, меньшего 0,2. Тест: n=3, p(n)=0,1667. Использовать цикл while.</p>																													
Вариант 6:	<p>Даны два целых числа A и B (A < B). Вывести все целые числа, расположенные между данными числами (включая сами эти числа), в порядке их возрастания. Использовать цикл while.</p>																													
Вариант 7:	<p style="text-align: center;">$p(n) = \frac{1}{2}; \frac{2}{4}; \dots; \frac{n}{2^n} \dots$</p> <p>Дана последовательность: $p(n) = \frac{1}{2}; \frac{2}{4}; \dots; \frac{n}{2^n} \dots$. Составить программу с использованием цикла while, выводящую на экран значение и номер члена последовательности, меньшего 0,1. Тест: n=6, p(n)=0,0938. Использовать цикл while.</p>																													
Вариант 8:	<p>Вводить с клавиатуры целые числа (>0) и выводить на экран квадраты этих чисел до тех пор, пока не будет введено число 0. Использовать цикл while.</p>																													
Вариант 9:	<p style="text-align: center;">$p(n) = \frac{1}{2}; \frac{4}{4}; \dots; \frac{n^2}{2^n} \dots$</p> <p>Дана последовательность: $p(n) = \frac{1}{2}; \frac{4}{4}; \dots; \frac{n^2}{2^n} \dots$. Составить программу с использованием цикла while, выводящую на экран значение и номер члена последовательности, меньшего 0,1. Тест: n=10, p(n)=0,0977. Использовать цикл while.</p>																													
Вариант 10:	<p>Дано число N (N>0). Вывести на экран N первых нечетных положительных чисел (не используя логические операторы). Использовать цикл while.</p>																													
Соответствие номеров по списку в журнале вариантам																														
Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22								
Вариант	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10

Практическая работа №7 "Составление программ циклической структуры с использованием инструкции do...while"

Задания к практической №7 по вариантам	
Вариант 1:	Дано вещественное число A и целое число N (> 0). Вывести все целые степени числа A от 1 до N. Использовать цикл do...while.
Вариант 2:	Дана последовательность: $p(n) = \frac{1}{1}; \frac{1}{4}; \dots; \frac{1}{n^2} \dots$. Составить программу с использованием цикла while, выводящую на экран значение и номер члена последовательности, меньшего 0,1. Тест: n=4, p(n)=0,0625. Использовать цикл do...while.
Вариант 3:	Дано вещественное число A и целое число N (N > 0). Вывести A в степени N: AN = A•A•...•A (числа A перемножаются N раз). Использовать цикл do...while.
Вариант 4:	Дана последовательность: $p(n) = \frac{1}{1}; \frac{1}{2}; \dots; \frac{1}{n} \dots$. Составить программу с использованием цикла while, выводящую на экран значение и номер члена последовательности, меньшего 0,5. Тест: n=3, p(n)=0,3333. Использовать цикл do...while.
Вариант 5:	Даны два целых числа A и B (A < B). Вывести все целые числа, расположенные между данными числами (включая сами эти числа), в порядке их возрастания. Использовать цикл do...while.
Вариант 6:	Дана последовательность: $p(n) = \frac{1}{2}; \frac{1}{4}; \dots; \frac{1}{2^n} \dots$. Составить программу с использованием цикла while, выводящую на экран значение и номер члена последовательности, меньшего 0,2. Тест: n=3, p(n)=0,1667. Использовать цикл do...while.
Вариант 7:	Вводить с клавиатуры целые числа (>0) и выводить на экран квадраты этих чисел до тех пор, пока не будет введено число 0. Использовать цикл do...while.
Вариант 8:	Дана последовательность: $p(n) = \frac{1}{2}; \frac{2}{4}; \dots; \frac{n}{2^n} \dots$. Составить программу с использованием цикла while, выводящую на экран значение и номер члена последовательности, меньшего 0,1. Тест: n=6, p(n)=0,0938. Использовать цикл do...while.
Вариант 9:	Дано число N (N>0). Вывести на экран N первых нечетных положительных чисел (не используя логические операторы). Использовать цикл do...while.
Вариант 10:	Дана последовательность: $p(n) = \frac{1}{2}; \frac{4}{4}; \dots; \frac{n^2}{2^n} \dots$. Составить программу с использованием цикла while, выводящую на экран значение и номер члена последовательности, меньшего 0,1. Тест: n=10, p(n)=0,0977. Использовать цикл do...while.
Вариант 11:	Составить программу, которая выводит на экран меню из 2 пунктов (1-продолжить; 2-выход) до тех пор, пока пользователь не введет число 2, соответствующее выходу из программы. Использовать цикл do...while.
Вариант 12:	Составить программу с использованием цикла do-while, которая запрашивает пароль (четырёхзначное целое число) до тех пор, пока он не будет правильно введен. Использовать цикл do... while.

Соответствие номеров по списку в журнале вариантам																														
Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Вариант	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6

Практическая работа №8 "Составление программ вычисления суммы"

Задания к практической №8 по вариантам Во всех вариантах считать $N > 0$	
Вариант 1:	Составить программу вычисления суммы вводимых с клавиатуры чисел. Условие окончания ввода – ввод числа 0.
Вариант 2:	Дано натуральное число N . Вычислить Тест: $N=4, S=0.6875$ $S = 1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \dots + (-1)^N \cdot \frac{1}{2^N}$
Вариант 3:	Составить программу подсчета количества вводимых с клавиатуры чисел. Условие окончания ввода – ввод числа 0
Вариант 4:	Дано натуральное число N . Вычислить Тест: $N=4, S=1.183865$ $S = \frac{1}{1^2} + \frac{1}{3^2} + \frac{1}{5^2} + \frac{1}{7^2} + \dots + \frac{1}{(2N+1)^2}$
Вариант 5:	Составить программу вычисления суммы N первых целых положительных чисел. Количество чисел N вводить с клавиатуры.
Вариант 6:	Дано натуральное число $N > 0$. Вычислить Тест: $N=3, S=4.833333$ $S = \frac{2}{1} + \frac{3}{2} + \frac{4}{3} + \dots + \frac{N+1}{N}$
Вариант 7:	Найти количество первых целых положительных чисел, сумма которых не превышает S . Значение S вводить с клавиатуры.
Вариант 8:	Дано натуральное число $N > 0$. Вычислить Тест: $N=3, S=386$ $S = 1 \cdot 2 + 2 \cdot 3 \cdot 4 + \dots + N \cdot (N+1) \cdot (N+2) \dots 2N$
Вариант 9:	Ввести десять чисел. Вывести их среднее арифметическое..
Вариант 10:	Ввести целое число N и набор из N вещественных чисел. Вывести сумму и произведение чисел из данного набора.

Соответствие номеров по списку в журнале вариантам																																	
Номер по списку	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	3			
Вариант	1	2	3	4	5	6	7	8	9	1	0	1	2	3	4	5	6	7	8	9	1	0	1	2	3	4	5	6	7	8	9	1	0

Практическая работа №9 "Составление программ на сочетание циклов и условий"

Задания к практической №9 по вариантам	
<p>Для решения заданий следует использовать "однопроходные" алгоритмы, позволяющие получить требуемый результат после однократного просмотра набора исходных данных. Однопроходные алгоритмы обладают важным преимуществом: для решения задачи не требуется хранить в памяти одновременно весь набор данных. Поэтому при программировании таких алгоритмов не нужно использовать массивы.</p>	
Вариант 1:	Найти сумму четных чисел во вводимой с клавиатуры последовательности чисел. Условие окончания ввода – ввод числа 0.
Вариант 2:	Найти максимальный элемент из десяти целых чисел, вводимых с клавиатуры.
Вариант 3:	Найти номер первого минимального элемента во вводимой с клавиатуры последовательности чисел. Условие окончания ввода – ввод числа 0.
Вариант 4:	Составить программу, вычисляющую сумму положительных чисел во вводимой с клавиатуры последовательности. Условие окончания ввода – ввод числа 0.
Вариант 5:	Найти номер первого максимального элемента из десяти целых чисел, вводимых с клавиатуры.
Вариант 6:	Найти максимальный элемент во вводимой с клавиатуры последовательности чисел. Условие окончания ввода – ввод числа 0.
Вариант 7:	Найти минимальный нечетный из данных десяти ненулевых целочисленных элементов, вводимых с клавиатуры. Если требуемые элементы отсутствуют, то вывести 0.
Вариант 8:	Найти сумму положительных чисел во вводимой с клавиатуры последовательности чисел. Условие окончания ввода – ввод числа 0.
Вариант 9:	Даны два числа $a > 0$ и $b > 0$ (a меньше b). Найти сумму целых четных чисел, расположенных в интервале (a, b)
Вариант 10:	Найти количество отрицательных чисел во вводимой с клавиатуры последовательности. Условие окончания ввода – ввод числа 0.
Соответствие номеров по списку в журнале вариантам	
Номер по списку	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Вариант	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Практическая работа №10 "Составление программ с использованием случайных чисел"

Задания к практической №10 по вариантам																															
Вариант 1:	Подсчитать сколько раз введенное с клавиатуры число N встретиться среди 10 случайных чисел сгенерированных в диапазоне от N-5 до N+5 включительно. Сгенерированные числа и результаты подсчета вывести на экран.																														
Вариант 2:	Вычислить среднее арифметическое 5 случайных чисел, заданных в диапазоне от 10 до 20 включительно. Сгенерированные числа и результат расчета вывести на экран.																														
Вариант 3:	Найти минимальное число из 10 случайных чисел. Диапазон генерации случайного числа задается пользователем в начале работы программы.																														
Вариант 4:	Вычислить сумму N случайных чисел, заданных в диапазоне от -5 до 5 включительно. Сгенерированные числа и результат расчета вывести на экран.																														
Вариант 5:	Найти порядковый номер максимального числа в последовательности из 10 случайных чисел, заданных в диапазоне от 5 до 10.																														
Вариант 6:	Найти сумму четных чисел в последовательности из 10 случайных чисел. Диапазон генерации случайного числа задается пользователем в начале работы программы. Сгенерированные числа и результат расчета вывести на экран.																														
Вариант 7:	Подсчитать количество нечетных чисел в последовательности из N случайных чисел, заданных в диапазоне от 2 до 10. Сгенерированные числа и результат расчета вывести на экран.																														
Вариант 8:	Найти максимальное число из N случайных чисел. Диапазон генерации случайного числа задается пользователем в начале работы программы.																														
Вариант 9:	Найти количество чисел больших X (X лежит в диапазоне от -A до A) в последовательности из N случайных чисел, сгенерированных в диапазоне от -A до A включительно. X,N,A - вводить с клавиатуры. Сгенерированные числа и результат расчета вывести на экран.																														
Вариант 10:	Случайным образом сгенерировать дату и время. Диапазон изменения годов: 2000 - 2010, месяцев: 1 - 12, дней: 1-31, часов: 0 - 23, минут: 0 - 59. Вывести сгенерированное значение в формате: ДД/ММ/ГГГГ ЧЧ:ММ.																														
Соответствие номеров по списку в журнале вариантам																															
Номер по списку	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td> </tr> </table>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
Вариант	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <tr> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td> </tr> </table>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		

Практическая работа №11 "Составление программ на использование одномерных массивов"

Задания к практической №11 по вариантам	
Вариант 1:	Найти минимальный элемент массива из 5 чисел. Массив заполняется в процессе работы программы случайными числами от 1 до 9. Вывести на экран значения элементов массива результат работы программы.
Вариант 2:	Найти сумму элементов массива из 5 чисел. Массив заполняется в процессе работы программы случайными числами от 2 до 10. Вывести на экран значения элементов массива результат работы программы.
Вариант 3:	Массив А из 10 элементов заполняется случайными числами от 0 до 9. Сформировать новый массив В, заполненный элементами массива А в обратном порядке (Пример: А:0,1,2,3,4,5,6,7,8,9 В:9,8,7,6,5,4,3,2,1,0) Вывести на экран массивы А и В.
Вариант 4:	Массив А из 10 элементов заполняется случайными числами от 0 до 9. Сформировать массивы В и С, заполненные элементами массива А с четными и нечетными индексами соответственно. Нулевой индекс считать четным. (Пример А:0,2,4,6,8,10,12,14,16,18 В:0,4,8,12,16 С:2,6,10,14,18) Вывести на экран массивы А, В и С.
Вариант 5:	Найти сумму элементов массива А из 10 элементов, заполненного случайными числами от 0 до 5. Исходный массив и результат работы программы вывести на экран.
Вариант 6:	Массив А из 10 элементов заполнен случайными числами от 2 до 7. Вычислить массив В, Элементы которого вычисляются по формуле: $V[i]=A[i]+10$.
Вариант 7:	Найти максимальный элемент массива из 10 чисел. Массив заполняется в процессе работы программы случайными числами от 1 до 9. Вывести на экран значения элементов массива и результат работы программы.
Вариант 8:	Массив А из 5 элементов заполнен случайными числами от 3 до 9. Осуществить циклический сдвиг элементов массива вправо на 1 элемент результат записать в массив В. (Пример: А:3,4,5,6,7 В:7,3,4,5,6) Вывести на экран оба массива.
Вариант 9:	В массиве из 10 случайных чисел от -5 до 5 подсчитать количество отрицательных элементов. Вывести на экран значения элементов массива результат работы программы.
Вариант 10:	Массив А из 10 элементов заполнен случайными числами от 2 до 7. Вставить элемент со значением 0 после элемента с номером N. Вывести на экран исходный и получившийся массивы.
Вариант 11:	Массив А из 5 элементов заполнен случайными числами от 4 до 11. Осуществить циклический сдвиг элементов массива влево на 1 элемент результат записать в массив В. (Пример: А:3,4,5,6,7 В:4,5,6,7,3) Вывести на экран оба массива.
Вариант 12:	Найти среднее арифметическое элементов массива из 10 случайных чисел от 3 до 9. Вывести на экран значения элементов массива и результат работы программы.
Вариант 13:	Поменять местами значения соседних элементов массива из 10 элементов, заполненного случайными числами от 5 до 15. (Пример: было:5,6,7,8,9,10,11,12,13,14 стало:6,5,8,7,10,9,12,11,14,13) В программе использовать только 1 массив. Исходный и полученный массивы вывести на экран.
Вариант 14:	В массиве из 10 случайных чисел от -10 до 10 заменить все отрицательные элементы нулем. Исходный и полученный массивы вывести на экран.
Вариант	Программа генерирует N (N больше 0) случайных чисел в диапазоне от 0 до 9.

15:	Значения элементов массива А должны содержать количество выпавших значений равных индексу элемента массива. Исходные числа и результирующий массив вывести на экран.
Вариант 16:	Найти сумму четных элементов массива из 10 случайных чисел от 2 до 9. Вывести на экран значения элементов массива и результат работы программы.
Вариант 17:	Исходно все элементы массива А из 10 элементов равны 0. Вводимые с клавиатуры значения помещаем в элемент А[0]. А все уже имеющиеся значения сдвигаем вправо. Условие окончания ввода - ввод числа 0. (Пример: А:0,0,0; ввели 3 А:3,0,0; ввели 5 А: 5,3,0; ввели 2 А: 2,3,5; ввели 4 А: 4,2,3) После ввода каждого числа выводить массив.
Вариант 18:	Массив А из 10 элементов заполнен случайными числами от 3 до 10. Заменить все элементы массива с индексом большим К (К больше 0 и меньше 10) обратными значениями. (Пример: N=5 А:1,2,3,4,5 К=2 А:1,2,3,-4,-5) Вывести на экран исходный и получившийся массивы.
Вариант 19:	Найти сумму первых К (К больше 0 и меньше 10) элементов массива А из 10 элементов, заполненного случайными числами от -7 до 7. Вывести на экран значения элементов массива и результат работы программы.
Вариант 20:	Массив А из 10 элементов заполнен случайными числами от 1 до 15. Найти количество элементов массива, удовлетворяющих неравенству А[k] больше L (L больше или равно 1 и меньше 15) и меньше Н (Н больше L и меньше или равняется 15) Вывести на экран значения элементов массива результат работы программы.

Соответствие номеров по списку в журнале вариантам

Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Вариант	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	1	2	3	4	5	6	7	8	9	10

Практическая работа №12 "Сортировка одномерного массива"

Задания к практической №12 по вариантам

Вариант 1:	Массив из 15 целых чисел заполнить случайными числами от 0 до 9. Произвести сортировку массива. Вывести на экран исходный и отсортированный массивы. Сортировать по возрастанию прямым обменом
Вариант 2:	Массив из 10 целых чисел заполнить случайными числами от -5 до 5. Произвести сортировку массива. Вывести на экран исходный и отсортированный массивы. Сортировать по убыванию прямым выбором
Вариант 3:	Массив из 10 целых чисел заполнить случайными числами от 10 до 20. Произвести сортировку массива. Вывести на экран исходный и отсортированный массивы. Сортировать по убыванию прямым обменом
Вариант 4:	Массив из 10 целых чисел заполнить случайными числами от -5 до 9. Произвести сортировку массива. Вывести на экран исходный и отсортированный массивы. Сортировать по возрастанию прямым выбором

Соответствие номеров по списку в журнале вариантам																														
Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Вариант	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2

Практическая работа №13 "Составление программ на использование двумерных массивов"

Задания к практической №13 по вариантам	
Вариант 1:	Написать программу, которая запрашивает размерность квадратного двумерного массива, заполняет массив случайными числами от 2 до 6 и вычисляет сумму элементов главной диагонали.
Вариант 2:	Написать программу, которая запрашивает размерность двумерного массива, заполняет его случайными числами от 4 до 9 и вычисляет сумму элементов, сумма индексов у которых четная.
Вариант 3:	Написать программу, которая запрашивает размерность двумерного массива, заполняет его случайными числами от -6 до 6 и заменяет на 0 значения отрицательных элементов массива.
Вариант 4:	Написать программу, которая запрашивает размерность двумерного массива, заполняет его случайными числами от 3 до 7 и вычисляет сумму элементов в строках массива.
Вариант 5:	Написать программу, которая запрашивает размерность квадратного двумерного массива, заполняет массив случайными числами от 5 до 9 и вычисляет сумму элементов побочной диагонали.
Вариант 6:	Написать программу, которая запрашивает размерность двумерного массива, заполняет его случайными числами от 2 до 9 и вычисляет сумму элементов, сумма индексов у которых нечетная.
Вариант 7:	Написать программу, которая запрашивает размерность двумерного массива, заполняет его случайными числами от 5 до 11 и находит минимальный элемент в массиве.
Вариант 8:	Написать программу, которая запрашивает размерность двумерного массива, заполняет его случайными числами от 2 до 13 и находит индексы максимального элемента в массиве.
Вариант 9:	Написать программу, которая запрашивает размерность квадратного двумерного массива, заполняет массив случайными числами от 3 до 15 и заменяет на 0 значения элементов, лежащих выше главной диагонали.
Вариант 10:	Написать программу, которая запрашивает размерность квадратного двумерного массива, заполняет массив случайными числами от 1 до 5 и зеркально отражает ее элементы относительно главной диагонали.
Вариант 11:	Написать программу, которая запрашивает размерность двумерного массива, заполняет его случайными числами от 4 до 12 и меняет местами первую и последнюю строку массива.
Вариант 12:	Написать программу, которая запрашивает размерность двумерного массива, заполняет его случайными числами от 5 до 15 и находит максимальный элемент в массиве.

Вариант 13:	Написать программу, которая запрашивает размерность двумерного массива, заполняет его случайными числами от 2 до 13 и находит индексы минимального элемента в массиве.
Вариант 14:	Написать программу, которая запрашивает размерность двумерного массива, заполняет его случайными числами от 4 до 12 и меняет местами первый и последний столбец массива.
Вариант 15:	Написать программу, которая запрашивает размерность двумерного массива, заполняет его случайными числами от 3 до 9 и находит минимальные значения элементов в строках.
Вариант 16:	Написать программу, которая запрашивает размерность двумерного массива, заполняет его случайными числами от -5 до 5 и вычисляет сумму положительных элементов.
Вариант 17:	Написать программу, которая запрашивает размерность двумерного массива, заполняет его случайными числами от -10 до 10 и вывести номера строк, содержащих только положительные элементы.
Вариант 18:	Написать программу, которая запрашивает размерность двумерного массива, заполняет его случайными числами от 4 до 15 и находит максимальные значения элементов в столбцах.
Вариант 19:	Написать программу, которая запрашивает размерность двумерного массива, заполняет его случайными числами от 0 до 10 и вычисляет сумму четных элементов.
Вариант 20:	Написать программу, которая запрашивает размерность двумерного массива, заполняет его случайными числами от 5 до 9 и заменяет на 0 значения элементов массива с нечетной суммой индексов.

Соответствие номеров по списку в журнале вариантам

Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Вариант	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	1	2	3	4	5	6	7	8	9	10

Практическая работа №14 "Составление программ с использованием символьных переменных"

Задания к практической №14 по вариантам	
Вариант 1:	Написать программу, которая вводит символ с клавиатуры и выводит его код, до тех пор, пока не будет введен символ '0' (код символа 48).
Вариант 2:	Написать программу, которая вводит один символ с клавиатуры и выводит на экран таблицу 10 следующих за ним символов в формате «символ = код».
Вариант 3:	Написать программу, которая вводит один символ с клавиатуры и выводит на экран таблицу 10 предыдущих символов в формате «символ = код».
Вариант 4:	Написать программу, которая вводит один символ с клавиатуры и выводит сообщение, является ли введенный символ числом (Коды числовых символов – от 48 до 57).
Вариант 5:	Написать программу, которая запрашивает ввод 5 символов с клавиатуры. И определяет являются ли эти символы возрастающей последовательностью соответственно таблице кодировки ASCII.
Вариант 6:	Написать программу, которая вводит символы с клавиатуры и выводит на экран в формате «символ = код» только символы с четными кодами. Условие окончания ввода - ввод символа '0'
Вариант 7:	Написать программу, которая вводит символы с клавиатуры и выводит на экран сообщение является ли данный символ буквой, цифрой или другим символом. Условие окончания ввода - ввод символа '0'
Вариант 8:	Написать программу, которая запрашивает ввод символов с клавиатуры до тех пор, пока не введен символ '0'. По окончании ввода выводится наибольший введенный символ в формате «символ = код».
Вариант 9:	Написать программу, которая запрашивает ввод 5 символов с клавиатуры. И определяет являются ли эти символы идущими подряд в таблице кодировки ASCII.
Вариант 10:	Написать программу, которая выводит на экран таблицу символов с кодами от 48 до 127 в формате «символ = код».
Соответствие номеров по списку в журнале вариантам	
Номер по списку	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
Вариант	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Практическая работа №15 "Составление программ с использованием символьных строк"

Задания к практической №15 по вариантам																															
Вариант 1:	Написать программу, которая запрашивает текстовую строку, номера начального и конечного символа подстроки и формирует требуемую подстроку. При вводе номеров символов организовать проверку на правильность ввода.																														
Вариант 2:	С клавиатуры вводится строка символов, содержащая '!'. Определить сколько символов стоят после двоеточия.																														
Вариант 3:	Написать программу, которая случайным образом заполняет текстовую строку из 10 символов, выводит ее на экран и вычисляет процент цифровых символов в строке (использовать коды символов 33 - 127).																														
Вариант 4:	Написать программу, которая запрашивает строку и «переворачивает» её.																														
Вариант 5:	С клавиатуры вводится строка символов, содержащая '!'. Определить сколько символов ему предшествуют.																														
Вариант 6:	Написать программу, которая случайным образом заполняет текстовую строку из 20 символов (использовать коды символов 48 - 57), выводит ее на экран, запрашивает у пользователя символ и вычисляет сколько раз данный символ встретился в строке.																														
Вариант 7:	Написать программу, которая запрашивает 10 кодов символов, из введенных кодов формирует текстовую строку и выводит ее на экран (использовать коды символов 33 - 127).																														
Вариант 8:	С клавиатуры вводится строка символов, содержащая '!'. Заменить все символы '!' на '*'. Подсчитать количество замен.																														
Вариант 9:	Написать программу, которая случайным образом заполняет текстовую строку из 10 символов, выводит ее на экран и вычисляет процент буквенных символов в строке (использовать коды символов 33 - 127).																														
Вариант 10:	Проверить одинаковое ли число открывающих и закрывающих скобок в строке.																														
Соответствие номеров по списку в журнале вариантам																															
Номер по списку	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td> </tr> </table>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		
Вариант	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td> </tr> </table>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30		

Практическая работа №16 "Лингвистические задачи"

Задания к практической №16 по вариантам	
Во всех вариантах исходная строка состоит из нескольких слов, разделенных одним или несколькими пробелами.	
Вариант 1:	Во введенной строке удалить все лишние пробелы.
Вариант 2:	Найти размер самого длинного слова в строке.
Вариант 3:	Проверить содержит ли строка введенное слово.
Вариант 4:	Найти размер самого короткого слова в строке.
Вариант 5:	Удалить часть символьной строки, заключенной в скобки (вместе со скобками).
Вариант 6:	Строка содержит произвольный английский текст. Подсчитать какая буква сколько раз содержится в данной строке.
Вариант 7:	Дана строка. Подсчитать самую длинную последовательность идущих подряд букв 'a'.
Вариант 8:	Написать программу, шифрующую и дешифрующую введенную строку. Шифрование осуществляется увеличением кода символа на 1.
Вариант 9:	Подсчитать кол-во слов в строке.
Вариант 10:	Во введенной строке удалить повторение слов, стоящих рядом.

Соответствие номеров компьютеров вариантам

Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Вариант	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10

**Практическая работа №17 "Составление программ с использованием функций,
не возвращающих значения"**

Задания к практической №17 по вариантам	
Вариант 1:	Написать программу, выводящую сумму и разность двух введенных чисел. Основная программа запрашивает два числа. Вычисления и вывод на экран реализовать в виде функции.
Вариант 2:	Написать программу, выводящую на экран длину гипотенузы прямоугольного треугольника по двум катетам. Основная программа запрашивает длины катетов. Вычисление и вывод на экран реализовать в виде функции.
Вариант 3:	Написать программу, выводящую на экран таблицу квадратов целых чисел, расположенных между двумя числами. Основная программа запрашивает границы диапазона чисел. Вычисление и вывод на экран реализовать в виде функции.
Вариант 4:	Написать программу, выводящую на экран N символов. Основная программа запрашивает символ и количество повторений этого символа. Вывод на экран реализовать в виде функции.
Вариант 5:	Написать программу, вычисляющую X в степени Y. Основная программа запрашивает основание и показатель степени. Вычисление и вывод на экран реализовать в виде функции.
Вариант 6:	Написать программу, выводящую на экран результат сравнения двух чисел в виде: A=10 B=5: A>B. Основная программа запрашивает два числа. Результат сравнения и вывод на экран реализовать в виде функции.
Вариант 7:	Написать программу, выводящую на экран сообщение является ли введенный с клавиатуры символ десятичной цифрой, буквой или другим символом. Основная программа запрашивает символ. Вычисление и вывод на экран реализовать в виде функции.
Вариант 8:	Написать программу, выводящую на экран путь, пройденный пешеходом, если известны скорость пешехода и время движения. Основная программа запрашивает скорость и время. Вычисление и вывод на экран реализовать в виде функции.
Вариант 9:	Написать программу, выводящую на экран площадь прямоугольника, если известны длины его сторон. Основная программа запрашивает длины сторон. Вычисление и вывод на экран реализовать в виде функции.
Вариант 10:	Написать программу, выводящую на экран сопротивление участка электрической цепи, состоящей из двух сопротивлений. Основная программа запрашивает значения сопротивлений и тип соединения (1-последовательное, 2-параллельное). Вычисление и вывод на экран реализовать в виде функции.

Соответствие номеров по списку в журнале вариантам																														
Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Вариант	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10

**Практическая работа №18 "Составление программ с функциям,
возвращающими значения"**

Задания к практической №18 по вариантам	
Вариант 1:	Написать программу, выводящую на экран результат сравнения двух чисел в виде: $A=10$ $B=5$: $A>B$. Основная программа запрашивает два числа, вызывает функцию и в зависимости от возвращенного значения выводит сообщение. Функция возвращает 1 если A больше B , 0 если A равно B , -1 если A меньше B .
Вариант 2:	Написать программу, выводящую на экран сообщение является ли введенный с клавиатуры символ десятичной цифрой, буквой или другим символом. Основная программа запрашивает символ, вызывает функцию и в зависимости от возвращенного значения выводит сообщение. Функция возвращает 1 если символ числовой, 0 если буквенный и -1 для остальных символов.
Вариант 3:	Написать программу, выводящую сумму или разность двух введенных чисел. Основная программа запрашивает два числа и действие, вызывает функцию и выводит результат. Функция возвращает результат требуемого действия.
Вариант 4:	Написать программу, выводящую на экран длину гипотенузы прямоугольного треугольника по двум катетам. Основная программа запрашивает длины катетов, вызывает функцию и выводит результат вычислений или сообщение об ошибке в зависимости от возвращенного функцией значения. Функция проверяет корректность переданных ей данных, производит вычисление и возвращает либо результат либо -1 если введены некорректные данные.
Вариант 5:	Написать программу, выводящую на экран корень квадратный введенного числа, до тех пор, пока не будет введено число, корень квадратный которого равен самому числу. Основная программа запрашивает число, вызывает функцию, в зависимости от возвращенного значения выводит результат или сообщение об ошибке. Функция проверяет корректность введенных данных и возвращает либо вычисленное значение, либо -1 в случае ошибки.
Вариант 6:	Написать программу, выводящую на экран N символов. Основная программа запрашивает символ и количество повторений этого символа, вызывает функцию, по возвращенному результату выводит или не выводит сообщение об ошибке. Функция проверяет корректность введенных данных и возвращает -1 если данные введены не корректно или выводит символы и возвращает 0.
Вариант 7:	Написать программу, вычисляющую разрядность введенного целого числа. Основная программа запрашивает число, вызывает функцию и выводит результат или сообщение об ошибке. Функция определяет и возвращает разрядность числа. Если введены неверные значения функция возвращает -1.
Вариант 8:	Написать программу, выводящую на экран площадь прямоугольника, если известны длины его сторон. Основная программа запрашивает длины сторон, вызывает функцию и выводит результат или сообщение об ошибке. Функция определяет и возвращает площадь или -1, если введены неверные данные.
Вариант 9:	Написать программу, выводящую на экран сопротивление участка электрической цепи, состоящей из двух сопротивлений. Основная программа запрашивает значения сопротивлений и тип соединения (1-последовательное, 2-параллельное), вызывает функцию и выводит на экран результат или сообщение об ошибке. Функция определяет и возвращает значение сопротивления. Если введены неверные значения функция возвращает -1.
Вариант 10:	Написать программу, выводящую на экран путь, пройденный пешеходом, если известны скорость пешехода и время движения. Основная программа запрашивает

	скорость и время, вызывает функцию и выводит результат или сообщение об ошибке. Функция определяет и возвращает в основную программу пройденный путь. Если введены неверные значения функция возвращает -1.																													
Соответствие номеров по списку в журнале вариантам																														
Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Вариант	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30

Практическая работа №19 "Составление программ с использованием функций, передающих значения через глобальные переменные"

Задания к практической №19 по вариантам	
Вариант 1:	Написать программу, вычисляющую площадь и периметр прямоугольника по длинам его сторон. Основная программа запрашивает длины сторон и передает их в функцию по значению, выводит полученный результат. Функция производит вычисления и возвращает значения через глобальные переменные.
Вариант 2:	Написать программу, выводящую на экран сумму и среднее арифметическое N случайных чисел, заданных в диапазоне от A до B. Основная программа запрашивает значения N, A, B, передает их в функцию по значению, выводит полученный результат. Функция генерирует необходимое количество случайных чисел из заданного диапазона, выводит их на экран, производит вычисления и результат возвращает через глобальные переменные.
Вариант 3:	Написать программу, вычисляющую площадь круга и длину окружности по ее радиусу. Основная программа запрашивает радиус и передает его в функцию по значению, выводит полученный результат. Функция производит вычисления и возвращает значения через глобальные переменные.
Вариант 4:	Написать программу, выводящую на экран сумму и разность двух введенных чисел. Основная программа запрашивает два числа, передает их в функцию по значению, выводит результат вычислений. Функция производит вычисления и возвращает результат через глобальные переменные.
Вариант 5:	Написать программу, вычисляющую X в степени Y и Y в степени X. Основная программа запрашивает X и Y, передает их в функцию по значению, выводит полученный результат. Функция производит вычисления и возвращает результат через глобальные переменные.
Вариант 6:	Написать программу, вычисляющую скидку на покупку и сумму покупки со скидкой. Основная программа запрашивает сумму покупки и процент скидки, передает их в функцию по значению, выводит полученный результат. Функция вычисляет сумму скидки и сумму покупки со скидкой и возвращает их через глобальные переменные.
Вариант 7:	Написать программу, выводящую на экран потребляемую мощность и сопротивление участка электрической цепи, если известны ток и падение напряжения на данном участке. Основная программа запрашивает величину тока и падения напряжения, передает их в функцию по значению и выводит полученный результат. Функция вычисляет потребляемую мощность и сопротивление на участке цепи и возвращает полученные значения через глобальные переменные.

Соответствие номеров по списку в журнале вариантам																														
Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Вариант	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2

Практическая работа №20 "Составление программ с передачей массива в функцию"

Задания к практической №20 по вариантам	
Вариант 1:	Написать программу поиска минимального элемента массива. Поиск оформить в виде функции. Вывод массива на экран оформить в виде функции. Основная программа формирует массив из 10 элементов, заполненный случайными числами от 0 до 9, выводит его на экран через функцию, вызывает функцию поиска минимального элемента, выводит значение минимального элемента на экран. Передачу массива в функцию организовать по адресу.
Вариант 2:	Написать программу вычисления суммы элементов массива. Вычисление оформить в виде функции. Вывод массива на экран оформить в виде функции. Основная программа формирует массив из 10 элементов, заполненный случайными числами от 0 до 9, выводит его на экран через функцию, вызывает функцию вычисления суммы элементов массива, выводит значение суммы на экран. Передачу массива в функцию организовать по адресу.
Вариант 3:	Написать программу умножения двух массивов. Умножение производить поэлементно. Умножение оформить в виде функции. Вывод массива на экран оформить в виде функции. Основная программа формирует два массива из 10 элементов, заполненных случайными числами от 0 до 9, выводит их на экран через функцию, вызывает функцию умножения массивов, выводит полученный массив через функцию. Передачу массивов в функции организовать по адресу.
Вариант 4:	Написать программу вычисления среднего арифметического элементов массива. Вычисление оформить в виде функции. Вывод массива на экран оформить в виде функции. Основная программа формирует массив из 10 элементов, заполненный случайными числами от 0 до 9, выводит его на экран через функцию, вызывает функцию вычисления среднего арифметического элементов массива, выводит значение среднего арифметического на экран. Передачу массива в функцию организовать по адресу.
Вариант 5:	Написать программу поиска максимального элемента массива. Поиск оформить в виде функции. Вывод массива на экран оформить в виде функции. Основная программа формирует массив из 10 элементов, заполненный случайными числами от 0 до 9, выводит его на экран через функцию, вызывает функцию поиска максимального элемента, выводит значение максимального элемента на экран. Передачу массива в функцию организовать по адресу.
Вариант 6:	Написать программу вычисления количества и суммы положительных элементов массива. Вычисления оформить в виде функции. Вывод массива на экран оформить в виде функции. Основная программа формирует массив из 10 элементов, заполненный случайными числами от -5 до 5, выводит его на экран через функцию, вызывает функцию вычисления количества и суммы, выводит полученные значения на экран. Передачу массива в функцию организовать по адресу. Возвращать значения через глобальные переменные.

Вариант 7:	Написать программу, переворачивающую массив (исходный 12345 результат 54321). Действия оформить в виде функции. Вывод массива на экран оформить в виде функции. Основная программа формирует массив из 10 элементов, заполненный случайными числами от 0 до 9, выводит его на экран через функцию, вызывает функцию преобразования массива, выводит полученный массив через функцию. Передачу массива в функцию организовать по адресу.
Вариант 8:	Написать программу, вычитающую из каждого элемента массива минимальное значение элементов массива. Поиск минимального значения оформить в виде функции. Вычитание оформить в виде функции. Вывод массива на экран оформить в виде функции. Основная программа формирует два массива из 10 элементов, заполненных случайными числами от 0 до 9, выводит их на экран через функцию, вызывает функцию вычитания массивов, выводит полученный массив через функцию. Передачу массивов в функции организовать по адресу.
Вариант 9:	Написать программу вычитания двух массивов. Вычитание производить поэлементно. Вычитание оформить в виде функции. Вывод массива на экран оформить в виде функции. Основная программа формирует два массива из 10 элементов, заполненных случайными числами от 0 до 9, выводит их на экран через функцию, вызывает функцию вычитания массивов, выводит полученный массив через функцию. Передачу массивов в функции организовать по адресу.
Вариант 10:	Написать программу, сравнивающую два массива. Большим считать тот массив, сумма элементов которого больше. Подсчет суммы элементов организовать в виде функции. Сравнение организовать в виде функции. Вывод массива на экран оформить в виде функции. Основная программа формирует два массива из 10 элементов, заполненных случайными числами от 0 до 9, выводит их на экран через функцию, вызывает функцию сравнения массивов, выводит полученный результат на экран. Функция сравнения производит подсчет суммы элементов массива через функцию и возвращает -1 если первый массив меньше второго, 0 если массивы равны, 1 если первый массив больше второго. Передачу массивов в функции организовать по адресу.
Вариант 11:	Написать программу, сложения двух массивов. Сложение производить поэлементно. Сложение оформить в виде функции. Вывод массива на экран оформить в виде функции. Основная программа формирует два массива из 10 элементов, заполненных случайными числами от 0 до 9, выводит их на экран через функцию, вызывает функцию сложения массивов, выводит полученный массив через функцию. Передачу массивов в функции организовать по адресу.

Соответствие номеров по списку в журнале вариантам

Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Вариант	1	2	3	4	5	6	7	8	9	10	11	1	2	3	4	5	6	7	8	9	10	11	1	2	3	4	5	6	7	8

Практическая работа №21 "Составление программ с использованием массивов структур"

Задания к практической №21 по вариантам	
Вариант 1:	<p>Написать программу, работающую с массивом структур со следующими полями:</p> <ul style="list-style-type: none"> • Табельный номер работника; • Фамилия Имя Отчество работника; • Стаж работы. <p>Программа должна позволять ввести новые данные, вывести содержимое массива на экран, вывести средний стаж работы, выйти из программы. Все действия программы организовать с помощью функций.</p> <p>Рекомендации: Основная функция только реализует главное меню. Действие каждого пункта меню выполняет отдельная функция.</p>
Вариант 2:	<p>Написать программу, работающую с массивом структур со следующими полями:</p> <ul style="list-style-type: none"> • Номер поезда; • Маршрут следования; • Среднее опоздание за рейс. <p>Программа должна позволять ввести новые данные, вывести содержимое массива на экран, вывести информацию по поезду с максимальным опозданием, выйти из программы. Все действия программы организовать с помощью функций.</p> <p>Рекомендации: Основная функция только реализует главное меню. Действие каждого пункта меню выполняет отдельная функция.</p>
Вариант 3:	<p>Написать программу, работающую с массивом структур со следующими полями:</p> <ul style="list-style-type: none"> • Название группы; • Курс; • Количество студентов в группе. <p>Программа должна позволять ввести новые данные, вывести содержимое массива на экран, вывести информацию по группе с минимальным количеством студентов, выйти из программы. Все действия программы организовать с помощью функций.</p> <p>Рекомендации: Основная функция только реализует главное меню. Действие каждого пункта меню выполняет отдельная функция.</p>
Вариант 4:	<p>Написать программу, работающую с массивом структур со следующими полями:</p> <ul style="list-style-type: none"> • Название товара; • Цена товара; • Остаток товара на складе. <p>Программа должна позволять ввести новые данные, вывести содержимое массива на экран, вывести среднюю цену определенного товара, выйти из программы. Все действия программы организовать с помощью функций.</p> <p>Рекомендации: Основная функция только реализует главное меню. Действие каждого пункта меню выполняет отдельная функция.</p>
Вариант 5:	<p>Написать программу, работающую с массивом структур со следующими полями:</p> <ul style="list-style-type: none"> • Маршрут автомобиля; • Пробег за поездку (км); • Расход бензина за поездку (л). <p>Программа должна позволять ввести новые данные, вывести содержимое массива на экран, вывести средний расход бензина (л на 100 км) за все поездки, выйти из программы. Все действия программы организовать с помощью функций.</p> <p>Рекомендации: Основная функция только реализует главное меню. Действие каждого пункта меню выполняет отдельная функция.</p>

Соответствие номеров по списку в журнале вариантам																														
Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Вариант	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5

Практическая работа №22 "Запись в файл и чтение из файла однотипных данных"

Задания к практической №22 по вариантам	
Вариант 1:	Написать две программы: Первая вводит с клавиатуры целые числа до тех пор, пока не введено число 0 и записывает их в файл. Вторая производит чтение из файла и выводит на экран считанные числа и их сумму. Использовать функции записи-чтения целых чисел. Файл сохранять в каталоге Student. Чтение из файла производить до конца файла.
Вариант 2:	Написать две программы: Первая вводит с клавиатуры N целых чисел и записывает их в файл. Вторая производит чтение из файла и выводит на экран считанные числа и минимальное из этих чисел. Использовать функции записи-чтения целых чисел. Файл сохранять в каталоге Student. Чтение из файла производить до конца файла.
Вариант 3:	Написать две программы: Первая вводит с клавиатуры целые числа до тех пор, пока не введено число 0 и записывает их в файл. Вторая производит чтение из файла и выводит на экран считанные числа и их среднее арифметическое. Использовать функции записи-чтения целых чисел. Файл сохранять в каталоге Student. Чтение из файла производить до конца файла.
Вариант 4:	Написать две программы: Первая вводит с клавиатуры N целых чисел и записывает их в файл. Вторая производит чтение из файла и выводит на экран считанные числа и максимальное из этих чисел. Использовать функции записи-чтения целых чисел. Файл сохранять в каталоге Student. Чтение из файла производить до конца файла.
Вариант 5:	Написать две программы: Первая случайным образом генерирует N целых чисел и записывает их в файл. Вторая производит чтение из файла и выводит на экран считанные числа и квадраты этих чисел. Использовать функции записи-чтения целых чисел. Файл сохранять в каталоге Student. Чтение из файла производить до конца файла.
Вариант 6:	Написать две программы: Первая вводит с клавиатуры символы до тех пор, пока не введен символ '0' и записывает их в файл. Вторая производит чтение из файла, формирует строку и выводит ее на экран. Использовать функции записи-чтения символов. Файл сохранять в каталоге Student. Чтение из файла производить до конца файла.

Соответствие номеров по списку в журнале вариантам																														
Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Вариант	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6

Практическая работа №23 "Запись в файл и чтение из файла блоков данных"

Задания к практической №23 по вариантам	
Вариант 1:	Написать две программы: Первая запрашивает заполнение вводом с клавиатуры структуры данных и выводит структуру данных в файл. Вторая производит чтение из файла и выводит данные на экран. Структура данных: Массив структур из 5 элементов. Структура имеет три поля: целое число, действительное число, строка. Использовать функции записи-чтения блоков данных. Файл сохранять в каталоге Student.
Вариант 2:	Написать две программы: Первая инициализирует структуру данных и выводит структуру данных в файл. Вторая производит чтение из файла и выводит данные на экран. Структура данных: Массив структур из 5 элементов. Структура имеет два поля: Целое число, строка. Использовать функции записи-чтения блоков данных. Файл сохранять в каталоге Student.
Вариант 3:	Написать две программы: Первая запрашивает заполнение вводом с клавиатуры структуры данных и выводит структуру данных в файл. Вторая производит чтение из файла и выводит данные на экран. Структура данных: Массив строк из 5 элементов. Использовать функции записи-чтения блоков данных. Файл сохранять в каталоге Student.
Вариант 4:	Написать две программы: Первая запрашивает заполнение вводом с клавиатуры структуры данных и выводит структуру данных в файл. Вторая производит чтение из файла и выводит данные на экран. Структура данных: Двумерный массив целых чисел из 3 строк и 3 столбцов. Использовать функции записи-чтения блоков данных. Файл сохранять в каталоге Student.
Вариант 5:	Написать две программы: Первая инициализирует структуру данных и выводит структуру данных в файл. Вторая производит чтение из файла и выводит данные на экран. Структура данных: Массив структур из 5 элементов. Структура имеет два поля: Целое число, массив действительных чисел из 2 элементов. Использовать функции записи-чтения блоков данных. Файл сохранять в каталоге Student.
Вариант 6:	Написать две программы: Первая заполняет вводом с клавиатуры структуру данных и выводит структуру данных в файл. Вторая производит чтение из файла и выводит данные на экран. Структура данных: Массив структур из 3 элементов. Структура имеет два поля: строка, массив действительных чисел из 2 элементов. Использовать функции записи-чтения блоков данных. Файл сохранять в каталоге Student.

Соответствие номеров по списку в журнале вариантам																									
Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Вариант	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1

**Практическая работа №24 "Составление программ с использованием
форматного обмена с файлами" (4 ч.)**

Задания к практической №24 по вариантам	
Вариант 1:	<p>Написать программу, вычисляющую среднее арифметическое 5 целых чисел. Исходные данные должны находиться в файле input.txt, результат работы записан в файл output.txt.</p> <p>Формат файла input.txt: 1 2 3 4 5</p> <p>Формат файла output.txt: 3.0</p>
Вариант 2:	<p>Написать программу, выводящую в текстовый файл таблицу квадратов всех целых чисел, расположенных между А и В. Исходные данные должны находиться в файле input.txt, результат работы записан в файл output.txt.</p> <p>Формат файла input.txt: 2 6</p> <p>Формат файла output.txt: 2 4 3 9 4 16 5 25 6 36</p>
Вариант 3:	<p>Написать программу, сортировки одномерного массива 10 действительных чисел. Исходные данные должны находиться в файле input.txt, результат работы записан в файл output.txt.</p> <p>Формат файла input.txt: 1.1 5.4 2.3 4.2 3.5</p> <p>Формат файла output.txt: 1.1 2.3 3.5 4.2 5.4</p>
Вариант 4:	<p>Написать программу, вычисляющую количество ненулевых элементов в двумерном массиве. Размер массива и исходные данные должны находиться в файле input.txt, результат работы записан в файл output.txt.</p> <p>Формат файла input.txt: 3 4 1 3 6 3 3 0 2 4 4 2 0 0</p> <p>Формат файла output.txt: 9</p>
Вариант 5:	<p>Написать программу, вычисляющую сумму двух массивов. Размер массива и исходные данные должны находиться в файле input.txt, результат работы записан в файл output.txt.</p> <p>Формат файла input.txt: 3 3 1 3 2 3 0 2 4 2 0 3 5 4 4 1 3 2 2 1</p> <p>Формат файла output.txt:</p>

	4 8 6
	7 1 5
	6 4 1

Соответствие номеров по списку в журнале вариантам																														
Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Вариант	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5

Практическая работа №25 "Составление программ с использованием указателей" (4 ч.)

Задания к практической №25 по вариантам

Задание: Изменить приведенный листинг, используя по всей программе указатели на все описанные переменные. Запустить и отладить программу. В отчете указать, что делает данная программа. Нарисовать блок-схему.

<p>Вариант 1:</p>	<pre>#include <stdio.h> #include <stdlib.h> #include <conio.h> void main() { int a[10],s=0; int i; randomize(); clrscr(); for(i=0;i<5;i=i+1) { a[i]=rand()%10; s=s+a[i]; printf("%i ",a[i]); } printf("\ns=%i",s); getch(); }</pre>	<p>Вариант 2:</p>	<pre>#include <stdio.h> #include <stdlib.h> #include <conio.h> void main() { int a[10],s; int i; randomize(); clrscr(); a[0]=rand()%10; printf("%i ",a[0]); s=a[0]; for(i=1;i<5;i=i+1) { a[i]=rand()%10; if(s>a[i]) s=a[i]; printf("%i ",a[i]); } printf("\ns=%i",s); getch(); }</pre>
<p>Вариант 3:</p>	<pre>#include <stdio.h> #include <stdlib.h> #include <conio.h> void main() { int a[10]; float s=0; int i; randomize(); clrscr(); for(i=0;i<5;i=i+1) { a[i]=rand()%10; s=s+a[i]; printf("%i ",a[i]); } printf("\ns=%.2f",s/5); getch(); }</pre>	<p>Вариант 4:</p>	<pre>#include <stdio.h> #include <stdlib.h> #include <conio.h> void main() { int a[10],s; int i; randomize(); clrscr(); a[0]=rand()%10; printf("%i ",a[0]); s=a[0]; for(i=1;i<5;i=i+1) { a[i]=rand()%10; if(s<a[i]) s=a[i]; printf("%i ",a[i]); } printf("\ns=%i",s); getch(); }</pre>

Соответствие номеров по списку в журнале вариантам																														
Номер по списку	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Вариант	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2

Критерии оценки:

- оценка «зачтено» выставляется студенту, если правильно выполнены все задания лабораторной работы;
- оценка «не зачтено» выставляется студенту, если неправильно выполнены некоторые задания или выполнены не все задания лабораторной работы.

**Комплект заданий для выполнения
контрольных работ**

Контрольная работа №1

по теме "Представление данных в памяти ЭВМ"

Вариант 1

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

IBM PC

2. Дешифровать данный текст, используя таблицу ASCII-кодов:

8A AE AC AF EC EE E2 A5 E0

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) $236_{(10)}$; б) $195_{(10)}$; в) $161_{(10)}$.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) $25_{(10)}$; б) $-111_{(10)}$; в) $-66_{(10)}$.

3а. Записать прямой код числа, интерпретируя его как шестнадцатибитовое целое без знака:

а) $24411_{(10)}$; б) $18612_{(10)}$.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) $29187_{(10)}$; б) $-19433_{(10)}$.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0001111010101101 ; б) 1001110110011100 .

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) $870,15625$; б) $-250,15625$.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) $C062A50000000000$; б) $C08A6C6000000000$.

Вариант 2

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

Автоматизация

2. Дешифровать данный текст, используя таблицу ASCII-кодов:

50 72 6F 67 72 61 6D

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) $246_{(10)}$; б) $172_{(10)}$; в) $145_{(10)}$.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) $9_{(10)}$; б) $-42_{(10)}$; в) $-77_{(10)}$.

3а. Записать прямой код числа, интерпретируя его как

шестнадцатибитовое целое без знака:

а) 25245₍₁₀₎; б) 24290₍₁₀₎.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) 27554₍₁₀₎; б) -17709₍₁₀₎.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 000111110111001; б) 1110110001001111.

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) -289,375; б) -374,15625.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) C072760000000000; б) C07D328000000000.

Вариант 3

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

Информатика

2. Дешифровать данный текст, используя таблицу ASCII-кодов:

50 72 6F 63 65 64 75 72 65

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) 191₍₁₀₎; б) 235₍₁₀₎; в) 201₍₁₀₎.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) 61₍₁₀₎; б) -36₍₁₀₎; в) -71₍₁₀₎.

За. Записать прямой код числа, интерпретируя его как шестнадцатибитовое целое без знака:

а) 30426₍₁₀₎; б) 25175₍₁₀₎.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) 31763₍₁₀₎; б) -25694₍₁₀₎.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0100101000110101; б) 1101010101010110.

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) 244,375; б) 452,15625.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) C074D28000000000; б) C080AB0000000000.

Вариант 4

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

Computer

2. Дешифровать данный текст, используя таблицу ASCII-кодов:

84 88 91 8A 8E 82 8E 84

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) $151_{(10)}$; б) $205_{(10)}$; в) $163_{(10)}$.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) $113_{(10)}$; б) $-118_{(10)}$; в) $-27_{(10)}$.

3а. Записать прямой код числа, интерпретируя его как шестнадцатибитовое целое без знака:

а) $23386_{(10)}$; б) $30977_{(10)}$.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) $19890_{(10)}$; б) $-17862_{(10)}$.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0010110010010011; б) 1100001111111111.

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) $-871,375$; б) $303,15625$.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) 40820B0000000000; б) C05DCA0000000000.

Вариант 5

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

Printer

2. Дешифровать данный текст, используя таблицу ASCII-кодов:

43 4F 4D 50 55 54 45 52

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) $162_{(10)}$; б) $169_{(10)}$; в) $216_{(10)}$.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) $57_{(10)}$; б) $-59_{(10)}$; в) $-89_{(10)}$.

3а. Записать прямой код числа, интерпретируя его как шестнадцатибитовое целое без знака:

а) $19802_{(10)}$; б) $18657_{(10)}$.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) $22142_{(10)}$; б) $-28086_{(10)}$.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0111011101000000; б) 1000100110011001.

6. 1) Записать код действительного числа, интерпретируя его как

величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) 823,375; б) -829,375.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) C06C918000000000; б) C06F918000000000.

Вариант 6

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

компьютеризация

2. Дешифровать данный текст, используя таблицу ASCII-кодов:

50 52 49 4E 54

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) 148₍₁₀₎; б) 161₍₁₀₎; в) 197₍₁₀₎.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) 57₍₁₀₎; б) -42₍₁₀₎; в) -14₍₁₀₎.

За. Записать прямой код числа, интерпретируя его как шестнадцатибитовое целое без знака:

а) 27606₍₁₀₎; б) 20426₍₁₀₎.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) 29564₍₁₀₎; б) -25136₍₁₀₎.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0011110010101111; б) 1001110101110100.

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) -765,5; б) 507,15625.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) 408B7B0000000000; б) C079F80000000000.

Вариант 7

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

УАМАНА

2. Дешифровать данный текст, используя таблицу ASCII-кодов:

4D 4F 44 45 4D

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) 245₍₁₀₎; б) 223₍₁₀₎; в) 168₍₁₀₎.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) 116₍₁₀₎; б) -113₍₁₀₎; в) -86₍₁₀₎.

За. Записать прямой код числа, интерпретируя его как

шестнадцатибитовое целое без знака:

а) 22976₍₁₀₎; б) 19745₍₁₀₎.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) 18933₍₁₀₎; б) -20578₍₁₀₎.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0011000101001110; б) 1101110011011111.

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) 883,375; б) 893,15625.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) C02E500000000000; б) 4045B00000000000.

Вариант 8

1. Зашифровать данный текст, используя таблицу ASCII-кодов:
световое перо

2. Дешифровать данный текст, используя таблицу ASCII-кодов:
4C 61 73 65 72

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) 222₍₁₀₎; б) 216₍₁₀₎; в) 226₍₁₀₎.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) 21₍₁₀₎; б) -125₍₁₀₎; в) -79₍₁₀₎.

За. Записать прямой код числа, интерпретируя его как шестнадцатибитовое целое без знака:

а) 22342₍₁₀₎; б) 23325₍₁₀₎.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) 27462₍₁₀₎; б) -27276₍₁₀₎.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0110101110001100; б) 1100101100101110.

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) -373,375; б) -122,546875.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) C07FE80000000000; б) C039600000000000.

Вариант 9

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

Микропроцессор

2. Дешифровать данный текст, используя таблицу ASCII-кодов:
88 AD E4 AE E0 AC A0 E2 A8 AA A0

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) $185_{(10)}$; б) $141_{(10)}$; в) $182_{(10)}$.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) $8_{(10)}$; б) $-37_{(10)}$; в) $-54_{(10)}$.

3а. Записать прямой код числа, интерпретируя его как шестнадцатибитовое целое без знака:

а) $26385_{(10)}$; б) $21127_{(10)}$.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) $28297_{(10)}$; б) $-17890_{(10)}$.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0010001111011101; б) 1101000110101101.

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) $-952,5$; б) $-405,546875$.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) C0602C0000000000; б) C069318000000000.

Вариант 10

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

Принтер

2. Дешифровать данный текст, используя таблицу ASCII-кодов:
42 69 6E 61 72 79

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) $131_{(10)}$; б) $198_{(10)}$; в) $135_{(10)}$.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) $95_{(10)}$; б) $-100_{(10)}$; в) $-48_{(10)}$.

3а. Записать прямой код числа, интерпретируя его как шестнадцатибитовое целое без знака:

а) $28033_{(10)}$; б) $26362_{(10)}$.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) $21818_{(10)}$; б) $-16177_{(10)}$.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0111010010101101; б) 111111110101110.

6. 1) Записать код действительного числа, интерпретируя его как

величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) -347,546875; б) -535,375.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) C03C800000000000; б) C08B114000000000.

Вариант 11

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

Дисковод

2. Дешифровать данный текст, используя таблицу ASCII-кодов:

49 6E 66 6F 72 6D 61 74 69 6F 6E

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) $139_{(10)}$; б) $214_{(10)}$; в) $156_{(10)}$.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) $20_{(10)}$; б) $-66_{(10)}$; в) $-18_{(10)}$.

За. Записать прямой код числа, интерпретируя его как шестнадцатибитовое целое без знака:

а) $20467_{(10)}$; б) $19491_{(10)}$.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) $27343_{(10)}$; б) $-22291_{(10)}$.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0001101001101111; б) 1100101010000001.

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) -807,15625; б) -378,15625.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) C06FC50000000000; б) C070C60000000000.

Вариант 12

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

Pentium 100

2. Дешифровать данный текст, используя таблицу ASCII-кодов:

91 A8 E1 E2 A5 AC A0 20 E1 E7 A8 E1 AB A5 AD A8 EF

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) $178_{(10)}$; б) $218_{(10)}$; в) $222_{(10)}$.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) $33_{(10)}$; б) $-97_{(10)}$; в) $-20_{(10)}$.

За. Записать прямой код числа, интерпретируя его как

шестнадцатибитовое целое без знака:

а) $21115_{(10)}$; б) $20225_{(10)}$.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) $25019_{(10)}$; б) $-21083_{(10)}$.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0011001110010000; б) 1100110101110101.

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) -590,5; б) 383,375.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) C06E4C0000000000; б) C05A980000000000.

Вариант 13

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

Арифмометр

2. Дешифровать данный текст, используя таблицу ASCII-кодов:

АС АЕ А4 А5 АВ А8 Е0 АЕ А2 А0 АД А8 А5

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) $172_{(10)}$; б) $250_{(10)}$; в) $150_{(10)}$.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) $103_{(10)}$; б) $-101_{(10)}$; в) $-32_{(10)}$.

3а. Записать прямой код числа, интерпретируя его как шестнадцатибитовое целое без знака:

а) $31659_{(10)}$; б) $31524_{(10)}$.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) $30532_{(10)}$; б) $-23601_{(10)}$.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0000000010000010; б) 1000100111011100.

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) -475,375; б) 73,15625.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) C074260000000000; б) C042460000000000.

Вариант 14

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

Сканер

2. Дешифровать данный текст, используя таблицу ASCII-кодов:
**A2 EB E7 A8 E1 AB A8 E2 A5 AB EC AD EB A9 20 ED AA E1 AF A5 E0 A8
AC A5 AD E2**

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) $189_{(10)}$; б) $167_{(10)}$; в) $178_{(10)}$.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) $105_{(10)}$; б) $-31_{(10)}$; в) $-6_{(10)}$.

3а. Записать прямой код числа, интерпретируя его как шестнадцатибитовое целое без знака:

а) $30027_{(10)}$; б) $31599_{(10)}$.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) $20399_{(10)}$; б) $-25589_{(10)}$.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0010001110110100; б) 1010011100000010.

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) 254,375; б) 552,546875.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) 406A6C0000000000; б) 40648C0000000000.

Вариант 15

1. Зашифровать данный текст, используя таблицу ASCII-кодов:
ВИНЧЕСТЕР

2. Дешифровать данный текст, используя таблицу ASCII-кодов:
43 6F 6D 70 75 74 65 72 20 49 42 4D 20 50 43

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) $251_{(10)}$; б) $220_{(10)}$; в) $128_{(10)}$.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) $119_{(10)}$; б) $-73_{(10)}$; в) $-60_{(10)}$.

3а. Записать прямой код числа, интерпретируя его как шестнадцатибитовое целое без знака:

а) $26399_{(10)}$; б) $29913_{(10)}$.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) $23274_{(10)}$; б) $-32389_{(10)}$.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0110000110010110; б) 1011001000111011.

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) -235,15625; б) -1002,546875.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) C080330000000000; б) 4081330000000000.

Вариант 16

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

IBM PC

2. Дешифровать данный текст, используя таблицу ASCII-кодов:

8A AE AC AF EC EE E2 A5 E0

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) $145_{(10)}$; б) $223_{(10)}$; в) $241_{(10)}$.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) $6_{(10)}$; б) $-93_{(10)}$; в) $-45_{(10)}$.

За. Записать прямой код числа, интерпретируя его как шестнадцатибитовое целое без знака:

а) $23205_{(10)}$; б) $27175_{(10)}$.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) $28644_{(10)}$; б) $-17439_{(10)}$.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0110000010011010; б) 1101111011010100.

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) 741,375; б) 180,15625.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) 406E4C0000000000; б) 4086194000000000.

Вариант 17

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

Автоматизация

2. Дешифровать данный текст, используя таблицу ASCII-кодов:

50 72 6F 67 72 61 6D

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) $248_{(10)}$; б) $253_{(10)}$; в) $214_{(10)}$.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) $79_{(10)}$; б) $-66_{(10)}$; в) $-96_{(10)}$.

3а. Записать прямой код числа, интерпретируя его как шестнадцатибитовое целое без знака:

а) $23125_{(10)}$; б) $23346_{(10)}$.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) $28330_{(10)}$; б) $-30175_{(10)}$.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0110100100101101; б) 1100011000100000.

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) 612,15625; б) -926,375.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) C071360000000000; б) 4059580000000000.

Вариант 18

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

Информатика

2. Дешифровать данный текст, используя таблицу ASCII-кодов:

50 72 6F 63 65 64 75 72 65

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) $191_{(10)}$; б) $210_{(10)}$; в) $164_{(10)}$.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) $67_{(10)}$; б) $-14_{(10)}$; в) $-84_{(10)}$.

3а. Записать прямой код числа, интерпретируя его как шестнадцатибитовое целое без знака:

а) $21087_{(10)}$; б) $25637_{(10)}$.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) $24788_{(10)}$; б) $-21728_{(10)}$.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0010010111000011; б) 1010011101101110.

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) -704,375; б) -653,375.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) 408C014000000000; б) C0846C0000000000.

Вариант 19

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

Computer

2. Дешифровать данный текст, используя таблицу ASCII-кодов:

84 88 91 8A 8E 82 8E 84

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) $194_{(10)}$; б) $255_{(10)}$; в) $178_{(10)}$.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) $28_{(10)}$; б) $-70_{(10)}$; в) $-41_{(10)}$.

3а. Записать прямой код числа, интерпретируя его как шестнадцатибитовое целое без знака:

а) $28273_{(10)}$; б) $24510_{(10)}$.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) $22809_{(10)}$; б) $-19228_{(10)}$.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0110001011000111; б) 1010100000001100.

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) 386,375; б) 857,375.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) 4071428000000000; б) C06D2C0000000000.

Вариант 20

1. Зашифровать данный текст, используя таблицу ASCII-кодов:

Printer

2. Дешифровать данный текст, используя таблицу ASCII-кодов:

43 4F 4D 50 55 54 45 52

3. Записать прямой код числа, интерпретируя его как восьмибитовое целое без знака:

а) $245_{(10)}$; б) $242_{(10)}$; в) $248_{(10)}$.

4. Записать дополнительный код числа, интерпретируя его как восьмибитовое целое со знаком:

а) $43_{(10)}$; б) $-47_{(10)}$; в) $-102_{(10)}$.

3а. Записать прямой код числа, интерпретируя его как шестнадцатибитовое целое без знака:

а) $16111_{(10)}$; б) $17675_{(10)}$.

4а. Записать дополнительный код числа, интерпретируя его как шестнадцатибитовое целое со знаком:

а) $21217_{(10)}$; б) $-17901_{(10)}$.

5. Записать в десятичной системе счисления целое число, если дан его дополнительный код:

а) 0100111010101100; б) 1001001101001101.

6. 1) Записать код действительного числа, интерпретируя его как величину типа Double, результат закодировать в шестнадцатеричной системе счисления:

а) 44,15625; б) 608,15625.

2) Дан код величины типа Double, записанный шестнадцатеричными цифрами. Преобразовать его в число:

а) 408E9C0000000000; б) C065318000000000.

Пример варианта контрольной работы № 2

1. Составить линейный алгоритм: расстояние между автомобилями — S км. Определить расстояние между ними через t ч, если первый движется со скоростью v_1 , а второй — v_2 км/ч. Примечание. Рассмотреть случаи движения навстречу друг другу и в противоположные стороны друг от друга.

2. Составить алгоритм с развилкой: выяснить, что больше: площадь правильного шестиугольника со стороной a или прямоугольника с измерениями b, c .

3. Составить алгоритм с использованием цикла: найти сумму членов ряда, для которых выполняется неравенство $|a_i| \geq \square$:

$$a_i = \sum_{i=0}^n (-1)^i \frac{x^{2i+1}}{(2i+1)!}$$

4. Составить алгоритм решения задачи целочисленной арифметики: вычеркнуть из записи данного натурального числа N первую цифру.

5. Составить алгоритм решения задачи с использованием вспомогательных алгоритмов: поменять порядок следования цифр в заданном натуральном числе N на обратный, удалив при этом из записи получаемого числа все вхождения цифры k .

Пример варианта контрольной работы № 3

1. Изобразить на плоскости (X, Y) область, в которой и только в которой истинно указанное выражение

$(X \leq 0) \text{ and } (\text{Sqr}(X) + \text{Sqr}(Y) \leq 4) \text{ or } (X \geq 0) \text{ and } (Y \leq -X + 2) \text{ and } (Y \geq X - 2)$

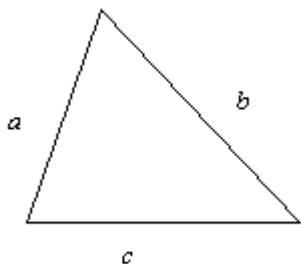
2. Объяснить, в чем заключается синтаксическая ошибка (или ошибки) в приведенной программе. Написать тот вариант программы, который, по Вашему мнению, будет правильным.

```
Program My_Error;
Var X, Y, Сумма: Integer;
Begin
    WriteLn( Введите значение X); ReadLn(X);
    WriteLn( Введите значение Y); ReadLn(Y);
    Сумма = X + Y;
    WriteLn(Сумма)
```

End.

3. Составить программу идентификации треугольника по сторонам a, b, c .

Определяемое свойство: является прямоугольным или не является прямоугольным (удобно использовать сравнение квадрата одной из сторон с суммой квадратов двух других сторон).



4. Решить задачу с использованием оператора выбора: по введенному числу грибов k напечатать фразу “Мы нашли в лесу k грибов” и согласовать при этом окончание слова “гриб” с числом k . (Количество грибов может быть любым целым числом: 1, 3, 34, 127 и т.п. Окончание фразы определяется значением последней цифры.)

5. Точно и четко сформулировать условие задачи, которая решается в данной программе:

```
Program Pr;  
Var K, I, D, S: Integer;  
Begin WriteLn('Введите значение K: '); ReadLn(K);  
  For I := 100 To 300 Do  
    Begin D := 2;  
      S := 0;  
      While D <= I div 2 + 1 Do  
        Begin If I mod D = 0 Then S := S + D;  
              D := D + 1  
        End;  
      If S = K Then WriteLn(I);  
    End  
  End  
End.
```

6. Решить задачу с использованием цикла: найти k -е простое число в арифметической прогрессии 11, 21, 31, 41, 51, 61, ...

7. Определить, что будет напечатано программой по ходу ее выполнения:

```
Program Schet;  
Var M, N: Word;  
Begin M := 0; N := 0;  
  Repeat M := M + 1;  
        N := N + M;  
        Write(M:5, N:3)  
  Until N mod 7 = 0;  
  Writeln;  
  M := 0; N := 0;  
  While M + N < 20 Do  
    Begin M := M + 1; N := N + M; Write(M:5, N:3) End  
End.
```

8. Решить задачу с использованием подпрограммы: вывести наибольшую из первых цифр трех заданных чисел. Например, если $a = 25$, $b = 730$, $c = 1995$, то надо вывести цифру 7.

9. В данной последовательности a_1, a_2, \dots, a_n определить максимальное число среди элементов с номерами, кратными числу k . Например, для последовательности $-1, 0, 12, -77, 22, -6, 70, 11, 3$ и $k = 3$ получаем ответ 12.

10. Вычислить

$$\sqrt{2 + \sqrt{2 + \dots + \sqrt{2}}}$$

(n слагаемых).

Контрольная работа № 4. Массивы, строки

Примерный вариант

1. Определить, какой объем оперативной памяти требуется для хранения каждой из объявленных переменных. Указать, какую из переменных не удастся разместить в памяти из-за превышения допустимого объема для статических переменных (возникнет ошибка Structure too large).

- Var A : Array[0..99] Of LongInt;
- Var B : Array[Integer] Of Real;
- Var C : Array[-100..100] Of Real;
- Var D : Array['a'..'z'] Of Extended;
- Var E : Array[1..100, 0..39] Of ShortInt;

2. Точно и четко сформулировать условие задачи, которая решается в данной программе:

```
Program Kr_N_4;
Const NMax = 50; Type Mass = Array[0..NMax] Of Real;
Var X : Mass; I, N : Byte; Vsp : Real;
Begin Write('Сколько элементов в массиве? '); ReadLn(N); N := N - 1;
      Randomize;
      For I := 0 To N Do
      Begin X[i] := -50 + Random * 100.1; Write(X[i]:10:5) End;
      WriteLn; I := 0;
      While I < N Do
      Begin Vsp := X[i]; X[i] := X[i + 1];
           X[i + 1] := Vsp; I := I + 2
      End;
      For I := 0 To N Do Write(X[i]:10:5); WriteLn;
End.
```

3. Решить задачу: подсчитать количество элементов массива, каждый из которых не меньше всех своих предшественников (т.е. элементов с меньшими индексами). Например, для массива $-2, 3, 0, 13, 13, 4, -3$ ответ будет 4.

4. *Решить задачу:* заполнить квадратную матрицу порядка N по следующему правилу: элементы выше главной диагонали совпадают с разностью первого и второго индексов, выше побочной диагонали — с произведением индексов. Примечание. К элементам, попадающим в пересечение областей, применить следующее правило: они получаются, как сумма значений, полученных по первому и второму правилу. Все остальные элементы матрицы равны нулю.

5. *Решить задачу:* в целочисленной прямоугольной таблице размером $M \times N$ определить наибольшее число, кратное заданному K , или сообщить об отсутствии таких чисел.

6. Точно и четко сформулировать условие задачи, которая решается в данной программе:

```

Program Kr_4_z2;
Var S : String; M, I, N, J : Byte; Vsp : Char;
Begin
    Write('Введите строку: '); ReadLn(S); N := Length(S);
    For I := 1 To N - 1 Do
        Begin
            m := I;
            For J := I + 1 To N Do If S[j] < S[m] Then m := j;
            Vsp := S[i]; S[i] := S[m]; S[m] := Vsp
        End;
    WriteLn('Ответ: ', S)
End.

```

6. *Решить задачу:* в строке заменить указанный символ C на символ $C1$. Подсчитать число замен.

Контрольная работа № 5 (рекурсия, множества, записи)

Примерный вариант

1. Определить, какой минимальный объём оперативной памяти требуется для хранения каждой из объявленных переменных.

a) Type CompCh = Record re, im: Real end;

Var Spis : array[1..100,0..39] of CompCh;

b) Type Kol = Record name : string[10]; fone : LongInt end;

Var T: array [-1..10] of Kol;

c) Var Uch: Record fam, name: string[15]; year : Word end;

d) Var M: set of '0'..'9';

2. Описать рекурсивную функцию $pow(x, n)$ от вещественного x ($x \neq 0$) и целого n , которая вычисляет величину x^n согласно формуле

$$x^n = \begin{cases} 1, & x = 0; \\ \frac{1}{x^{|n|}}, & x < 0; \\ x \cdot x^{n-1}, & x > 0. \end{cases}$$

3. Вычислить значения выражений

a) [2,3,5,7] <= [1..9]; b) ['c', 'b'] = ['b'..'c'];

c) `trunc(3.9)` in `[1,3,5]`; d) `[2..13]*[3,13..60]+[4..10]-[5..15]*[6]`;
e) `[2..10]-[4,6]-[2..12]*[8..15]`; f) `(['0'..'7']+['2'..'9'])*(['a']+['z'])`.

4. Решить задачу с использованием множества: имеется строка, содержащая русский текст. Вывести те буквы, которые являются общими для первых трёх предложений текста.

5. Заданы некоторые типы данных и объявлены переменные:

```
Type Complex = Record re, im: Real end;
```

```
Point = Record x, y: Real end;
```

```
Var z, w: Complex; p: Point; re: Real;
```

Определить, какие значения будут иметь переменные `z.re`, `z.im`, `w.re`, `w.im`, `p.x`, `p.y`, `re` после выполнения следующих операторов:

```
with z do begin re := 0; im := 1 end;
```

```
w := z; re := 2;
```

```
with z do re := 1;
```

```
with z, w do im := -im;
```

```
with p do begin x := re; y := 2 end;
```

6. Сведения о результатах сессии студентов второго курса (фамилия, номер группы и оценки по трем экзаменам) хранятся в массиве:

```
Type Stud = Record
```

```
    fam : string[12];
```

```
    num : 121..126;
```

```
    otm1, otm2, otm3 : 2..5;
```

```
End;
```

```
Var Kurs : array [1..100] of Stud;
```

Написать программу, которая вводит эту информацию и печатает номера групп в порядке убывания средней успеваемости их студентов.

Контрольная работа № 6

(файлы, модули)

Примерный вариант

1. Точно и однозначно сформулировать условие задачи, решение которой приведено ниже.

```
Program Example_File_1;
```

```
Var F : File Of Real; G : File Of Integer;
```

```
    S : String; A : Real; B : Integer;
```

```
Begin
```

```
    Write('Имя файла? '); ReadLn(S); Assign(F, S);
```

```
    Write('Имя файла? '); ReadLn(S); Assign(G, S);
```

```
    ReSet(F); ReWrite(G);
```

```
    While Not Eof(F) Do
```

```
        Begin
```

```
            Read(F, A); B := Trunc(A); B := Ord(Odd(B)) * B; Write(G, B)
```

```
        End; Close(F); Close(G)
```

```
End.
```

2. Решить задачу: заполнить файл целыми числами из отрезка `[-1000;`

1000] с помощью датчика случайных чисел. Переписать в один из новых файлов те из компонент исходного файла, модуль которых является простым числом, в другой — все остальные.

3. Решить задачу: сведения о результатах сессии студентов второго курса (фамилия, номер группы и оценки по трем экзаменам) хранятся в файле:

```
Type Stud = Record
    fam:string[12];
    num:121..126;
    otm1, otm2, otm3: 2..5;
end;
Var Kurs: File of Stud;
```

Написать программу, которая вводит эту информацию и печатает номера групп в по-рядке убывания средней успеваемости их студентов.

4. Решить задачу: удалить из файла, содержащего целые числа, неположительные компоненты. Дополнительных файлов и массивов не использовать.

5. Решить задачу: разработать модуль для работы с векторами вида $a(a_1, a_2)$. Реализовать операции сложения, вычитания, нахождения длины вектора. Используя модуль, переписать из данного файла в другой те векторы, длины которых больше заданного K .

Критерии оценки:

- **Оценка «5»:** контрольная работа выполнена правильно, сделаны правильные выводы; практическое задание выполнено без ошибок;
- **Оценка «4»:** контрольная работав основном выполнена правильно, но при этом допущены одна-две несущественные ошибки;
- **Оценка «3»:** контрольная работа выполнена с тремя существенными ошибками;
- **Оценка «2»:** контрольная работа не выполнена.

Темы рефератов

1. История развития термина алгоритм
2. Эволюция языков программирования
3. Классификация программного обеспечения ПК
4. Типы данных, определяемые программистом. Перечисляемый и интервальный типы
5. Форматы вывода
6. Обработка переполнения
7. Выделение цифр числа
8. Формирование числовых последовательностей
9. Дополнительные сведения о процедурах и функциях
10. Стили структурного программирования. Разработка программы
11. Открытый массив. Сортировка массива.
12. Процедуры при работе с множествами
13. Записи с вариантами
14. Применение ООП
15. Компоненты приложения
16. Разработка интерфейса пользователя. Рекомендации и методы.

Критерии оценки:

- оценка «зачтено» выставляется студенту, если тема раскрыта полностью;

- оценка «не зачтено» выставляется студенту, если тема не раскрыта или имеются серьёзные ошибки в изложении.

4. Контрольно-оценочные средства для аттестации по учебной дисциплине

Тестовые формы заданий

Вариант – 1

1. Алгоритм — это:

- а) правила выполнения определенных действий;
- б) ориентированный граф, указывающий порядок исполнения некоторого набора команд;
- в) понятное и точное предписание исполнителю совершить последовательность действий, направленных на достижение поставленных целей;
- г) набор команд для компьютера;
- д) протокол вычислительной сети.

2. Суть такого свойства алгоритма как *результативность* заключается в том, что:

- а) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);
- б) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд;
- в) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;
- г) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату;
- д) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма.

3. Алгоритм называется циклическим:

- а) если он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
- б) если ход его выполнения зависит от истинности тех или иных условий;
- в) если его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;
- г) если он представим в табличной форме;
- д) если он включает в себя вспомогательный алгоритм.

4. В алфавит языка Pascal не входит служебное слово:

- а) BEGIN;
- б) STEP;
- в) WHILE;
- г) THEN.

5. Подпрограммой называют:

- а) часть программы, содержащую неоднократно выполняемые команды;
- б) независимый программный модуль;
- в) произвольный фрагмент программы;
- д) часть программы, служащей для решения некоторой вспомогательной задачи.

6. Служебное слово VAR в программе на языке Pascal фиксирует:

- а) начало раздела программы, содержащего перечень констант;
- б) начало раздела программы, содержащего список меток;
- в) начало раздела программы, содержащего описание сложных типов данных;
- г) начало раздела программы, содержащего описание переменных.

7. Операторы в программе на языке Pascal отделяются друг от друга:

- а) двоеточием;
- б) точкой с запятой;
- в) пробелом;
- г) запятой.

8. Составное логическое выражение $(X \geq A) \text{ AND } (X \leq B)$ истинно при:

- а) $x \in [A, B]$;
- б) $x \in [A, \infty] \cup [B, \infty]$;
- в) $x \in [-\infty, A] \cup [B, +\infty]$;
- г) $x \in [A, \infty] \cup [-\infty, B]$.

9. При каком значении X при исполнении программы будет получен ответ “ДА”:

```
Program T21;  
Var X: Integer;  
Begin  
  Readln(X); If X MOD 2=0 Then Writeln('ДА') Else  
  Writeln('НЕТ')
```

End.

- а) 3;
- б) 7;
- в) 15;
- г) 4.

10. Физическим именем файла в языке программирования Pascal называют:

- а) имя программы;
- б) имя переменной, используемой в программе при осуществлении операций над файлом;
- в) имя файла, под которым он записан на внешнем устройстве;
- г) имя файла, под которым программа хранится на диске.

11. Какое из приведенных неравенств верно:

- а) $-4.9876543234 \text{ E}-02 < -0,03$;
- б) $-4.9876543234 \text{ E}-02 < -0,5$;
- в) $-4.9876543234 \text{ E}-02 > -0,03$;
- г) $-4.9876543234 \text{ E}-02 < -0,4$.

12. Какая из перечисленных операций не является логической:

- а) NOT;
- б) OR;
- в) DIV;
- г) AND.

Вариант – 2

1. Суть такого свойства алгоритма как *массовость* заключается в том, что:

- а) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);
- б) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд;
- в) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;
- г) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату;
- д) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма.

2. Алгоритм называется линейным:

- а) если он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;
- б) если ход его выполнения зависит от истинности тех или иных условий;
- в) если его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;
- г) если он представим в табличной форме;
- д) если он включает в себя вспомогательный алгоритм.

3. В качестве имени в языке Pascal нельзя использовать сочетания:

- а) AR;
- б) BR;
- в) WR;
- г) OR.

4. Укажите последовательность команд, в результате выполнения которых значения переменных X и Y поменяются местами:

- а) $X=X+Y$; $Y=X-Y$; $X=X-Y$;
- б) $V=X$; $X=Y$; $Y=X$;

- в) $X=Y; Y=X;$
- г) $Y=X; B=X; X=Y;$
- д) $C=X; X=Y; X=C.$

5. Числа в языке Pascal различаются:

- а) как натуральные и вещественные;
- б) как натуральные и целые;
- в) как целые и вещественные;
- г) как целые и правильные дроби.

6. Служебное слово LABEL в программе на языке Pascal фиксирует:

- а) начало раздела программы, содержащего список меток;
- б) начало раздела программы, содержащего описание переменных;
- в) начало раздела программы, содержащего описание сложных типов данных;
- г) начало раздела программы, содержащего перечень констант.

7. Для вывода результатов работы программы на языке Pascal служит оператор:

- а) READ;
- б) WRITE;
- в) VAR;
- г) GOTO.

8. Какая из перечисленных операций не является логической:

- а) NOT;
- б) OR;
- в) MOD;
- г) AND.

9. Сложное логическое выражение $(X \leq A) \text{ OR } (X \geq B)$ истинно при:

- а) $x \in [-\infty, A] \cup [B, +\infty];$
- б) $x \in [A, B];$
- в) $x \in [A, \infty] \cup [B, \infty];$
- г) $x \in [A, \infty] \cup [-\infty, B].$

10. Результатом вычисления функции $\text{Cору}(\text{'информатика'}, 3, 5)$ будет слово:

- а) атика;
- б) рма;
- в) форма;
- г) инфор.

11. Дана программа:

```
Program T26; Var a: array[1..8] of Integer; M, k:
Integer;
Begin For k:=1 to 8 Do Readln(a[k]); M:=a[1];
For k:=2 to 8 Do If M<a[k] Then M:=A[k]; Write(M);
```


End.

Сколько раз будет исполнен оператор $M:=A[k]$ при заданном массиве: 3, 8, 7, 9, 4, 10, 2, 12:

а) 7;

б) 8;

в) 4;

г) 1.

12. Функция обработки признака конца файла F имеет вид:

а) REWRITE(F);

б) RESET(F);

в) CLOSE(F);

г) EOF(F).

Вариант – 3

1. Системы программирования:

а) обеспечивают непосредственное решение пользовательских задач;

б) позволяют создавать новые программы на языках программирования;

в) обеспечивают работу всех аппаратных устройств компьютера и доступ пользователя к ним;

г) представляют собой совокупность программ, используемых для различных операций с документами;

д) предназначены для уничтожения компьютерных вирусов и зараженных ими файлов.

2. Характерным признаком линейной программы является:

а) строго последовательное выполнение операторов в порядке их записи;

б) наличие в каждой программной строке только одного операторов;

в) использование в ней исключительно операторов присваивания;

г) присутствие в ней операторов условного и безусловного перехода;

д) присутствие в ней операторов цикла.

3. Алгоритм включает в себя ветвление, если:

а) если он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий;

б) если ход его выполнения зависит от истинности тех или иных условий;

в) если его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий;

г) если он представим в табличной форме;

д) если он включает в себя вспомогательный алгоритм.

4. Массив — это:

а) поименованный набор фиксированного числа однотипных данных;

- б) ограниченная апострофами последовательность любых символов;
- в) совокупность разнородных данных, описываемых и обрабатываемых как единое целое;
- г) именованный набор однотипных данных на диске;
- д) набор переменных, начинающихся с одной буквы.

5. Служебное слово TYPE в программе на языке Pascal фиксирует:

- а) начало раздела программы, содержащего описание сложных типов данных;
- б) начало раздела программы, содержащего перечень констант;
- в) начало раздела программы, содержащего список меток;
- г) начало раздела программы, содержащего описание переменных.

6. Комментарий к тексту программы на языке Pascal заключается:

- а) в круглые скобки;
- б) в фигурные скобки;
- в) в квадратные скобки;
- г) между служебными словами Begin, End.

7. Оператор организации ввода данных с клавиатуры записывается с использованием служебного слова:

- а) VAR;
- б) WRITE;
- в) READ;
- г) GOTO.

8. Сколько раз будут выполнены операторы тела цикла при выполнении следующего фрагмента программы:

```
For K:=M Downto 1 Do  
  If (N MOD K=0) AND (M MOD K=0) Then Goto 1;  
  1: Writeln(K);
```

при N=96, M=36:

- а) 25;
- б) 24;
- в) 36;
- г) 12.

9. Результатом работы программы:

```
Program T34;  
Var x: string[6];  
  Begin x:='мим'+ 'озадаченный'; Writeln(x); End.
```

будет слово:

- а) мим озадаченный;
- б) мимозадаченный;
- в) мимоза;

г) озадаченный мим.

10. Логическим именем файла в языке программирования Pascal называют:

- а) имя переменной, использующейся в программе при осуществлении операций над файлом;
- б) имя файла, под которым он записан на внешнем устройстве;
- в) имя программы;
- г) имя файла, под которым программа хранится на диске.

11. Чему равно значение выражения EXP(4*LN(5)):

- а) 1024;
- б) 20;
- в) 25;
- г) 625.

12. Какая из перечисленных операций является логической:

- а) NOT;
- б) DIV;
- в) MOD;
- г) TRUNC.

Бланк ответов

М.П.

Дисциплина

Основы программирования

Номер варианта теста

Дата

Группа

ФИО

№ ВОПРОСА	ВАРИАНТ ОТВЕТА
1	
2	
3	
...	
12	

Эталоны ответов:

Дисциплина

Основы
программирования

Номер варианта теста

№1

Вопро
с 1
в

Вопро
с 2
г

Вопро
с 3
а

Вопро
с 4
б

Вопро
с 5
д

Вопро
с 6
г

Вопро
с 7
б

Вопро
с 8
а

Вопро
с 9
г

Вопро
с 10
в

Вопро
с 11
а

Вопро
с 12
в

Дисциплина								Основы программирования			
Номер варианта теста								№2			
Вопро с 1	Вопро с 2	Вопро с 3	Вопро с 4	Вопро с 5	Вопро с 6	Вопро с 7	Вопро с 8	Вопро с 9	Вопро с 10	Вопро с 11	Вопро с 12
в	в	г	д	в	а	б	в	а	в	г	г

Дисциплина								Основы программирования			
Номер варианта теста								№3			
Вопро с 1	Вопро с 2	Вопро с 3	Вопро с 4	Вопро с 5	Вопро с 6	Вопро с 7	Вопро с 8	Вопро с 9	Вопро с 10	Вопро с 11	Вопро с 12
б	а	б	а	а	б	в	б	в	а	в	а

КРИТЕРИИ ОЦЕНОК:

Вопросы 1-12 по 1 баллу

6-7 баллов – оценка «удовлетворительно»

8-10 баллов – оценка «хорошо»

11-12 баллов – оценка «отлично»

Вариант - 1

1. Логическое выражение

$(N \text{ Mod } 10 \text{ Mod } 2) \text{ Or } (N \text{ Div } 10 \text{ Mod } 10 \text{ Mod } 2 = 0) \text{ Or } (N \text{ Div } 100 \text{ Mod } 2 = 0)$
должно принимать значение TRUE тогда и только тогда, когда истинно высказывание

- 1) в трёхзначном натуральном числе все цифры чётные;
- 2) в трёхзначном натуральном числе одна чётная цифра;
- 3) в трёхзначном натуральном числе две чётных цифры;
- 4) в трёхзначном натуральном числе хотя бы одна чётная цифра;
- 5) в трёхзначном натуральном числе нет чётных цифр.

2. Ошибку "Structure too large" (структура превышает максимально допустимый объём статической памяти) вызовет описание

- 1) Type Vector = Array[Byte] Of Integer; Var C : Array[1..10] Of Vector;
- 2) Var T : File Of String;
- 3) Type A = Record S : String; A, B, C : Array[10..20] Of Real End;
Var M : Array[1..5, 1..8] Of A;
- 4) Var K : Array [Byte, Byte] Of String[6];

5) Var S : Array[-10000..10000] Of String[2].

3. К процедурам для работы с динамическими переменными не относится

- 1) Mark;
- 2) New;
- 3) Release;
- 4) Seek;
- 5) Dispose.

4. Имеется описание

Type A = Array[0..100] Of Real; B = ^A; Var M : Array[1..5] Of B;

Для хранения массива M необходим объем памяти (байт)

- 1) 606;
- 2) 4;
- 3) 20;
- 4) 12120;
- 5) 6.

5. Фрагмент программы

```
    K := 0;
    While Not Eof(F) Do
    Begin ReadLn(F, S); I := 1;
        While I <= Length(S) Do
        Begin If S[I] In ['А'..'Я', 'а'..'я', 'р'..'я']
            Then Begin K := K + 1;
                    Delete(S, I, 1); I := I - 1
                End;
            I := I + 1
        End
    End
End;
```

выполняет следующее действие:

- 1) удаляет из текстового файла F все русские буквы;
- 2) определяет в текстовом файле количество символов, являющихся русскими буквами;
- 3) определяет в текстовом файле количество символов, не являющихся русскими буквами;
- 4) определяет в текстовом файле количество символов;
- 5) удаляет из текстового файла F все символы, не являющиеся русскими буквами.

6. В фрагменте программы (здесь Var F : File Of Integer; I, K, Vsp : Integer;)

```
Reset(F); K := FileSize(F) - 1;
For I := 0 To K Do
Begin Seek(F, I); Read(F, Vsp); Seek(F, FileSize(F)); Write(F, Vsp) End;
```

выполняется

- 1) сортировка файла;
- 2) изменение порядка следования элементов на обратный;

- 3) дописывание в конец исходного файла полную его копию с сохранением порядка следования элементов;
- 4) дописывание в конец исходного файла полную его копию с изменением порядка следования элементов на противоположный;
- 5) не выполняется никаких действий по изменению файла.

7. Имеется описание

Type Dn = (pn, vt, sr, cht, ptn, sb, vs); Mn = Set Of Dn; Var V : Mn;

и фрагмент программы

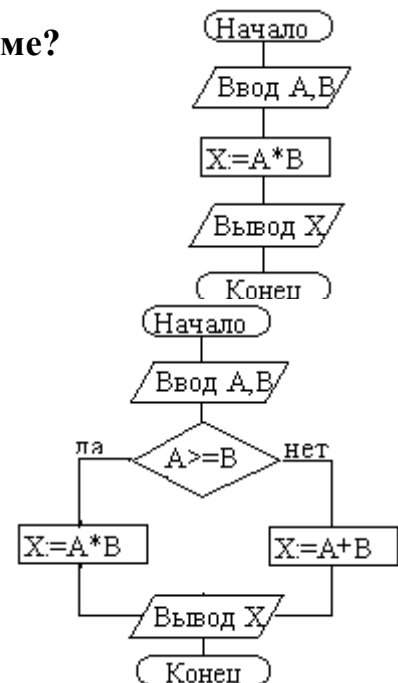
V := [pn..ptn] * [sr, ptn..vs] – [sb];

После исполнения этого фрагмента переменная V имеет значение

- 1) [pn..vs];
- 2) [sr, ptn];
- 3) [sb];
- 4) [];
- 5) [pn..ptn].

8. Алгоритм какого типа изображен на блок-схеме?

- 1) циклический;
- 2) разветвляющийся;
- 3) вспомогательный;
- 4) линейный;
- 5) комбинация развилки и цикла.



9. После исполнения фрагмента программы, изображенного на блок-схеме,

при A = 5, B = 4 значение X будет равно

- 1) 20;
- 2) 9;
- 3) 5;
- 4) 4;
- 5) 1.

10. В приведенном фрагменте программы

(N типа LongInt, N > 0)

```

P := 1;
While P <= N Do
Begin
    Left := N Div (P * 10) * (P * 10);
    Right := N Mod P;
    K := ((N Mod (P * 10) Div P + 1) Mod 10) * P;
    N := Left + K + Right; P := P * 10
End;

```

натуральное число N изменяется по следующему правилу

- 1) не изменяется;
- 2) в каждый разряд прибавляется 1;
- 3) из каждого разряда вычитается 1;
- 4) в каждый разряд прибавляется 1, если значение в разряде — не девять, иначе заменяется на нуль;
- 5) каждая девятка в десятичной записи числа заменяется на нуль.

11. Цикл с предусловием выполняется так:

- 1) выполняется тело цикла, изменяется параметр цикла, проверяется условие продолжения выполнения цикла;
- 2) изменяется параметр цикла, проверяется условие продолжения выполнения цикла, выполняется тело цикла;
- 3) проверяется условие продолжения выполнения цикла, выполняется тело цикла;
- 4) тело цикла выполняется N раз (N — натуральное);
- 5) определяется, сколько раз должен быть выполнен цикл, и далее цикл с предусловием сводится к циклу с параметром.

12. В текстовом файле каждая строка заканчивается

- 1) числами 10 и 13;
- 2) символами с кодами 10 и 13;
- 3) символом с кодом 13;
- 4) числом 0;
- 5) символом с кодом 10.

13. Процедуры ReadLn и WriteLn можно использовать при работе с

- 1) типизированными файлами;
- 2) нетипизированными файлами;
- 3) типизированными и нетипизированными файлами;
- 4) текстовыми файлами;
- 5) любыми файлами.

14. Значение выражения $\text{Ord}(x > y) + \text{Ord}(\text{Ord}(z = 'F'))$

при $x = 7, y = 0, z = 'F'$ равно

- 1) TRUE;
- 2) FALSE;
- 3) 0;
- 4) 1;
- 5) 2.

15. Идентификатор в Turbo Pascal не может начинаться с

- 1) латинской буквы;
- 2) заглавной латинской буквы;
- 3) цифры;
- 4) знака подчёркивания;
- 5) латинской буквы, а затем знака подчёркивания.

16. В приведенном фрагменте программы (First — ссылка на первый элемент списка; список объявлен следующим образом:

Type SS = ^List; List = Record A : LongInt; Next : SS End;)

P := First; S := 0; While Not (P = Nil) Do Begin S := S + 1; P := P^.Next End;

определяется

- 1) первый элемент списка;
- 2) сумма элементов списка;
- 3) сумма первого и последнего элементов списка;
- 4) количество элементов списка;
- 5) количество звеньев списка, где указатель на следующее звено не Nil.

17. При выполнении фрагмента программы

```
Var C : Integer;  
    Procedure R1(Var A : Integer; C : Boolean);  
        Procedure R2;  
            Var C : String;  
            Begin A := 1 End;  
        Begin C := True; R2 End;  
    Begin C := 100; R1(C, False); WriteLn(C) End.
```

будет напечатано значение переменной C

- 1) True;
- 2) 1;
- 3) 100;
- 4) неизвестно что, поскольку значение переменной C не определено;
- 5) False.

18. Цикл в фрагменте программы

```
P := 4; Repeat P := P * 0.1 Until P < 0.0001;
```

будет выполнен

- 1) 0 раз;
- 2) 1 раз;
- 3) 4 раза;
- 4) 5 раз;
- 5) бесконечное число раз.

19. Кодировается шестнадцатитрибитовое целое со знаком (тип Integer).

1111111111110000₂ — это код числа

- 1) -15;
- 2) 15;
- 3) 16;
- 4) -16;
- 5) -30000.

20. Свойством алгоритма является

- 1) результативность;
- 2) цикличность;
- 3) возможность изменения последовательности выполнения команд;
- 4) возможность выполнения алгоритма в обратном порядке;
- 5) простота при записи на языках программирования.

Вариант - 2

1. Из перечисленных ниже в программе обязательен

- 1) раздел Var;
- 2) раздел Const;
- 3) раздел Type;
- 4) раздел Label;
- 5) раздел Begin ... End.

2. Ввод данных — это

- 1) процесс передачи данных из оперативной памяти на внешний носитель;
- 2) процесс ввода с клавиатуры каких-либо значений;
- 3) передача данных от внешнего носителя в оперативную память для обработки;
- 4) присваивание конкретных значений переменным, которые используются в программе;
- 5) запись файла на диск.

3. Значение R после выполнения операции логического присваивания

$$R := \text{Not} (A \text{ Or } B \text{ Or } (X > 2) \text{ And } (Y < 0))$$

при $A = \text{False}$, $B = \text{False}$, $X = 3$, $Y = 2$ будет равно

- 1) -1;
- 2) False;
- 3) True;
- 4) 0;
- 5) 1.

4. С помощью какой из приведенных серий команд переменной B присваивается значение выражения?

$$\left(\frac{x+y}{x-y} - \frac{x-y}{x+y} \right) \cdot \left(\frac{x-y}{x+y} + \frac{x+y}{x-y} \right)$$

- а) $A := (x + y) / (x - y)$; $B := (A - 1 / A) * (1 / A + A)$;
- б) $A := (x + y) / (x - y)$; $B := \text{Sqr}(A) - \text{Sqr}(1 / A)$;
- в) $A := (x - y) / (x + y)$; $B := \text{Sqr}(1 / A) - \text{Sqr}(A)$;

- 1) а;
- 2) б;
- 3) в;
- 4) всех трех;
- 5) ни один из ответов 1–4 не является верным.

5. Значения переменных a и b после выполнения следующих действий

$$a := 15 \text{ Div } (16 \text{ Mod } 7); b := 34 \text{ Mod } a * 5 - 29 \text{ Mod } 5 * 2;$$

будут равны

- 1) $a = 1$, $b = 160$; 2) $a = 1$, $b = 4$; 3) $a = 7$, $b = 25$; 4) $a = 7$, $b = 22$; 5) $a = 7$, $b = 28$.

6. Во фрагмент алгоритма

```
For K := 10 To 99 Do
  Begin P1 := K Div 10;
        P2 := K Mod 10;
        S := P1 + P2;
        If ____ Then WriteLn(K)
  End;
```

печатающий все двузначные числа, в записи которых есть цифра N или сумма цифр которых равна самим числам, нужно вписать логическое выражение

- 1) $(P1 = N) \text{ Or } (P2 = N) \text{ And } (S = K)$;
- 2) $(P1 = N) \text{ Or } (P2 = N) \text{ Or } (S = K)$;
- 3) $(P1 = N) \text{ And } (P2 = N) \text{ Or } (S = K)$;
- 4) $((P1 = N) \text{ Or } (P2 = N)) \text{ And } (S = K)$;
- 5) $(P1 = N) \text{ And } (P2 = N) \text{ And } (S = N)$.

7. Значения переменных p и d после выполнения фрагмента алгоритма

```
k := 47; Case k Mod 9 Of
  5: Begin d := k; p := True End;
  0..2: Begin d := 2; p := False End;
  8: Begin d := 1; p := False End
  Else Begin d := 1; p := True End
End;
```

равны

- 1) $p = \text{True}, d = 1$;
- 2) $p = \text{False}, d = 2$;
- 3) $p = \text{False}, d = 3$;
- 4) $p = \text{True}, d = 47$;
- 5) $p = \text{True}, d = 2$.

8. Тело цикла в программе

```
a := 1; b := 1; While a + b < 8 Do begin a := a + 1; b := b + 2 end;
```

выполнится

- 1) 1 раз;
- 2) 2 раза;
- 3) 3 раза;
- 4) ни разу;
- 5) бесконечное число раз.

9. Элементы массива $p[1..5]$ равны соответственно 1, -1, 5, 2, 4. Значение выражения

$$p[1] * p[3] - p[2 * p[2] + p[p[5] - p[2]]]$$

равно

- 1) 8;
- 2) -8;
- 3) 12;

- 4) -12;
- 5) 6.

10. Задана строка *St*. Фрагмент алгоритма

```
S := 0;  
For I := 1 To Length (St) Do  
Begin  
    Val(St[I], d, k);  
    If K = 0 Then S := S + d  
End;
```

- 1) определяет количество цифр в строке;
- 2) подсчитывает количество нулей в строке;
- 3) определяет сумму номеров позиций в строке, где стоят цифры;
- 4) подсчитывает сумму цифр в строке;
- 5) определяет сумму номеров позиций в строке, где стоят нули.

11. Какая из приведенных серий операторов определяет и печатает индекс последнего отрицательного элемента в линейном массиве из *n* элементов?

- а) $i := n$; While ($i \geq 1$) And ($m[i] > 0$) Do Dec (i); If $i < 1$ Then WriteLn (' $i = 0$ ') Else WriteLn (' $i =$ ', i);
- б) $k := 0$; For $i := 1$ To n Do If $m[i] < 0$ Then $k := i$; WriteLn (' $i =$ ', k);
- в) $i := n$; Repeat $i := i - 1$ Until ($m[i] < 0$); WriteLn (' $i =$ ', i);

- 1) а, б;
- 2) б, в;
- 3) а, б, в;
- 4) б;
- 5) ни один из ответов 1–4 не верен.

12. Задан линейный массив $M[1..n]$.

```
Function Control (M: Myarray): Boolean;  
Var I : Integer;  
Begin I := 1;  
    While (I <= n) And (M[I] > 0) Do Inc(I);  
    Control := (I <= n);  
End;
```

Если в данном массиве все элементы положительные, приведенная функция возвращает значение

- 1) n ;
- 2) True;
- 3) False;
- 4) $I \leq n$;
- 5) ни один из ответов 1–4 не верен.

13. Задан двумерный массив $X[1..n, 1..m]$. Процедура

```
Procedure Sub (Var X: Myarray);  
Var i, j: Integer;  
Begin For i := 1 To n Do
```

For j := 1 To m Div 2 Do X[i, 2 * j] := X[i, 2 * j] + X[i, 1];

End;

- 1) к элементам столбцов в первой половине матрицы прибавляет элементы первого столбца соответствующей строки;
- 2) добавляет к матрице еще M столбцов с элементами, равными соответствующим элементам первого столбца;
- 3) к элементам четных столбцов прибавляет элементы первого столбца соответствующей строки;
- 4) к элементам четных строк прибавляет элементы первой строки соответствующего столбца;
- 5) меняет порядок столбцов таблицы.

14. Задан двумерный массив X[1..n, 1..m]. Функция

```
Function Check (X: Myarray): Boolean;  
  Var i, j : Integer; t : Boolean;  
  Begin  t := True; i := 1;  
         While t And (i <= n) Do  
           Begin  j := 1; While (j <= m) And (X[i, j] <> 0) Do Inc (j);  
                 t := (j = m + 1); Inc (i)  
           End;  
         End;  
         Check := Not t
```

End;

возвращает значение

- 1) True, если все элементы массива ненулевые;
- 2) True, если в массиве есть элемент, равный нулю;
- 3) False, если в массиве есть элемент, равный нулю;
- 4) Not t;
- 5) ни один из ответов 1–4 не верен.

15. Среди перечисленных соответствий, которые необходимо соблюдать между формальными и фактическими параметрами

- а) соответствие по типу параметров;
- б) соответствие по количеству параметров;
- в) соответствие по типу используемых вспомогательных переменных;
- г) соответствие по порядку перечисления;

лишним является

- 1) а;
- 2) б;
- 3) в;
- 4) г;
- 5) ни один из ответов 1–4 не верен.

16. Определите тип выражения (здесь A : Array[1..20] Of Real; B : Boolean; C : Integer)

C + Ord(Round(A[7]) + Ord(B)) – Trunc(A[1])

- 1) Real;

- 2) Integer;
- 3) Boolean;
- 4) Byte;
- 5) среди ответов 1–4 нет верного.

17. Список объявлен следующим образом

```
Type Ukaz = ^Zveno; Zveno = Record X : String; N : Ukaz End;
Var First : Ukaz; {ссылка на начало списка}
```

В следующем фрагменте программы

```
P := First;
While P^.N <> Nil Do
Begin B := P; M := P;
While B <> Nil Do Begin If B^.X < M^.X Then M := B; B
:= B^.N End;
S := P^.X; P^.X := M^.X; M^.X := S; P := P^.N
End;
```

выполняется

- 1) перемещение компонента к началу списка;
- 2) сортировка компонентов списка в порядке возрастания;
- 3) сортировка компонентов списка в порядке убывания;
- 4) перестановка соседних компонентов списка;
- 5) добавление в список нескольких новых компонент.

18. Фрагмент программы

```
S := A; A := B; B := S;
```

выполняет

- 1) обмен значений переменных A , B ;
- 2) присваивание переменным A , B значения S ;
- 3) замена значения переменной A значением переменной B ;
- 4) во фрагменте не выполняется никаких действий;
- 5) замена значения переменной B значением переменной A .

19. Имеется следующее описание

```
Type U = ^Zveno; Zveno = Record X, Y : Boolean; Pred, Next : U
End;
Var Logic : Boolean; A, B : Pointer; X, Y : U;
```

К ошибке компиляции "Несовместимость типов" приведет следующее присваивание

- 1) $A := X^.Next^.Next$;
- 2) $X := Y$;
- 3) $Logic := X^.X$;
- 4) $X^.Next := A$;
- 5) $X^ := Y^.Next$.

20. При присваивании изменяется

- 1) алгоритм;
- 2) имя переменной;

- 3) тип переменной;
- 4) значение переменной;
- 5) значение константы.

Бланк ответов

М.П.	Дисциплина	<u>Основы программирования</u>
	Номер варианта теста	_____
	Дата	_____
	Группа	_____
	ФИО	_____

№ ВОПРОСА	ВАРИАНТ ОТВЕТА
1	
2	
3	
...	
20	

Эталоны ответов:

	Дисциплина	Основы
		<u>программирования</u>
	Номер варианта теста	<u>№1</u>
4	Вопрос 1	4
4	Вопрос 2	4
4	Вопрос 3	4
3	Вопрос 4	3
2	Вопрос 5	2
3	Вопрос 6	3
2	Вопрос 7	2
4	Вопрос 8	4
1	Вопрос 9	1
4	Вопрос 10	4
3	Вопрос 11	3
2	Вопрос 12	2
4	Вопрос 13	4
5	Вопрос 14	5
3	Вопрос 15	3
4	Вопрос 16	4
2	Вопрос 17	2
4	Вопрос 18	4
4	Вопрос 19	4
1	Вопрос 20	1

	Дисциплина	Основы
		<u>программирования</u>
	Номер варианта теста	<u>№2</u>
1	Вопрос 1	1
3	Вопрос 2	3
3	Вопрос 3	3
4	Вопрос 4	4
4	Вопрос 5	4
2	Вопрос 6	2
2	Вопрос 7	2
2	Вопрос 8	2
5	Вопрос 9	5
4	Вопрос 10	4
1	Вопрос 11	1
3	Вопрос 12	3
3	Вопрос 13	3
2	Вопрос 14	2
3	Вопрос 15	3
2	Вопрос 16	2
2	Вопрос 17	2
1	Вопрос 18	1
5	Вопрос 19	5
4	Вопрос 20	4

КРИТЕРИИ ОЦЕНОК:

Вопросы 1-20 по 1 баллу

11-13 баллов – оценка «удовлетворительно»

14-18 баллов – оценка «хорошо»

19-20 баллов – оценка «отлично»

Экзаменационные билеты и практические задания к ним

**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА
ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ПУТЕЙ
СООБЩЕНИЯ» в г.Алатыре**

ОДОБРЕНО на заседании ЦК математических и общих естественнонаучных дисциплин Протокол № _____ Председатель _____ Р.В.Пасюнина « ____ » _____ 2020г.	Экзаменационный билет №1 <u>Основы программирования</u> Дисциплина Группа <u>ПО-19-1 Семестр 5</u>	УТВЕРЖДАЮ: Заместитель директора по УР _____ Т.Ю.Базилевич « ____ » _____ 2020г.
---	---	--

Коды проверяемых и общих компетенций ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 3.1, ОК 1, ОК 2, ОК 3, ОК 4, ОК 5, ОК 6, ОК 7, ОК 8, ОК 9.

Место проведения экзамена – кабинет №301

Инструкция:

Внимательно прочитайте задание.

При выполнении задания вы можете воспользоваться методическими пособиями по программе из папки «К экзамену»

Время выполнения задания – 30 минут.

Задания

1. Этапы решения задач с использованием ЭВМ.
2. Сортировка массивов. Методы сортировки массивов.

Практическое задание № 1. Составить программу для решения квадратного уравнения.

Критерии оценки

Оценка «5»:	- Выполнены полностью часть 1, 2, 3
Оценка «4»	- Выполнены полностью часть 1,2 задание из части 3 выполнено не полностью
Оценка «3»	- Выполнена только часть 1 и 2, или 1 и 3, или 2 и 3
Оценка «2»	- Работа не выполнена

Преподаватель: _____/Скворцова И.И./

КРИТЕРИИ ОЦЕНОК ПО ДИСЦИПЛИНЕ

Таблица 4

Оценка «5»:	<ul style="list-style-type: none"> - ответ полный и правильный на основании изученных теорий; - материал изложен в определенной логической последовательности, литературным языком; - ответ самостоятельный. - работа выполнена полностью и правильно; - сделаны правильные выводы; - работа выполнена по плану с учетом техники безопасности
Оценка «4»	<ul style="list-style-type: none"> - ответ полный и правильный на основании изученных теорий; - материал изложен в определенной логической последовательности, при этом допущены две-три несущественные ошибки, исправленные по требованию преподавателя; - работа выполнена правильно с учетом 2-3 несущественных ошибок, исправленных самостоятельно по требованию преподавателя.
Оценка «3»	<ul style="list-style-type: none"> - ответ полный, но при этом допущена существенная ошибка, или неполный, несвязный. - работа выполнена правильно не менее чем на половину или допущена существенная ошибка.
Оценка «2»	<ul style="list-style-type: none"> - при ответе обнаружено непонимание студентом основного содержания учебного материала или допущены существенные ошибки, которые студент не смог исправить при наводящих вопросах преподавателя; - отсутствие ответа; - допущены две (и более) существенные ошибки в ходе работы, которые студент не может исправить даже по требованию преподавателя; - работа не выполнена

ЛИТЕРАТУРА

Перечень учебных изданий, Интернет-ресурсов, дополнительной литературы

Основные источники:

Основные источники:

1. Калинина, Н.А. Основы программирования на языке С : курс лекций / Калинина Н.А., Костюкова Н.И. — Москва : Интуит НОУ, 2016. — 236 с. — ISBN 978-5-9556-0057-4. — URL: <https://book.ru/book/917908>
 2. Костюк, Ю. Л. Лекции по основам программирования : учебное пособие / Ю. Л. Костюк. — Томск : ТГУ, 2019. — 260 с. — ISBN 978-5-94621-827-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/148655>
 3. Окулов, С. М. Основы программирования : учебное пособие / С. М. Окулов. — 10-е изд. — Москва : Лаборатория знаний, 2020. — 339 с. — ISBN 978-5-00101-759-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/135560>
- Тракимус, Ю. В. Основы программирования : учебное пособие / Ю. В. Тракимус, В. П. Хиценко. — Новосибирск : НГТУ, 2020. — 66 с. — ISBN 978-5-7782-4089-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/152224>