Частное профессиональное образовательное учреждение
«Донецкий институт Южного Университета (ИУБиП)»

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 02F0D676003BB12EBC4FA21E91B6D6C80C
Владелец: ЧПОУ ДИ ЮУ
Действителен: с 22.03.2024 до 22.06.2025

Комплект контрольно-оценочных средств по учебной дисциплине

ОП.04	**Основы алгоритмизации и программирования**
(индекс)	(наименование)
	специальность
09.02.07	**Информационные системы и программирование**
(шифр)	(наименование)

СОДЕРЖАНИЕ

	стр.
1. ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ	3
2. РЕЗУЛЬТАТЫ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ	5
3. ОЦЕНКА ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	7
4. КОНТРОЛЬНО-ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ИТОГОВОЙ АТТЕСТАЦИИ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ	51

1 ПАСПОРТ КОМПЛЕКТА КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ

В результате освоения учебной дисциплины Основы алгоритмизации и программирования обучающийся должен обладать предусмотренными ФГОС по специальности СПО 09.02.07 Информационные системы и программирование следующими умениями, знаниями, которые формируют профессиональную компетенцию, и общими компетенциями:

Умения:

У1 – разрабатывать алгоритмы для конкретных задач;

У2 – использовать программы для графического отображения алгоритмов;

У3 – определять сложность работы алгоритмов;

У4 – работать в среде программирования;

У5 – реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;

У6 – оформлять код программы в соответствии со стандартом кодирования;

У7 – выполнять проверку, отладку кода программы.

Знания:

З1 – понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;

З2 – эволюцию языков программирования, их классификацию, понятие системы программирования;

З3 – основные элементы языка, структуру про граммы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;

З4 – подпрограммы, составление библиотек подпрограмм;

З5 – объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на при мере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Общие компетенции:

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

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

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде

ОК.05 Осуществлять устную и письменную коммуникацию на государствен-ном языке Российской Федерации с учетом особенностей социального и культурно-го контекста

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках

Профессиональные компетенции:

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

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

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

2 РЕЗУЛЬТАТЫ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ, ПОДЛЕЖАЩИЕ ПРОВЕРКЕ

В результате аттестации по учебной дисциплине осуществляется комплексная проверка следующих умений и знаний, а также динамика формирования общих компетенций:

Таблица 2.1 – Результаты освоения учебной дисциплины

Результаты обучения: умения, знания и общие компетенции	**Критерии оценки**	**Форма контроля и оценивания**
Умения: У1 – разрабатывать алгоритмы для конкретных задач; У2 – использовать программы для графического отображения алгоритмов; У3 – определять сложность работы алгоритмов; У4 – работать в среде программирования; У5 – реализовывать построенные алгоритмы в виде программ на конкретном языке программирования; У6 – оформлять код программы в соответствии со стандартом кодирования; У7 – выполнять проверку, отладку кода программы **Знания:** З1 – понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции; З2 – эволюцию языков программирования, их классификацию, понятие системы программирования; З3 – основные элементы языка, структуру про граммы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти; З4 – подпрограммы, составление библиотек подпрограмм; З5 – объектно-ориентированную модель программирования,	Оценка **«Отлично»** - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко. Оценка **«Хорошо»** - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками. Оценка **«Удовлетворительно»** - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки. Оценка **«Неудовлетворительно»** - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.	Устный опрос Практическая работа Самостоятель-ная работа Тестирование Экзамен

основные принципы объектно-ориентированного программирования на при мере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения		
Общие компетенции:		
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам ОК 02. Использовать современные средства поиска, анализа интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности ОК 04. Эффективно взаимодействовать и работать в коллективе и команде ОК.05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках		

3 ОЦЕНКА ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1 Формы и методы оценивания

Предметом оценки служат умения и знания, предусмотренные ФГОС по дисциплине ОП.04 Основы алгоритмизации и программирования, направленные на формирование общих компетенций.

Таблица 3.1 – Контроль и оценка освоения учебной дисциплины по темам (разделам)

Элемент учебной дисциплины	**Формы и методы контроля**					
	Текущий контроль		**Рубежный контроль**		**Промежуточная аттестация**	
	Форма контроля	**Проверяемые ОК, У, З**	**Форма контроля**	**Проверяемые ОК, У, З**	**Форма контроля**	**Проверяемые ОК, У, З**
Раздел 1. Введение в программирование			Устный опрос Самостоятельная работа	У1 – У7, З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5	Экзамен	У1 – У7, З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5
Тема 1.1 Языки программирования	Устный опрос	У1 – У7, З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4,				

		ПК 2.5				
Тема 1.2 Типы данных	Устный опрос Самостоятельная работа	У1 – У7, З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5				
Раздел 2. **Обзор языка программирования**			Устный опрос Лабораторное занятие Самостоятельная работа	У1 – У7, З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5	Экзамен	У1 – У7, З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5
Тема 2.1 Операторы языка программирования	Устный опрос Лабораторное занятие Самостоятельная работа	У1 – У7, З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5				
Раздел 3.			Устный опрос	У1 – У7,	Экзамен	У1 – У7,

Функции и структура программы			Лабораторное занятие Самостоятельная работа	З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5		З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5
Тема 3.1 Процедуры и функции	Устный опрос Лабораторное занятие	У1 – У7, З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5				
Тема 3.2 Структуризация в программировании	Устный опрос	У1 – У7, З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5				
Тема 3.3 Модульное программирование	Устный опрос Лабораторное занятие					

	Самостоятельная работа					
Раздел 4. Основные конструкции языков программирования			Устный опрос Лабораторное занятие Самостоятельная работа	У1 – У7, З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5		
Тема 4.1 Указатели	Устный опрос Лабораторное занятие Самостоятельная работа	У1 – У7, З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5				
Раздел 5. Объектно-ориентированное программирование			Устный опрос Лабораторное занятие Самостоятельная работа	У1 – У7, З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5		

Тема 5.1 Основные принципы объектно-ориентированного программирования (ООП)	Устный опрос Лабораторное занятие	У1 – У7, З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5				
Тема 5.2 Интегрированная среда разработчика	Устный опрос Лабораторное занятие	У1 – У7, З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5				
Тема 5.3 Визуальное событийно-управляемое программирование	Устный опрос	У1 – У7, З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5				
Тема 5.4 Разработка оконного	Устный опрос Лабораторное	У1 – У7, З1 – З5,				

приложения	занятие	ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5				
Тема 5.5 Этапы разработки приложений	Устный опрос Лабораторное занятие	У1 – У7, З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5				
Тема 5.6 Иерархия классов	Устный опрос Лабораторное занятие Самостоятельная работа	У1 – У7, З1 – З5, ОК 01, ОК 02, ОК 04, ОК 05, ОК 09 ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 2.4, ПК 2.5				

3.2 Типовые задания для оценки освоения учебной дисциплины

3.2.1 Лабораторные занятия

Лабораторное занятие № 1

Знакомство со средой программирования

По указанию преподавателя выберите свое индивидуальное задание в Visual Studio. Уточните условие задания, количество, наименование, типы исходных данных. С помощью окна свойств установите первоначальный цвет формы, шрифт выводимых символов.

Индивидуальные задания

1. Разместите на форме четыре кнопки (Button). Сделайте на кнопках следующие надписи: «красный», «зеленый», «синий», «желтый». Создайте четыре обработчика события нажатия на данные кнопки, которые будут менять цвет формы в соответствии с текстом на кнопках.

2. Разместите на форме две кнопки (Button) и одну метку (Label). Сделайте на кнопках следующие надписи: «привет», «до свидания». Создайте обработчики события нажатия на данные кнопки, которые будут менять текст метки на слова, написанные на кнопках. Создайте обработчик события создания формы (Load), который будет устанавливать цвет формы и менять текст метки на строку «Начало работы».

3. Разместите на форме ряд кнопок (Button) напротив каждой поле ввода (TextBox) и одну метку (Label). Создайте обработчики события нажатия на данные кнопки, которые будут менять текст в метке. Текст метке берется из поля ввода напротив нажимаемой кнопки.

4. Разместите на форме ряд кнопок (Button), и одно поле ввода (TextBox). Создайте обработчики события нажатия на данные кнопки, которые будут менять текст на нажатой кнопке. Текст на кнопке берется из поля ввода.

5. Разместите на форме ряд кнопок (Button) и ряд меток (Label). Создайте обработчики события нажатия на данные кнопки, которые будут менять цвет двух меток. Создайте обработчик события нажатия кнопки мыши на форме (Click), который будет устанавливать цвет всех меток в белый.

6. Разместите на форме ряд кнопок (Button) и ряд меток (Label). Создайте обработчик события создания формы (Load), который будет делать все метки невидимыми. Создайте обработчики события нажатия на кнопки, которые будут менять свойство метки Visible, тем самым делать их видимыми.

7. Разместите на форме ряд кнопок (Button), напротив каждой по-ле ввода (TextBox). Создайте обработчики события нажатия на данные кнопки, которые будут менять заголовок окна. Текст в заголовке берется из поля ввода напротив нажимаемой кнопки.

8. Разместите на форме две кнопки (Button) и одну метку (Label). Сделайте на кнопках следующие надписи: «скрыть», «показать». Создайте обработчики события нажатия на данные кнопки, которые будут скрывать или показывать метку. Создайте обработчик события создания формы (Load), который будет устанавливать цвет формы и менять текст метки на строку «Начало работы».

9. Разместите на форме три кнопки (Button) и одно поле ввода (TextBox). Сделайте на кнопках следующие надписи: «скрыть», «показать», «очистить». Создайте обработчики события нажатия на данные кнопки, которые будут скрывать или показывать поле ввода.

При нажатии на кнопку «очистить» текст из поля ввода должен быть удален.

10. Разместите на форме две кнопки (Button) и одно поле ввода (TextBox). Сделайте на кнопках следующие надписи: «заполнить», «очистить». Создайте обработчики события нажатия на данные кнопки, которые будут очищать или заполнять поле ввода знаками «******». Создайте обработчик события создания формы (Load), который будет устанавливать цвет формы и менять текст в поле ввода на строку «+++++».

11. Разработайте игру, которая заключается в следующем. На форме размещены пять кнопок (Button). При нажатии на кнопку некоторые кнопки становятся видимыми, а другие – невидимыми. Цель игры – скрыть все кнопки.

12. Разработайте игру, которая заключается в следующем. На форме размещены четыре кнопки (Button) и четыре метки (Label). При нажатии на кнопку часть надписей становится невидимой, а часть, наоборот, становится видимой. Цель игры – скрыть все надписи.

13. Разместите на форме ряд кнопок (Button). Создайте обработчики события нажатия на данные кнопки, которые будут делать неактивными текущую кнопку. Создайте обработчик события изменения размера формы (Resize), который будет устанавливать все кнопки активный режим.

14. Разместите на форме ряд кнопок (Button). Создайте обработчи-ки события нажатия на данные кнопки, которые будут делать неактив-ными следующую кнопку. Создайте обработчик события нажатия кнопки мыши на форме (Click), который будет устанавливать все кнопки в активный режим.

15. Разместите на форме три кнопки (Button) и одно поле ввода (TextBox). Сделайте на кнопках следующие надписи: «*****», «++++++», «00000». Создайте обработчики события нажатия на данные кнопки, которые будут выводить текст, написанный на кнопках, в поле ввода. Создайте обработчик события создания формы (Load), который будет устанавливать цвет формы и менять текст в поле ввода на строку «Готов к работе».

16. Разместите на форме ряд полей ввода (TextBox). Создайте обработчики события нажатия кнопкой мыши на данные поля ввода, которые будут выводить в текущее поле ввода его номер. Создайте обработчик события изменения размера формы (Resize), который будет очищать все поля ввода.

17. Разместите на форме поле ввода (TextBox), метку (Label) кнопку (Button). Создайте обработчик события нажатия на кнопку, который будет копировать текст из поля ввода в метку. Создайте обработчик события нажатия кнопки мышки на форме (Click), который будет устанавливать цвет формы и менять текст метки на строку «Начало работы» и очищать поле ввода.

18. Разместите на форме поле ввода (TextBox) и две кнопки (Button) с надписями: «блокировать», «разблокировать». Создайте обработчики события нажатия на кнопки, которые будут делать активным или неактивным поле ввода. Создайте обработчик события нажатия кнопки мышки на форме (Click), который будет устанавливать цвет формы и делать невидимыми все элементы.

19. Реализуйте игру минер на поле 3×3 из кнопок (Button). Первоначально все кнопки не содержат надписей. При попытке нажатия на кнопку на ней либо показывается количество мин, либо надпись «мина!» и меняется цвет окна.

20. Разместите на форме четыре кнопки (Button). Напишите для каждой обработчик события, который будет менять размеры и местоположение на окне других кнопок.

Лабораторное занятие № 2
Составление программ линейной структуры

Ниже приведено 20 вариантов задач. По указанию преподавателя выберите свое индивидуальное задание. Уточните условие задания, количество, наименование, типы исходных данных. В соответствии с этим установите необходимое количество окон TextBox, тексты заголовков на форме, размеры шрифтов, а также типы переменных и функции преобразования при вводе и выводе результатов. Для проверки правильности программы после задания приведен контрольный пример: тестовые значения переменных, используемых в выражении, и результат, который при этом получается.

1. $t=\dfrac{2\cos\left(x-\dfrac{\pi}{6}\right)}{0.5+\sin^2 y}\left(1+\dfrac{z^2}{3-z^2/5}\right)$.

   При $x=14.26,\ y=-1.22,\ z=3.5\times10^{-2}$ $t=0.564849$.

2. $u=\dfrac{\sqrt[3]{8+\left|x-y\right|^2+1}}{x^2+y^2+2}-e^{|x-y|}\left(\mathrm{tg}^2 z+1\right)^x$.

   При $x=-4.5,\ y=0.75\times10^{-4},\ z=0.845\times10^2$ $u=-55.6848$.

3. $v=\dfrac{1+\sin^2(x+y)}{\left|x-\dfrac{2y}{1+x^2y^2}\right|}x^{|y|}+\cos^2\left(\mathrm{arctg}\dfrac{1}{z}\right)$.

   При $x=3.74\times10^{-2},\ y=-0.825,\ z=0.16\times10^2,\ v=1.0553$.

4. $w=\left|\cos x-\cos y\right|^{\left(1+2\sin^2 y\right)}\left(1+z+\dfrac{z^2}{2}+\dfrac{z^3}{3}+\dfrac{z^4}{4}\right)$.

   При $x=0.4\times10^4,\ y=-0.875,\ z=-0.475\times10^{-3}$ $w=1.9873$.

5. $\alpha=\ln\left(y^{-\sqrt{|x|}}\right)\left(x-\dfrac{y}{2}\right)+\sin^2\mathrm{arctg}(z)$.

   При $x=-15.246,\ y=4.642\times10^{-2},\ z=20.001\times10^2$ $\alpha=-182.036$.

6. $\beta=\sqrt{10\left(\sqrt[3]{x}+x^{y+2}\right)}\left(\arcsin^2 z-\left|x-y\right|\right)$.

   При $x=16.55\times10^{-3},\ y=-2.75,\ z=0.15$ $\beta=-38.902$.

7. $\gamma = 5\operatorname{arctg}(x) - \frac{1}{4}\arccos(x)\frac{x+3|x-y|+x^2}{|x-y|z+x^2}.$

При $x=0.1722, y=6.33, z=3.25\times10^{-4}$ $\gamma=-172.025.$

8. $\varphi = \frac{e^{|x-y|}|x-y|^{x+y}}{\operatorname{arctg}(x)+\operatorname{arctg}(z)} + \sqrt[3]{x^6+\ln^2 y}.$

При $x=-2.235\times10^{-2}, y=2.23, z=15.221$ $\varphi=39.374.$

9. $\psi = \left|x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}}\right| + (y-x)\frac{\cos y - \frac{z}{(y-x)}}{1+(y-x)^2}.$

При $x=1.825\times10^{2}, y=18.225, z=-3.298\times10^{-2}$ $\psi=1.2131.$

10. $a = 2^{-x}\sqrt{x+\sqrt[4]{|y|}}\sqrt[3]{e^{x-1/\sin z}}.$

При $x=3.981\times10^{-2}, y=-1.625\times10^{3}, z=0.512$ $a=1.26185.$

11. $b = y^{\sqrt[3]{|y|}} + \cos^3(y)\frac{|x-y|\left(1+\frac{\sin^2 z}{\sqrt{x+y}}\right)}{e^{|x-y|}+\frac{x}{2}}.$

При $x=6.251, y=0.827, z=25.001$ $b=0.7121.$

12. $c = 2^{(y^x)} + \left(3^x\right)^y - \frac{y\left(\operatorname{arctg} z - \frac{\pi}{6}\right)}{|x|+\frac{1}{y^2+1}}.$

При $x=3.251, y=0.325, z=0.466\times10^{-4}$ $c=4.025.$

13. $f = \frac{\sqrt[4]{y+\sqrt[3]{x-1}}}{|x-y|\left(\sin^2 z+\operatorname{tg} z\right)}.$

14. $g=\dfrac{y^{x+1}}{\sqrt[3]{|y-2|+3}}+\dfrac{x+\dfrac{y}{2}}{2|x+y|}(x+1)^{-1/\sin z}$.

При $x=12.3\times10^{-1}$, $y=15.4$, $z=0.252\times10^{3}$ $g=82.8257$.

15. $h=\dfrac{x^{y+1}+e^{y-1}}{1+x|y-\operatorname{tg}z|}(1+|y-x|)+\dfrac{|y-x|^{2}}{2}-\dfrac{|y-x|^{3}}{3}$.

При $x=2.444$, $y=0.869\times10^{-2}$, $z=-0.13\times10^{3}$ $h=-0.49871$.

16. $y=\sqrt{cx}-2.7\dfrac{|c|+|x|}{c^{2}x^{2}}\cdot e^{ax}+\cos\dfrac{(a+b)^{2}}{cx-b}$

$a=3.7;\ b=0.07;\ c=1.5;\ x=5.75$

17. $y=4.5\dfrac{(a+b)^{2}}{(a-b)^{2}}-\sqrt{(a+b)(a-b)}+10^{-1}\dfrac{\ln(a-b)}{\ln(a+b)}\cdot e^{x^{2}}$

$a=7.5;\ b=1.2;\ x=0.5$

18. $y=2.4\left|\dfrac{x^{2}+b}{a}\right|+(a-b)\sin^{2}(a-b)+10^{-2}(x-b)$

$a=5.1;\ b=0.7;\ x=-0.05$

19. $y=\dfrac{ax-\sqrt{b}}{5.7(x^{2}+b^{2})}-\dfrac{|x+b|-a^{2}}{x^{2}}\operatorname{tg}^{2}b$

$a=0.1;\ b=2.4;\ x=-0.3$

20. $y=\sqrt{\dfrac{c-dx^{2}}{x}}+\dfrac{\ln(x^{2}+c)}{0.7x+ad}-\dfrac{10^{-2}}{c-dx^{3}}$

$a=4.5;\ c=7.4;\ d=-2.1;\ x=0.15$

Лабораторное занятие № 3

Составление программ разветвляющейся структуры

По указанию преподавателя выберите индивидуальное задание из нижеприведенного списка. В качестве f(x) использовать по выбору: sh(x), x^2, e^x. Отредактируйте вид формы и текст программы, в соответствии с полученным заданием.

1. $a=\begin{cases}(f(x)+y)^2-\sqrt{f(x)y}, & xy\rangle 0\\ (f(x)+y)^2+\sqrt{|f(x)y|}, & xy\langle 0\\ (f(x)+y)^2+1, & xy=0.\end{cases}$

2. $b=\begin{cases}\ln(f(x))+(f(x)^2+y)^3, & x/y\rangle 0\\ \ln|f(x)/y|+(f(x)+y)^3, & x/y\langle 0\\ (f(x)^2+y)^3, & x=0\\ 0, & y=0.\end{cases}$

3. $c=\begin{cases}f(x)^2+y^2+\sin(y), & x-y=0\\ (f(x)-y)^2+\cos(y), & x-y\rangle 0\\ (y-f(x))^2+\operatorname{tg}(y), & x-y\langle 0.\end{cases}$

4. $d=\begin{cases}(f(x)-y)^3+\operatorname{arctg}(f(x)), & x\rangle y\\ (y-f(x))^3+\operatorname{arctg}(f(x)), & y\rangle x\\ (y+f(x))^3+0.5, & y=x.\end{cases}$

5. $e=\begin{cases}i\sqrt{f(x)}, & i-\text{нечетное},\ x\rangle 0\\ i/2\sqrt{|f(x)|}, & i-\text{четное},\ x\langle 0\\ \sqrt{|if(x)|}, & \text{иначе}.\end{cases}$

6. $g=\begin{cases}e^{f(x)-|b|}, & 0.5\langle xb\langle 10\\ \sqrt{|f(x)+b|}, & 0.1\langle xb\langle 0.5\\ 2f(x)^2, & \text{иначе}.\end{cases}$

7. $s=\begin{cases}e^{f(x)}, & 1\langle xb\langle 10\\ \sqrt{|f(x)+4*b|}, & 12\langle xb\langle 40\\ bf(x)^2, & \text{иначе}.\end{cases}$

8. $j=\begin{cases}\sin(5f(x)+3m|f(x)|), & -1\langle m\langle x\\ \cos(3f(x)+5m|f(x)|), & x\rangle m\\ (f(x)+m)^2, & x=m.\end{cases}$

9. $l=\begin{cases}2f(x)^3+3p^2, & x\rangle |p|\\ |f(x)-p|, & 3\langle x\langle |p|\\ (f(x)-p)^2, & x=|p|.\end{cases}$

10. $k=\begin{cases}\ln(|f(x)|+|q|), & |xq|\rangle 10\\ e^{f(x)+q}, & |xq|\langle 10\\ f(x)+q, & |xq|=10\end{cases}$

11. $m=\dfrac{\max(f(x),y,z)}{\min(f(x),y)}+5.$

12. $n=\dfrac{\min(f(x)+y,y-z)}{\max(f(x),y)}.$

13. $p=\dfrac{|\min(f(x),y)-\max(y,z)|}{2}.$

14. $q=\dfrac{\max(f(x)+y+z,xyz)}{\min(f(x)+y+z,xyz)}.$

15. $c=\begin{cases}f(\sin(x))^2+\sin(f(y)), & x-y=0\\ (f(\cos(x)))+\cos(f(y)), & x-y\rangle 0\\ (y-f(\operatorname{tg}(x)))^2+\operatorname{tg}(y), & x-y\langle 0.\end{cases}$

16. $a=\begin{cases}\dfrac{(ax^2+2)}{(x^2+1)}f(x), & 1<|x|<3,\\ a^2+f(x), & |x|\geq 3\\ ax\dfrac{f(x)}{(x+2)}, & |x|\leq 1.\end{cases}$

17. $c=\begin{cases}f(x)^3-y^3\cdot\cos(x), & x+y=0\\ (f(x)\cdot y)^2-\cos(y), & x+y\rangle 0\\ (y\cdot f(x))^2+\pi, & x+y\langle 0.\end{cases}$

18. $k=\begin{cases}\ln(|f(x^2)|+|k|), & |x\cdot k|\rangle 10\\ \pi^{f(x)+q}, & |x\cdot k|\langle 10\\ f(x)-k, & |x\cdot k|=10\end{cases}$

19. $c=\begin{cases}\sin(f(x))+\cos(f(y)), & x-y=0\\ \operatorname{tg}(f(x+y), & x-y\rangle 0\\ \sin^2(f(x))+\cos^2(f(y)), & x-y\langle 0.\end{cases}$

20. $r=\max(\min(f(x),y),z).$

Лабораторное занятие № 4

Составление программ циклической структуры

Составьте программу табулирования функции y(x), выведите на экран значения x и y(x). Нужный вариант задания выберите из нижеприведенного списка по указанию преподавателя. Откорректируйте элементы управления в форме в соответствии со своим вариантом задания.

1) $y = 10^{-2} bc / x + \cos\sqrt{a^3 x}$,
$x_0 = -1.5; x_k = 3.5; dx = 0.5;$
$a = -1.25; b = -1.5; c = 0.75;$

2) $y = 1.2(a\text{-}b)^3 e^{x^2} + x$,
$x_0 = -0.75; x_k = -1.5; dx = -0.05;$
$a = 1.5; b = 1.2;$

3) $y = 10^{-1} ax^3 tg(a - bx)$,
$x_0 = -0.5; x_k = 2.5; dx = 0.05;$
$a = 10.2; b = 1.25;$

4) $y = ax^3 + \cos^2(x^3 - b)$,
$x_0 = 5.3; x_k = 10.3; dx = 0.25;$
$a = 1.35; b = -6.25;$

5) $y = x^4 + \cos(2 + x^3 - d)$,
$x_0 = 4.6; x_k = 5.8; dx = 0.2;$
$d = 1.3;$

6) $y = x^2 + tg(5x + b / x)$,
$x_0 = -1.5; x_k = -2.5; dx = -0.5;$
$b = -0.8;$

7) $y = 9(x + 15\sqrt{x^3 + b^3})$,
$x_0 = -2.4; x_k = 1; dx = 0.2;$
$b = 2.5;$

8) $y = 9x^4 + \sin(57.2 + x)$,
$x_0 = -0.75; x_k = -2.05; dx = -0.2;$

9) $y = 0.0025bx^3 + \sqrt{x + e^{0.82}}$,
$x_0 = -1; x_k = 4; dx = 0.5;$
$b = 2.3;$

10) $y = x \cdot \sin(\sqrt{x + b - 0.0084})$,
$x_0 = -2.05; x_k = -3.05; dx = -0.2;$
$b = 3.4;$

11) $y = x + \sqrt{|x^3 + a - be^x|}$,
$x_0 = -4; x_k = -6.2; dx = -0.2;$
$a = 0.1;$

12) $y = 9(x^3 + b^3)tgx$,
$x_0 = 1; x_k = 2.2; dx = 0.2;$
$b = 3.2;$

13) $y = |x - b|^{1/2} / |b^3 - x^3|^{3/2} + \ln|x - b|$,
$x_0 = -0.73; x_k = -1.73; dx = -0.1;$
$b = -2;$

14) $y = (x^{5/2} - b)\ln(x^2 + 12.7)$,
$x_0 = 0.25; x_k = 5.2; dx = 0.3;$
$b = 0.8;$

15) $y = 10^{-3}|x|^{5/2} + \ln|x + b|$,
$x_0 = 1.75; x_k = -2.5; dx = -0.25;$
$b = 35.4;$

16) $y = 15.28|x|^{-3/2} + \cos(\ln|x| + b)$,
$x_0 = 1.23; x_k = -2.4; dx = -0.3;$
$b = 12.6;$

17) $y = 0.00084(\ln|x|^{5/4} + b)/(x^2 + 3.82)$, $x_0 = -2.35; x_k = -2; dx = 0.05;$ $b = 74.2;$

18) $y = 0.8 \cdot 10^{-5}(x^3 + b^3)^{7/6}$, $x_0 = -0.05; x_k = 0.15; dx = 0.01;$ $b = 6.74;$

19) $y = (\ln(\sin(x^3 + 0.0025)))^{3/2} + 0.8 \cdot 10^{-3}$, $x_0 = 0.12; x_k = 0.64; dx = 0.2;$

20) $y = a + x^{2/3}\cos(x + e^x)$, $x_0 = 5.62; x_k = 15.62; dx = 0.5;$ $a = 0.41$

Лабораторные занятия № 5-6
Обработка одномерных массивов
Обработка двумерных массивов

Задание 1 (индивидуальное).

1. В массиве из 20 целых чисел найти наибольший элемент и поменять его местами с первым элементом.

2. В массиве из 10 целых чисел найти наименьший элемент и поменять его местами с предпоследним элементом.

3. Дан массив F, содержащий 18 элементов. Вычислить и вывести элементы нового массива по формуле pi = 0.13fi 3 – 2.5fi + 8. Вывести отрицательные элементы массива P.

4. В массиве R, содержащем 25 элементов, заменить значения отрицательных элементов квадратами значений, значения положительных увеличить на 7, а нулевые значения оставить без изменения. Вывести массив R.

5. Дан массив A целых чисел, содержащий 30 элементов. Вычислить и вывести сумму тех элементов, которые кратны 5.

6. Дан массив A целых чисел, содержащий 30 элементов. Вычислить и вывести сумму тех элементов, которые нечетны и отрицательны.

7. Дан массив A целых чисел, содержащий 30 элементов. Вычислить и вывести количество и сумму тех элементов, которые делятся на 5 и не делятся на 7.

8. Дан массив A вещественных чисел, содержащий 25 элементов. Вычислить и вывести число отрицательных элементов и число членов, принадлежащих отрезку [1,2].

9. Дан массив Z целых чисел, содержащий 35 элементов. Вычислить и вывести R = S + P, где S – сумма четных элементов, меньших 3, P – произведение нечетных элементов, больших 1.

10. Дан массив Q натуральных чисел, содержащий 20 элементов. Найти и вывести те элементы, которые при делении на 7 дают остаток 1,2 или 5.

11. Дан массив, содержащий 10 элементов. Вычислить произведение элементов, стоящих после первого отрицательного элемента. Вывести исходный массив и результат вычислений.

12. Дан массив, содержащий 14 элементов. Вычислить сумму элементов, стоящих до первого отрицательного элемента. Вывести исходный массив и результат вычислений.

13. Дан массив, содержащий 12 элементов. Все четные элементы сложить, вывести массив и результат.

14. Дан массив, содержащий 15 элементов. Все положительные элементы возвести в квадрат, а отрицательные умножить на 2. Вывести исходный и полученный массив.

15. Дан массив, содержащий 14 элементов. Все отрицательные элементы заменить на 3. Вывести исходный и полученный массив.

16. Массив задан датчиком случайных чисел на интервале [–33, 66]. Найти наименьший нечетный элемент.

17. Разработать программу, выводящую количество максимальных элементов в массиве из пятидесяти целочисленных элементов.

18. Разработать программу, циклически сдвигающую элементы целочисленного массива влево. Нулевой элемент массива ставится на последнее место, остальные элементы сдвигаются влево на одну позицию. Запрещается использовать второй массив.

19. Дано два массива с неубывающими целыми числами. Напишите программу, формирующую новый массив из элементов первых двух. В результирующем массиве не должно быть одинаковых элементов.

20. Дан массив целых чисел из 30 элементов. Найдите все локальные максимумы. (Элемент является локальным максимумом, если он не имеет соседей, больших, чем он сам).

Задание 2 (индивидуальное).

1. Дана матрица A(3,4). Найти наименьший элемент в каждой строке матрицы. Вывести исходную матрицу и результаты вычислений.

2. Дана матрица A(3,3). Вычислить сумму второй строки и произведение первого столбца. Вывести исходную матрицу и результаты вычислений.

3. Вычислить сумму S элементов главной диагонали матрицы B(10,10). Если S>10, то исходную матрицу преобразовать по формуле bij = bij + 13.5; если S<=10, то bij = bij 2 – 1.5. Вывести сумму S и преобразованную матрицу.

4. Дана матрица F(15,15). Вывести номер и среднее арифметическое элементов строки, начинающейся с 1. Если такой строки нет, то вывести сообщение «Строки нет».

5. Дана матрица F(7,7). Найти наименьший элемент в каждом столбце. Вывести матрицу и найденные элементы.

6. Найти наибольший элемент главной диагонали матрицы A(15,15) и вывести всю строку, в которой он находится.

7. Найти наибольшие элементы каждой строки матрицы Z(16,16) поместить их на главную диагональ. Вывести полученную матрицу.

8. Найти наибольший элемент матрицы A(10,10) и записать нули в ту строку и столбец, где он находится. Вывести наибольший элемент, исходную и полученную матрицу.

9. Дана матрица R(9,9). Найти наименьший элемент в каждой строке и записать его на место первого элемента строки. Вывести исходную и полученную матрицы.

10. Вычислить количество H положительных элементов последнего столбца матрицы X(5,5). Если H < 3, то вывести все положительные элементы матрицы, если H>=3, то вывести сумму элементов главной диагонали матрицы.

11. Вычислить и вывести сумму элементов матрицы A(12,12), расположенных над главной диагональю матрицы.

12. Найти номер столбца матрицы, в котором находится наименьшее количество положительных элементов.

13. Дан двухмерный массив 20 × 20 целочисленных элементов. Найдите все

локальные максимумы. (Элемент является локальным максимумом, если он не имеет соседей, больших, чем он сам).

14. Дана матрица 7 × 7. Найти наибольший элемент среди стоящих на главной и побочной диагоналях и поменять его местами с элементом, стоящим на пересечении этих диагоналей.

15. Задана матрица, содержащая N строк и M столбцов. Седловой точкой этой матрицы назовем элемент, который одновременно является минимумом в своей строке и максимумом в своем столбце. Найдите количество седловых точек заданной матрицы.

16. Дана квадратная матрица 10 × 10. Реализуйте программу для транспонирования матрицы по главной и побочной диагоналям.

17. Требуется совершить обход квадратной матрицы по спирали так, как показано на рисунке: заполнение происходит с единицы из левого верхнего угла и заканчивается в центре числом N^2, где N – порядок матрицы. Реализуйте программу для матрицы 10 × 10.

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

18. Требуется заполнить змейкой квадратную матрицу так, как показано на рисунке: заполнение происходит с единицы из левого верхнего угла и заканчивается в правом нижнем числом N^2, где N – порядок матрицы. Реализуйте программу для матрицы 10 × 10.

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

19. Дана шахматная доска (матрица 8 × 8). Разработать программу, показывающую последовательность ходов конем с произвольной клетки. Конь ходит в соответствии с шахматными правилами, но в произвольную сторону (сгенерировать случайным образом). В клетку, с которой начинается ход, выводится единица. В клетку, в которую идет далее конь, записывается двойка и т. д. Ходить конем на клетки, на которых уже побывал конь, нельзя. Алгоритм останавливает работу, когда конем ходить некуда. Максимальная последовательность ходов – 64.

20. Проверка на симпатичность. Рассмотрим таблицу, содержащую n строк и m столбцов, в каждой клетке которой расположен ноль или единица. Назовем такую таблицу

симпатичной, если в ней нет ни одного квадрата 2 на 2, заполненного целиком нулями или целиком единицами. Так, например, таблица 4 на 4, расположенная слева, является симпатичной, а расположенная справа таблица 3 на 3 – не является.

1	0	1	0
1	1	1	0
0	1	0	1
0	0	0	0

0	0	1
0	0	1
1	1	1

Лабораторное занятие № 7
Работа со строками

Индивидуальное задание.

1. Дана строка, состоящая из групп нулей и единиц. Посчитать количество нулей и единиц.

2. Посчитать в строке количество слов.

3. Найти количество знаков препинания в исходной строке.

4. Дана строка символов. Вывести на экран цифры, содержащиеся в строке.

5. Дана строка символов. Сформировать новую строку, в которую включить все символы исходной строки, стоящие на четных местах. При этом должен быть обратный порядок следования символов по отношения к исходной строке.

6. Сформировать и вывести две новых строки на основе входной строки символов. В первую строку включить все символы, стоящие на четных местах, во вторую – символы, стоящие на нечетных местах в исходной строке.

7. Дана строка символов, состоящая из произвольных десятичных цифр, разделенных пробелами. Вывести количество четных чисел в этой строке.

8. Дана строка символов. Вывести на экран количество строчных русских букв, входящих в эту строку.

9. Сформировать и вывести три новых строки на основе входной строки символов. В первую строку включить все цифры, во вторую – все знаки препинания: точки, запятые, двоеточия, точки с запятой, восклицательные и вопросительные знаки, в третью строку – все остальные символы. Например, входная строка содержит: «выходные дни: 1, 2 января, 8 марта, 1 мая, 9 мая!», после обработки первая строка будет содержать: «12819», вторая строка: «:,,,!», третья строка: «выходные дни января марта мая».

10. Дана строка символов. Вывести на экран только строчные русские буквы, входящие в эту строку.

11. Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами. В каждом слове заменить первую букву на прописную.

12. Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами. Удалить первую букву в каждом слове.

13. Дана строка символов, состоящая из произвольного текста на английском

языке, слова разделены пробелами. Поменять местами i и j-ю буквы. Для ввода i и j на форме добавить свои поля ввода.

14. Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами. Заменить все буквы латинского алфавита на знак «+».

15. Дана строка символов, содержащая некоторый текст на русском языке. Заменить все большие буквы «А» на символ «*».

16. Дана строка символов, содержащая некоторый текст. Разработать программу, которая определяет, является ли данный текст палиндромом, т. е. читается ли он слева направо так же, как и справа налево (например, «А роза упала на лапу Азора»).

17. Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами. Сформировать новую строку, состоящую из чисел длин слов в исходной строке.

18. Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами. Поменять местами первую и последнюю буквы каждого слова.

19. Поменять местами первое и второе слово в исходной строке.

20. Сформировать новую строку, где поменять местами первое и последнее слово из исходной строки.

Лабораторное занятие № 8
Работа с данными типа множество

Индивидуальное задание.

1. Опишите множество Pr(1..20) и поместите в него все простые числа в диапазоне 1..20. Составьте блок-схему.

2. Опишите множество Alf(‘a’..’я’) и поместите в него гласные буквы. Составьте блок-схему.

3. Опишите множества M1(1,2) и M2(2,1). Сравните множества M1 и M2 на равенство. Составьте блок-схему.

4. Опишите множества M1(‘a’, ‘b’) M2(‘b’, ’a’, ‘c’). Сравните два этих множества на неравенство. Составьте блок-схему.

5. Опишите множества M1(‘a’, ‘b’,’c’) M2(‘a’, ‘c’). Сравните два этих множества с использованием операции >=. Составьте блок-схему.

6. Опишите множества M1(1, 2, 3) M2(1, 2, 3, 4). Сравните два этих множества с использованием операции <=. Составьте блок-схему.

7. Опишите множества M1(1, 2) M2(5, 6). Получите результирующее множество M3=M1-M2. Определите содержится в M3 элемент 7. Составьте блок-схему.

8. Опишите множества M1(1, 2, 3,4) M2(3, 4, 1). Получите результирующее множество M3=M1-M2. Определите содержится в M3 элемент 7. Составьте блок-схему.

9. Опишите множества M1(1, 2, 3) M2(1, 4, 2, 5). Получите результирующее множество M3=M1*M2. Определите содержится в M3 элемент 2. Составьте блок-схему.

10. Опишите множества M1(1, 2) M2(5, 6). Получите результирующее множество M3=M1+M2. Определите содержится в M3 элемент 2. Составьте блок-схему.

Лабораторные занятия № 9-10
Файлы последовательного доступа.

Типизированные файлы.

Индивидуальное задание.

1. Составить программу с тремя функциями, выводящими на экран строки символов #, @, $.

2. Составить программу для вычисления площади кольца по значениям внутреннего и внешнего радиусов, используя функцию вычисления площади круга.

3. Даны три целых числа. Определить, сумма цифр которого из них больше. Подсчет суммы цифр организовать через функцию.

4. Создать типизированный файл, содержащий данные о студентах группы: фамилия и инициалы (одно поле записи), год рождения, адрес (улица, дом, квартира), средний балл при поступлении. Переписать в текстовый и вывести в ячейки эл. таблицы данные о студентах со средним проходным баллом, большим 3,8.

5. Создать типизированный файл, содержащий данные о редких книгах, хранящихся с библиотеке: название, автор (фамилия и инициалы), год издания, место издания, инвентарный номер. Переписать в текстовый файл и вывести в ячейки эл. таблицы данные о книгах, изданных в Санкт-Петербурге;

6. Создать типизированный файл, содержащий данные о рейсах самолетов: номер рейса, пункт назначения, время в пути (дробное число), тип самолета, время отправления (два поля записи: часы и минуты). Переписать в текстовый файл и вывести в ячейки эл. таблицы данные о рейсах, вылетающих после десяти вечера, но до полуночи;

7. Создать типизированный файл, содержащий данные о автобусных маршрутах: пункт отправления, пункт назначения, время в пути (дробное число), время отправления (два поля записи: часы и минуты), количество остановок в пути. Переписать в текстовый файл и вывести в ячейки эл. таблицы данные о рейсах, делающих более трех остановок в пути.

8. Создать типизированный файл, содержащий данные о наименованиях продукции молокозавода, поступивших в продажу: название продукта, дата изготовления (три поля записи: год, месяц и число), срок хранения в днях, закупочная цена (дробное число). Переписать в текстовый файл и вывести в ячейки эл. таблицы данные о продуктах, выпущенный в июле 2004 года.

9. Создать типизированный файл, содержащий данные о фирмах: название фирмы, фамилия и инициалы владельца (одно поле), адрес (три поля: город, улица, дом), телефон. Переписать в текстовый файл и вывести в ячейки эл. таблицы данные о владельцах фирм, фамилия которых начинается с буквы М.

10. Создать типизированный файл, содержащий данные о автомашинах, находящихся в розыске: марка, цвет, год выпуска, номер, дата угона (три поля: число, месяц, год). Переписать в текстовый файл и вывести в ячейки эл. таблицы данные о машинах, угнанных в июле 2002 года;

11. Создать типизированный файл, содержащий данные о сотрудниках фирмы: фамилия, имя, отчество, год рождения, должность, год поступления на работу. Переписать в текстовый файл и вывести в ячейки эл. таблицы данные о сотрудниках фирмы, устроившихся на работу в период с 1999 по 2002 год включительно.

12. Создать типизированный файл, содержащий данные о спортсменах-пловцах: фамилия и инициалы, пол, год рождения, рост, вес, лучшее время, за которое спортсмен проплывает 50 метров. Переписать в текстовый файл и вывести в ячейки эл. таблицы

данные о спортсменах, рост которых больше 175 см.

13. Создать типизированный файл, содержащий данные о клиентах ателье: фамилия и инициалы, адрес (три поля: улица, дом, квартира), вид заказа, стоимость заказа. Переписать в текстовый файл и вывести в ячейки эл. таблицы данные о клиентах, проживающих на улице Международная.

1. Составить программу с функцией, выводящей на экран строку нужной длины нужным символом. Функция принимает два параметры – длину строки и символ. Возвращаемых значений нет.

2. Составить программу для вычисления площади кольца по значениям внутреннего и внешнего радиусов, используя функцию вычисления площади круга. Функции передается один параметр – радиус и возвращается результат – вычисленная площадь.

3. Даны три целых числа. Определить, сумма цифр которого из них больше. Подсчет суммы цифр организовать через функцию, которая имеет два входных параметра – складываемые числа и результат – вычисленную сумму

Лабораторные занятия № 12-13
Организация процедур.
Организация функций.

Тип string, предназначенный для работы со строками символов в кодировке Unicode, является встроенным типом C#. Ему соответствует базовый класс System.String библиотеки .Net.

Тип string относится к ссылочным типам, хотя работа с ним во многом напоминает работу с размерными типами. С объектом типа string можно работать посимвольно, т.е. поэлементно.

Класс string обладает богатым набором методов для сравнения строк, поиска в строке и других действий со строками. Все методы возвращают ссылку на новую строку, созданную в результате преобразования копии исходной строки. Для того чтобы сохранить данное преобразование, нужно установить на него новую ссылку. Область применения типа string – это поиск, сравнение, извлечение информации из строки.

Создайте новое консольное приложение с именем Zad10. Составьте программу для нахождения количества вхождений символа в строке и протестируйте её.

```
static void Main(string[] args)
{
    string a = "кол около колокола";
    Console.WriteLine("Дана строка: {0}", a);
    char b = 'о';
    int k = 0;
    for (int x=0; x<a.Length; x++)
    {
        if (a[x]==b)
        {
            k++;
        }
    }
    Console.WriteLine("Символ {0} содержится в ней {1} раз", b, k);
}
```

Рисунок 1 – Код программы

Попробуем заменить все вхождения буквы о на букву а.

```
static void Main(string[] args)
{
    string a = "кол около колокола";
    Console.WriteLine("Дана строка: {0}", a);
    char b = 'o';
    char c = 'a';
    int k = 0;
    for (int x=0; x<a.Length; x++)
    {
        if (a[x]==b)
        {
            a[x]=c;
        }
    }
    Console.WriteLine("Символ {0} содержится в ней {1} раз", b, k);
}
```

Рисунок 2 – Код программы

Запустить программу будет не возможно. Компилятор запрещает напрямую изменять значение строки.

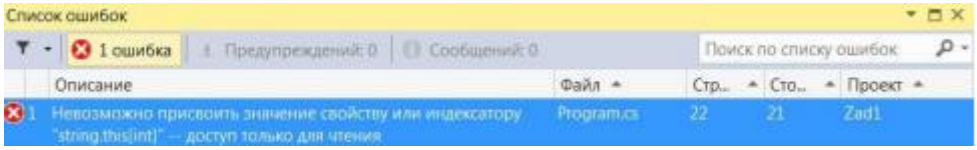


Рисунок 3 – Ошибка

Строковый тип stringbuilder определен в пространстве имен System.Text и предназначен для создания строк, значение которых можно изменять. Объекты данного класса всегда создаются с помощью явного вызова конструктора класса, т.е. Через операцию new. С объектами класса stringbuilder можно работать посимвольно. Внесите изменения в созданную ранее программу следующим образом:

```
static void Main(string[] args)
{
    StringBuilder a = new StringBuilder ("кол около колокола");
    Console.WriteLine("Дана строка: {0}", a);
    char b = 'o';
    char c = 'a';
    int k = 0;
    for (int x=0; x<a.Length; x++)
    {
        if (a[x]==b)
        {
            k++;
        }
    }
    Console.WriteLine("Символ {0} содержится в ней {1} раз", b, k);
}
```

Рисунок 4 – Код программы

Протестируйте работу программы. Она ничем не отличается от аналогичной программы с типом string.

Измените программу так, чтобы все вхождения буквы о были заменены на букву а. Возникает ли ошибка при работе данной программы:

```
static void Main(string[] args)
{
    StringBuilder a = new StringBuilder ("кол около колокола");
    Console.WriteLine("Дана строка: {0}", a);
    char b = 'o';
    char c = 'a';
    int k = 0;
    for (int x=0; x<a.Length; x++)
    {
        if (a[x]==b)
        {
            a[x]=c;
        }
    }
    Console.WriteLine("Преобразованная строка: {0}", a);
}
```

Рисунок 5 – Код программы

Завершите работу с программой.

Лабораторное занятие № 14
Применение рекурсивных функций

1. Разработать рекурсивную функцию вычисления факториала.

2. Разработать рекурсивную функцию, которая для двух заданных натуральных чисел определяет их наибольший общий делитель.

3. Разработать рекурсивную функцию, определяющую число Фибоначчи с номером n.

Лабораторное занятие № 15
Программирование модуля

1. Напишите рекурсивную функцию, определяющую количество единиц в двоичном представлении натурального числа.

2. Напишите рекурсивную функцию, которая по заданным натуральным числам m и n выводит все различные представления числа «в виде суммы m натуральных слагаемых». Представления, различающиеся лишь порядком слагаемых, считаются одинаковыми.

3. Напишите рекурсивную функцию, которая переворачивает элементы массива в обратном порядке.

Лабораторное занятие № 16
Создание библиотеки подпрограмм

Открываем visual studio. Выбираем: “Создать новый проект”, в открывшемся окне выбираем “консольное приложение Widows 32”, также вводим название проекта mynewlib и указываем путь где мы сохраним проект.

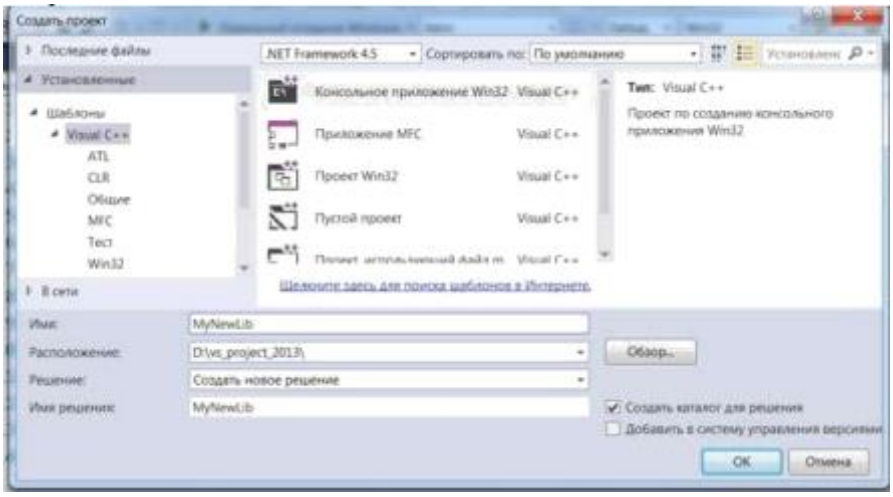


Рисунок 1 – Создание проекта

Нажимаем ОК и мы переходим к следующему окошку.

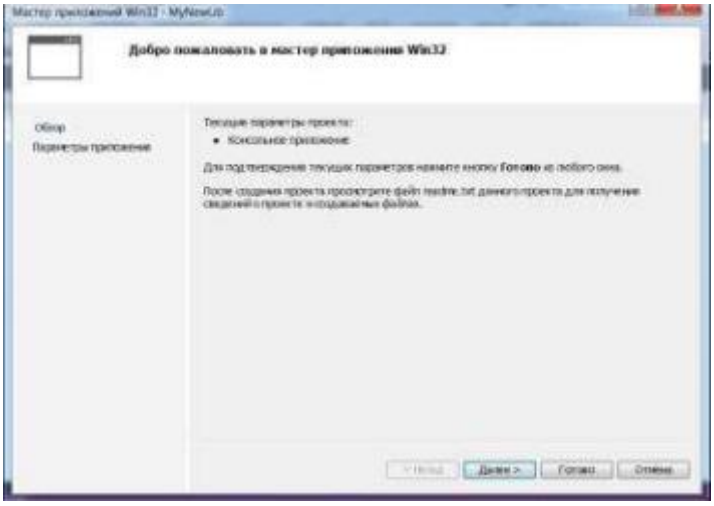


Рисунок 2 – Создание проекта: этап обработки

В новом окне выбираем “Статическая библиотека” и убираем галочку “предварительно откомпилированные заголовок”, далее нажимаем «готово».

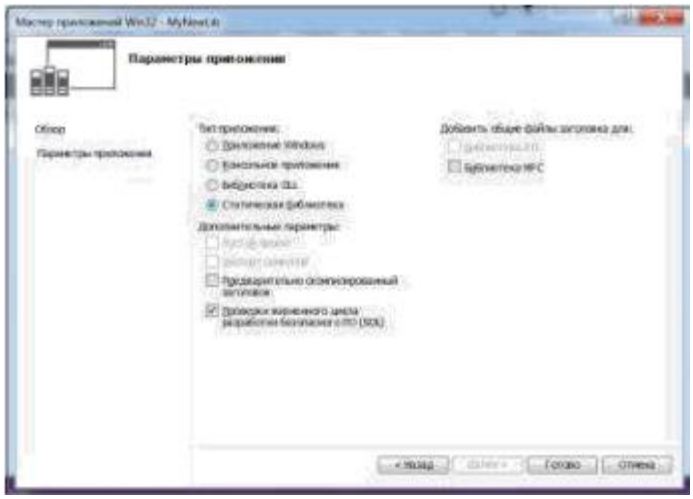


Рисунок 3 – Выбор библиотеки

Таким образом, у нас создается новый проект в котором будут создаваться не программы exe а статические библиотеки lib.

Дальше в этот проект mynewlib добавляем два файла один заголовочный function.h:

```
#ifndef FUNCTION_H
#define FUNCTION_H
Void func_hellow();
Void func_privet();
#endif
Файл function.cpp:
#include <iostream>
Using std::cout;
Using std::endl;
#include "function.h";
Void func_hellow()
{
Cout <<"Hellow world gacpada"<<endl;
}
Void func_privet()
{
Cout <<"privet eb tu bl9 mazafaka bich"<<endl;
```

Включаем их в проект, а дальше компилируем нажимаем клавиши ctrl+alt+F7 и у нас создается в папке Debug или Release смотря какой режим выбрать. (мы выбраем режим Debug) и в папке Debug создался файл библиотеки mynewlib.lib

Для того, чтобы создавать качественные библиотеки нам нужно каждую функцию записать в отдельный файл, это для того, чтобы в mynewlib.lib было создано несколько объектных файлов .obj в данном случае мы могли бы файл function.cpp разбить на два файла допустим файл f1.cpp в котором записать определение функции func_hellow() и на файл f2.cpp в нем записать определение второй функции func_privet(), тогда у нас будет правильная библиотека и если мы в проекте будем использовать одну функцию, допустим func_hellow(), а func_privet() нет, то объектный код .obj с функцией func_privet() не добавиться , а только .obj func_hellow() будет добавляться, это фажный момент, возможно ниже попробуем смоделировать два разных варианта создания библиотек. Просто при использовании библиотеки в которой каждой функции принадлежит отдельный объектный код, только включаются те функции которые используются, а если все функции в одном .obj то естественно включатся все функции и даже те которые мы не используем.

Нужно подключить библиотеку к нашему проекту, для этого:

- создаем новый проект пустой, проект будет пусть консольный вин 32 и создаем в нем файл Main.cpp со следующим кодом:

```
#include <iostream>
Using std::cout;
Using std::endl;
#include "function.h"
Int main()
{
Func_hellow();
Func_privet();
Return 0;
}
```

Таким образом, мы включили в проект заголовочный файл function.h в котором находятся определения функций.

Вызываем функции, (если мы сейчас это все скомпилируем, то у нас вылезет ошибка так как мы реально не добавили файл function.h и не добавили саму библиотеку в проект)

- Копируем файлы mynewlib.lib и файл function.h в папку с файлами нашего проекта – это та папка где находится наш файл main.cpp.

- С помощью горячих клавиш Shift+Alt+A

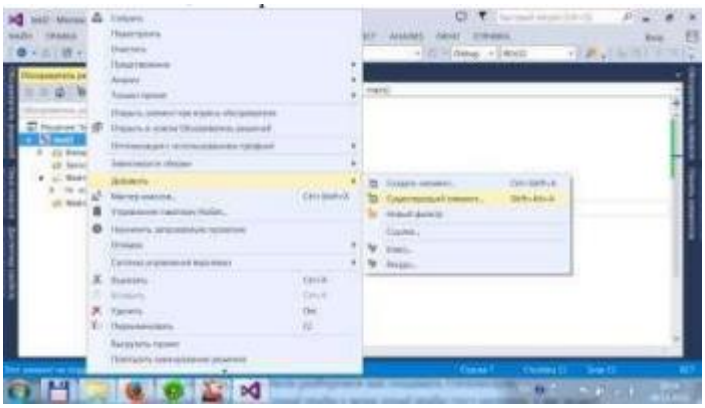


Рисунок 4 – Вызов функции

Дальше появится окно.

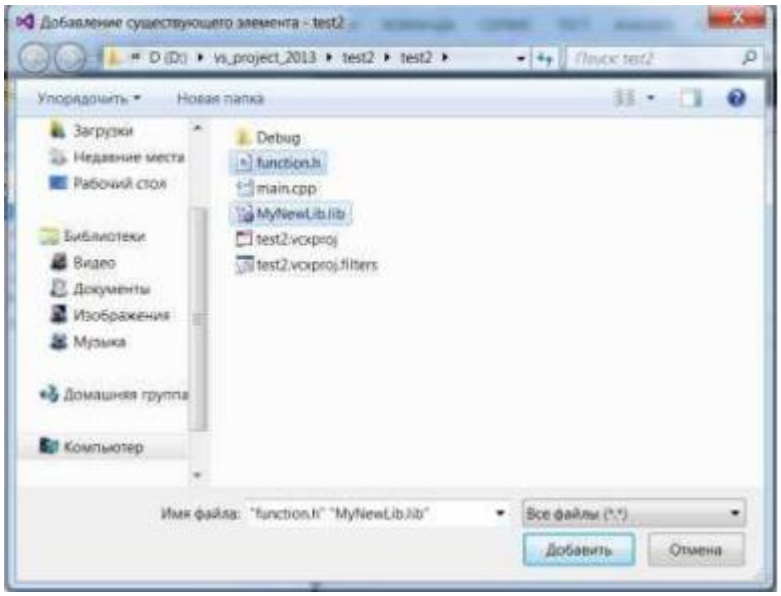


Рисунок 5 – Выбор функции

Выбираете файлы и нажимаете добавить и все файлы добавлены, при открытии обозревателя решений у вас они должны быть видны из него.

Статическая библиотека mynewlib.lib добавлена и заголовочный файл с определениями функций из библиотеки mynewlib.lib включен это файл function.h, теперь нажимаем Ctrl+F5 и все компилируется.

Лабораторное занятие № 17

Использование указателей для организации связанных списков

Индивидуальное задание.

1. Разработать программу, динамически порождающую на окне кнопки. Левый верхний угол кнопки определяется местоположением курсора при щелчке. Вывести надпись на кнопке с координатами ее левого верхнего угла.

2. Разработать программу, динамически порождающую на окне кноп-ки и поля ввода. Левый верхний угол элемента управления определяется местоположением курсора

при щелчке. Кнопка порождается, если курсор находится в левой половине окна, в ином случае порождается поле ввода.

3. На форме размещен элемент управления Panel. Написать про-грамму, которая при щелчке мыши на элементе управления Panel добавляет в него кнопки Button, а при щелчке на форме в нее добавляются поля ввода textbox.

4. На форме размещены 3 панели (элемент управления Panel). Написать программу, которая при щелчке мыши на первой панели добавляет во вторую панель кнопки Button, при щелчке на второй панели добавляет в третью панель поля ввода textbox, а при щелчке на третьей панели добавляет на первую панель метки Label. и Написать программу, добавляющую на форму кнопки. Кнопки добавляются в узлы прямоугольной сетки. Расстояния между кнопками и расстояния между крайней кнопкой и границей окна должны быть равны как по горизонтали, так и по вертикали.

6. Разработать программу, при щелчке мыши динамически порождающую на окне кнопки или поля ввода. Каждый четный элемент управления является кнопкой, нечетный – полем ввода. Левый верхний угол кнопки определяется местоположением курсора при щелчке. Для поля ввода положение курсора определяет координаты правого нижнего угла.

7. Создать программу с кнопкой, меткой и полем ввода. При щелчке на соответствующий элемент на форме динамически должен создаваться подобный ему элемент. Предусмотреть возможность вывода количества кнопок, меток и полей ввода.

8. Создать программу, добавляющую различные элементы управления на форму и на панель Panel. Тип элементов управления выбирается случайным образом. Предусмотреть возможность вывода информации о количестве элементов по типам и информацию о расположении элементов.

9. Разработать программу, добавляющую на форму последовательность элементов управления случайной длины. Тип элементов управления задается случайным образом. Предусмотреть возможность вывода информации о количестве элементов по типам.

10. Написать программу, динамически порождающую на окне кнопки или метки. Левый верхний угол элемента управления определяется местоположением курсора при щелчке. При нажатии правой кнопки мыши на форме с нее удаляются все кнопки.

11. Написать программу, динамически порождающую на окне поочередно кнопки или поля ввода. Левый верхний угол элемента управления определяется местоположением курсора при щелчке. При нажатии правой кнопки мыши на форме с нее удаляются все порожденные элементы.

12. Разработать программу с двумя кнопками на форме. При нажатии на первую на форму добавляется одна панель Panel. При нажатии на вторую кнопку в каждую панель добавляется поле ввода.

13. Разработать программу с двумя кнопками на форме. При нажатии на первую на форму добавляется одна кнопка или поле ввода. При нажатии на вторую кнопку каждое поле увеличивается по вертикали в два раза.

14. Написать программу с кнопкой и тремя полями ввода. При нажатии на кнопку программа анализирует содержимое первого поля и динамически порождает элемент управления. Если в первом поле ввода содержится буква «К», то на форму добавляется кнопка, если «П» – поле ввода, если «М» – метка. Во втором и третьем поле ввода содержатся координаты левого верхнего угла будущего элемента управления.

15. Разработать программу, добавляющую на форму метки с текстом. Местоположение и размеры меток определяются в программе динамически через поля ввода. В заголовок окна, анализируя размер всех меток, вывести количество маленьких и больших меток. Маленькой меткой считается метка размером менее 50 пикселей по горизонтали и вертикали.

16. Создать программу с двумя кнопками на форме, динамически порождающую на окне метки или поля ввода. При нажатии на первую кнопку каждая метка увеличивается по горизонтали в два раза. При нажатии на вторую кнопку каждое поле уменьшается по вертикали в два раза.

17. Разработать программу, динамически порождающую на окне кнопки и поля ввода. Координаты элемента управления определяются случайным образом. Элементы управления не должны накладываться друг на друга. Если нет возможности добавить элемент управления (нет места для размещения элемента), то предусмотреть вывод информации об этом.

18. Разработать программу, динамически порождающую на окне кнопки и поля ввода. Координаты элемента управления определяются случайным образом. При наведении курсора на элемент управления он должен быть удален с формы.

19. Разработать программу, динамически порождающую при щелчке на окне различные элементы (поля ввода, кнопки, метки). Тип элементов определяется с помощью радиокнопок. Все элементы располагаются горизонтально в ряд. При достижении правой границы окна начинается новый ряд элементов.

20. Разработать программу, динамически порождающую или поле ввода (при нажатии на окне левой кнопкой мыши), или кнопку (при нажатии на окне правой кнопкой мыши). Все элементы располагаются наискосок, начиная с левого верхнего угла окна. Реализовать обработчик события изменения размера окна, в котором удалить все порожденные элементы.

Лабораторное занятие № 1
Изучение интегрированной среды разработчика

Необходимо освоить основные этапы и действия со средой на примере создания простейшего приложения. Для этого надо создать отдельную папку, в которую будут помещаться все файлы, относящиеся к одному приложения.

1. Создать простейшее приложение C++ Builder в отдельной папке. Для файлов приложения подобрать подходящие имена. Просмотреть размер получившего приложения командой: Project/Information for Project. Примерный размер простейшего приложения – 25 Кбайт.

2. Просмотреть содержимое и изучить структуру следующих файлов:

- Файла модуля: Unit1.cpp; при необходимости переключиться на текст клавишей F12;
- Файла проекта Project1.cpp командой: Project/View Source; откроется отдельная закладка в окне Редактора;
- Файла формы Unit1.dmf командой локального меню: View as Text; вернуться назад в представление формы как окна можно командой локального меню: View as Form.

Временно выйти из среды и просмотреть содержимое папки. Изучить состав

образованных файлов приложения и их назначение.

3. Задание иконки разработанного приложения. Для этого требуется выполнить два этапа:

1. Создание иконки (файла с расширением .ico). Если файл с таким расширением есть, то этот этап можно опустить. Для создания иконки можно воспользоваться встроенным в среду Графическим редактором (вызвать графический редактор командой: Tools/Image Iditor; выбрать в нем пункт меню File/New/Icon File (ico); задать свойства иконки: размер 32x32, цвет=16; рисовать иконку по пикселям; сохранить файл иконки в папке с файлами проекта командой: File/Save As…; выйти из Image Iditor)

2. Подключение иконки к приложению (выполнить команду: Project/Options…; в открывшемся окне перейти на закладку Application; выбрать иконку кнопкой Load Icon…; в появившемся окне выбрать путь к файлу иконки; нажать кнопку Ok).

Лабораторное занятие № 19

Создание проекта с использованием компонентов для работы с текстом

1. Разработать программу, которая при вводе текста в первый компонент Edit1, во втором компоненте Edit2 отображает реальную длину вводимой строки. Кроме этого, при выходе из компонента Edit1 его содержимое копируется в буфер обмена и удаляется, а при возвращении в программу появляется снова.

2. Разработать программу, которая запрещает ввод в компонент Edit1 подряд двух одинаковых символов.

3. Разработать программу, которая считает количество нажатий на кнопку и выдает это значение в компоненте Edit.

4. Разработать программу, которая считывает строку под определенным номером и помещает её в текстовое поле.

5. Разработать программу, которая сохраняет текст, набранный в поле Memo в файл, имя которого задано в текстовом поле Edit.

Лабораторное занятие № 20

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

1. Разработать приложение «Калькулятор» с использованием изученных элементов интерфейса. Создать программный код разделяемых событий для кнопок с цифрами и знаками операций. Приложение должно содержать проверки возможных ошибок при вводе исходных значений. В программе использовать две глобальные вещественные переменные, в которые будут сохраняться операнды, над которыми будут выполняться действия, а так же символьную переменную для хранения символа выбранной операции.

2. Создать простейший вид калькулятора. Калькулятор должен иметь панель для отображения числа, десять цифровых кнопок, кнопку разделителя целой и дробной частей, кнопку «=», кнопку смены знака и четыре кнопки с основными математическими действиями

3. Дополнительно к разработанному ранее калькулятору добавляются кнопки памяти.

4. Разработать инженерный калькулятор с вычислением стандартных математических функций: квадратного корня, возведения в квадрат, тригонометрических

функций, возведения в степень и другие.

Лабораторное занятие № 21

События компонентов (элементов управления), их сущность и назначение

Необходимо освоить основные свойства формы и способы задания им значений в требуемом диапазоне, а так же освоить основные приемы создания обработчиков событий формы.

Для этого надо создать отдельную папку, в которую будут помещаться все файлы, относящиеся к одному приложения. Установить следующие свойства формы с использованием окна Инспектора объектов:

1. Выравнивание формы различными вариантами (свойство Align);
2. Оформить рамку формы различными вариантами (свойство borderstyte);
3. Задать заголовок окна формы (свойству Caption присвоит значение «Моя первая форма»);
4. Подобрать цвет формы (свойство Color);
5. Выбрать вид указателя формы из списка стандартных указателей (свойство Cursor);
6. Поэкспериментировать со свойством прозрачности формы (свойства alphablend и alphablendvalue);
7. Изображение всплывающей подсказки формы (в свойство Hint записать текст подсказки – «Это форма», в свойство showhint записать значение true);
8. Указать состояние формы (свойство windowstaty);
9. Создать иконку (значок) для формы. Для этого надо выполнить два этапа:

Создание иконки. Для этого требуется выполнить:

- вызвать графический редактор командой Tools\Image Iditor.
- выбрать в нем пункт меню File\New\Icon File(ico);
- задать свойства иконки (размер 32x32, цвет=16);
- рисовать иконку по пикселям;
- сохранить файл иконки в папке со всеми файлами данного проекта командой File\Save As…; выйти из Image Editor.

Подключение иконки к форме. Для этого надо выполнить:

- выбрать свойство Icon формы, дважды щелкаем напротив этого свойства на кнопке […]. При этом появляется окно Picture Editor;
- щелкнуть по кнопке Load…. Появиться окно Load Picture. Выбрать в нем нужную иконку;
- щелкнуть по кнопке «Открыть» в окне «Load Picture». При этом происходит возврат в окно Picture Editor, в котором высвечивается нужная иконка;
- щелкнуть по кнопке Ок. Нужная иконка появиться в заголовке.

Сохранить все файлы проекта командой меню File\Save All.

10. В событии onmousedown левая кнопка мыши случайно меняет цвет формы, правая кнопка мыши сворачивает окна приложения до иконки.

11. Событие ondblclick вызывает закрытие формы методом Close, а так как форма является главной, то ее закрытие завершает работу всего приложения.

12. Событие onmousemove выводит текущие координаты мыши в заголовке формы.

13. В событии oncreate активизировать датчик случайных чисел.

14. В событии onkeydown при нажатии клавиш [Alt]+[X] осуществить выход из приложения. Событие onkeydown анализирует код нажатой клавиши и выполняет необходимые действия для формы в зависимости от нажатой клавиши, а именно:

- при нажатии клавиш управления курсором форма должна перемещаться в указанном направлении (свойства Left и Top для формы);
- при одновременном нажатии клавиши [Shift] и клавиш управления курсором форма должна увеличиваться в размерах в нужном направлении (свойства Left, Top, Width, Height для формы);
- при нажатии клавиши [Ctrl] и клавиш управления курсором, происходит уменьшении размер формы в нужном направлении.

События onmousewheeldown и onmousewheelup меняют прозрачность формы (свойстваalphablend и alphablendvalue для формы).

Лабораторное занятие № 22
Создание процедур на основе событий

1. Описать две глобальные целочисленные переменные для счетчика секунд и минут секундомера.

2. Расположить на форме четыре панели для отображения текущего времени, даты, название дня недели и панель с секундомером. Подобрать цвета, стиль и расположение панелей на форме.

3. В любом месте формы расположить два компонента Timer: один для получения текущего времени, другой для отображения работы секундомера. Задать для обоих компонентов свойства Interval в значение 1000.

4. Добавить на форму две кнопки: одну для выхода из приложения, другую для включения/выключения секундомера.

5. Написать программный код события oncreate для формы, в котором:

- включить таймер для текущего времени (свойство Enabled);
- отключить таймер для секундомера;
- обнулить значения счетчиков для секундомера;
- кнопке задать заголовок «Включить»;
- в панель для секундомера вывести значения счетчиков с учетом впереди стоящего нуля, т.е. В свойство Caption строку «00:00»;
- отобразить на панелях текущую дату, день недели и временя.

6. Написать программный код для события onclick командной кнопки:

Проверить значение свойства Caption этой кнопки: если там значение «Включить», то поменять его на «Выключить» и наоборот;

7. Написать программный код для события ontimer компонента Timer1, в котором обновлять показания в панелях текущей даты, дня недели и времени.

8. Написать программный код для события ontimer компонента Timer2, в котором по названию на кнопке выводить или обнулять счетчики таймера..

9. Добавить на форму кнопку «Pause». Написать для нее программный код события onclick, в котором останавливать или продолжать работу таймера Timer2 (менять свойство Enabled компонента Timer2 на противоположное).

10. Добавить на форму две графические кнопки со стрелками. Написать программных код событий onclick для них, в которых увеличивать (стрелка вверх) или

уменьшать (стрелка вниз) на 100 значение свойства Interval компонента Timer2, что повлияет на частоту срабатывания события ontimer. Проверять на допустимые значения свойства Interval.

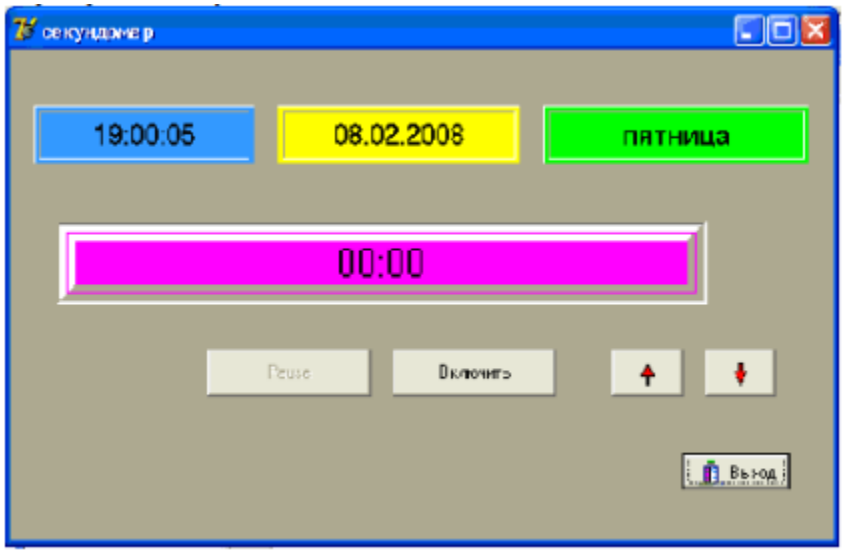


Рисунок 1 – Секундомер

Лабораторное занятие № 23

Создание проекта с использованием кнопочных компонентов

Разработать приложение «Конвертор величин измерения» с использованием компонента ввода строки Edit. Создать программный код необходимых событий.

В событии onkeypress компонента Edit выполнять контроль за вводом информации. Щелчком по кнопке осуществлять перевод величин в одном направлении. Выделять цветом и шрифтом активный компонент Edit.

Добавить кнопку для перевода информации в обратном направлении. Так же проводить контроль за правильностью ввода.

Разместить на форме кнопку «Сохранять» и после каждого выполненного конвертирования, результаты помещать в текстовый файл в форме:

Время	Операция	Результат

Варианты индивидуальных заданий

1. Перевод из рублей в доллары
2. Перевод из рублей в евро
3. Перевод из чисел между 16-ричной и 10-тичной системами счисления.
4. Переводы температур между Цельсием и Фаренгейту: $t_c = \frac{5(t_f - 32)}{9}$
5. Переводы из чисел между 2-ичной и 10-тичной системами счисления
6. Переводы углов между градусами и радианами
7. Перевод из чисел между 16-ричной и 2-ичной системами счисления
8. Переводы секунд в часы, минуты и секунды
9. Переводы длин между верстами и метрами (1 верста=1066,8 метра)
10. Переводы длин между саженями и метрами (1 саженьа=2,1336 метра)
11. Переводы длин между дюймами и сантиметрами (1дюйм=2,54сантиметра)
12. Переводы длин между морскими милями и метрами (1 м.м.=1852,2 метра)

13. Переводы весов между фунтами и килограммами (1 фунт=0,40951 кг)
14. Переводы скоростей между км/ч и м/с
15. Переводы весов между пудами и килограммами (1 пуд=16,380 кг)
16. Переводы для кругов площадей в радиусы

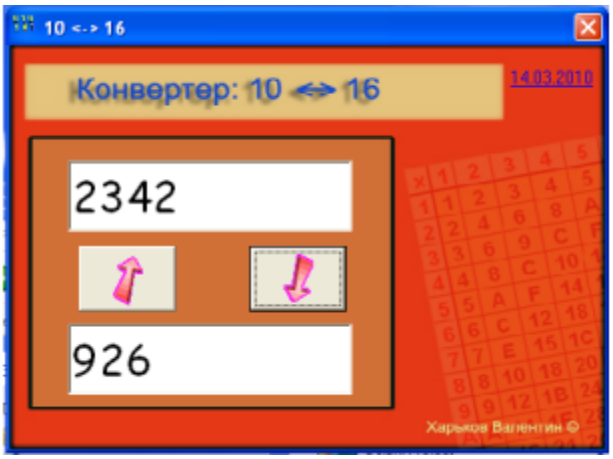


Рисунок 1 – Приложение «Конвертер единиц измерения»

Лабораторное занятие № 24

Создание проекта с использованием компонентов стандартных диалогов и системы меню.

1. Расположить на форме: панель, которая должна занимать большую часть формы – «Минное поле»; на ней кнопку размером 50x50 с картинкой мины; в свободном пространстве формы четыре метки с именами: labelx, labely, labeld, Labels (свойство Name) для отображения текущего положения на поле.

2. Описать две глобальные целые переменные d_old (предыдущее расстояние до кнопки) и d_new (новое расстояние до кнопки).

3. Создать функцию rast() для класса формы tform1, в которой вычисляется расстояние между серединой кнопки и текущим положением курсора мыши. Прототип функции задать в файле .h в описании класса формы раздела public. Описание и вызов член-функции разместить в файле .cpp. Как параметры функции передаются координаты мыши, возвращается вычисленное расстояние. Функции может выглядеть следующим образом:

```
Double tform1::rast (int x, int y)
{       int xc, yc;
Xc=Form1->Button1->Left+Form1->Button1->Width/2;
Yc=Form1->Button1->Top+Form1->Button1->Height/2;
Return (sqrt(pow((x-xc),2)+pow((y-yc),2)));
}
```

4. В событии onshow для формы организовать работу с паролем. Считать подготовленный заранее пароль из текстового файла «Key.txt» (создается заранее в Блокноте или из среды Builder командой File\New\Other\Text). Работу с паролем оформить через команды диалога ввода-вывода.

5. В событии oncreate для формы реализовать действия:

- запустить датчик случайных чисел;
- задать случайное положение кнопки (свойства Left, Top) в диапазоне размера

панели минус ширина кнопки;

- сделать кнопку невидимой (свойство visible);

- задать начальное значение переменной d_old, равное ширине панели.

6. В событии onmousemove для панели реализовать действия:

- вывести в надписи labelx и labely текущее положение курсора мыши на панели;

- в надпись labeld разместить значение переменной d_new, полученное с помощью функции вычисления расстояния;

- сравнить два расстояния d_new и d_old и вывести сообщение «Ближе» или «Дальше» в компонент надпись Labels;

- перезаписать в значение переменной d_old текущее расстояние d_new для следующих перемещений и сравнений;

- если курсор мыши, вблизи кнопки, то сделать кнопку видимой.

7. Щелчком по кнопке (событие onclick кнопки) сделать кнопку невидимой, задав ей новые случайные координаты.

8. После каждого обнаружения кнопки («Мины»), уменьшать ее размер на 10 пикселей. Осуществить данное действие в событии onclick формы.

9. По желанию пользователя сменить пароль, перезаписав его в текстовый файл. Осуществить данное действие в событии onclose формы.

10. Добавить счетчик количества попыток обнаружения кнопки. Проверять его с заданным общим количеством попыток. Выводить на метку информацию об оставшихся количествах попыток.

11. Добавить компонент Timer. Использовать его для задания времени игрового процесса.

12. В событии ontimer выводить на метку количество оставшегося игрового времени.

13. Остановить игровой процесс в случае определения победителя. Выигрывает пользователь в случае обнаружения десяти «Мин», выигрывает ПК, если время игры истекло, а все «Мины» не найдены.

Лабораторное занятие № 25
Разработка функциональной схемы работы приложения

Необходимо освоить основные свойства формы и способы задания им значений в требуемом диапазоне, а так же освоить основные приемы создания обработчиков событий формы.

Для этого надо создать отдельную папку, в которую будут помещаться все файлы, относящиеся к одному приложения. Установить следующие свойства формы с использованием окна Инспектора объектов:

1. Выравнивание формы различными вариантами (свойство Align);

2. Оформить рамку формы различными вариантами (свойство borderstyte);

3. Задать заголовок окна формы (свойству Caption присвоит значение «Моя первая форма»);

4. Подобрать цвет формы (свойство Color);

5. Выбрать вид указателя формы из списка стандартных указателей (свойство Cursor);

6. Поэкспериментировать со свойством прозрачности формы (свойства alphablend и

alphablendvalue);

7. Изображение всплывающей подсказки формы (в свойство Hint записать текст подсказки – «Это форма», в свойство showhint записать значение true);

8. Указать состояние формы (свойство windowstaty);

9. Создать иконку (значок) для формы.

Лабораторное занятие № 26

Разработка оконного приложения с несколькими формами

Разработать приложение «Тренажер памяти» с использованием компонента счетчика cspinedit и индикатора cgauge. На форму выводится число, состоящее из заданного количества цифр, и задерживается на заданное количество секунд. Пользователь должен запомнить это число и правильно его ввести. Проводится ряд таких опросов. В конце теста делается заключение о качестве Вашей памяти.

На форме расположить:

- три компонента cspinedit для задания количества опросов, количества цифр в числе и временного интервала в секундах для запоминания числа;

- метку Label для вывода на ней сформированного датчиком случайных чисел числа с указанным количеством цифр;

- компонент Edit для ввода пользователем отгадываемого значения;

- невизуальный компонент Timer для времени отображения числа;

- снабдить форму необходимым количеством кнопок.

Описать глобальные переменные:

- целые: для количества опросов, цифр в числе, временной интервал, количество выполненных опросов, количество верных и неверных ответов.

- строковые: отгадываемое число и число, вводимое пользователем.

Создать программный код событий:

- oncreate для формы, в котором задать значения по умолчанию для тренажера; запустить датчик; очистить компоненты Label и Edit;

- ontimer для таймера, в котором скрывать случайное число в Label;

- onkeypress для компонента Edit, в котором при нажатии на клавишу [Enter] проводить сверку введенного пользователем числа в Edit и числа, скрытом в Label.

Использовать два компонента индикатора процесса cgauge: один для отображения количества совершенных попыток и другой – для отображения количества правильно выполненных заданий на текущий момент.

1. При запуске приложения выводится окна с приглашением ввода имени тестируемого.

2. После прохождения сеанса тестирования в тестовом файле сохраняется информация в виде: имя тестируемого и процент качества его памяти.

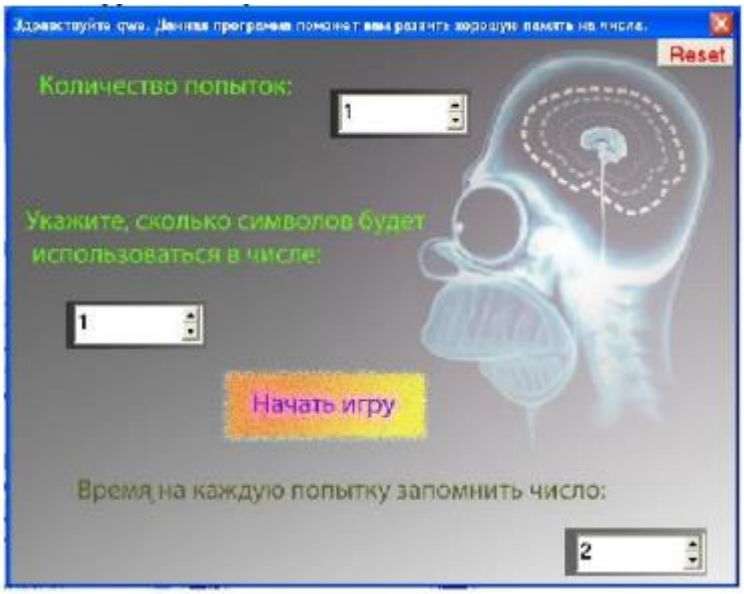


Рисунок 1 – Приложение «Тренажер памяти»

Лабораторное занятие № 27

Разработка игрового приложения

Создать игру "Тараканьи бега".

Суть игры состоит в том, чей жучёк прибежит к финишу первым. Для игры нужны две картинки жучков, один будет красный, другой синий. Картинки можно нарисовать в редакторе Paint, или загрузить к себе в папку с проектом представленные здесь.

Картинки:

Рисунок 1 – Изображение объектов «Тараканы»

Создав новый проект, следует расположить на форме следующие объекты:

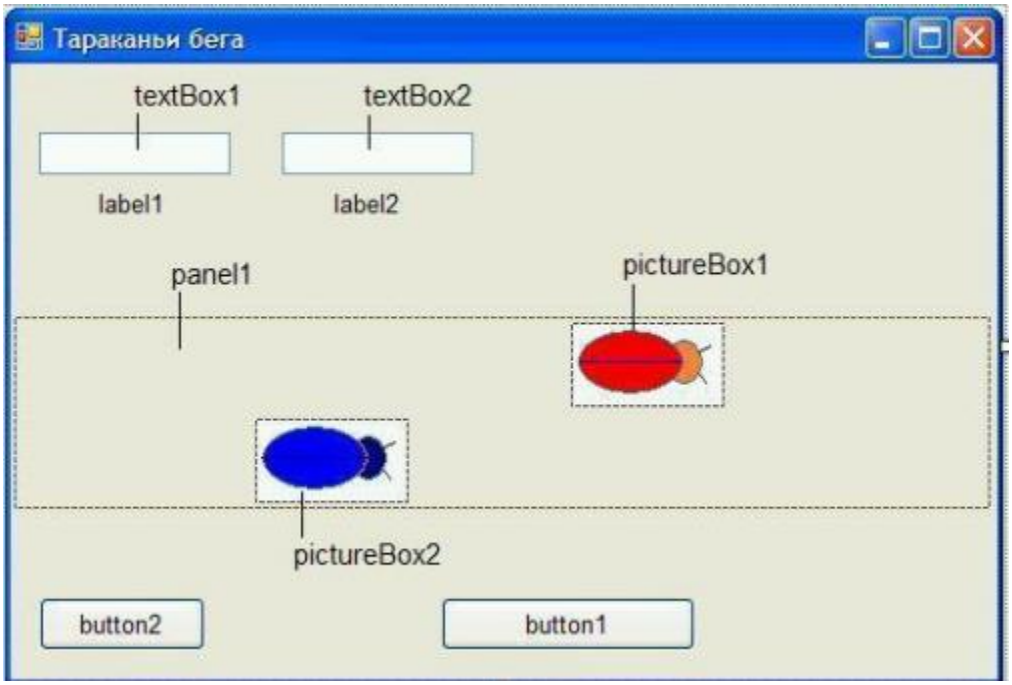


Рисунок 2 – Форма приложения

В Объекты picturebox1 и picturebox2 следует загрузить картинки. Для этого в

свойстве Image нажать маленькую кнопочку и, далее, нажать кнопку Import на панели ресурсов.

Когда картинки будут загружены, следует панели panel1 присвоить в свойстве backcolor такой же цвет, как и фон картинок жучков.

Двойным щелчком на выделенной форме следует создать событие и прописать в нём начальные установки программы:

```
Private void Form1_Load(object sender, eventargs e)
{
Textbox1.Text = "Вася";
Textbox2.Text = "Петя";
Label1.Text = "0";
Label2.Text = "0";
Picturebox1.Left = 1;
Picturebox2.Left = 1;
}
```

Кнопке button1 в свойстве Text следует присвоить значение "СТАРТ", а button2 значение "Сброс".

Запустить приложение на исполнение; форма должна принять вид, показанный на рисунке:

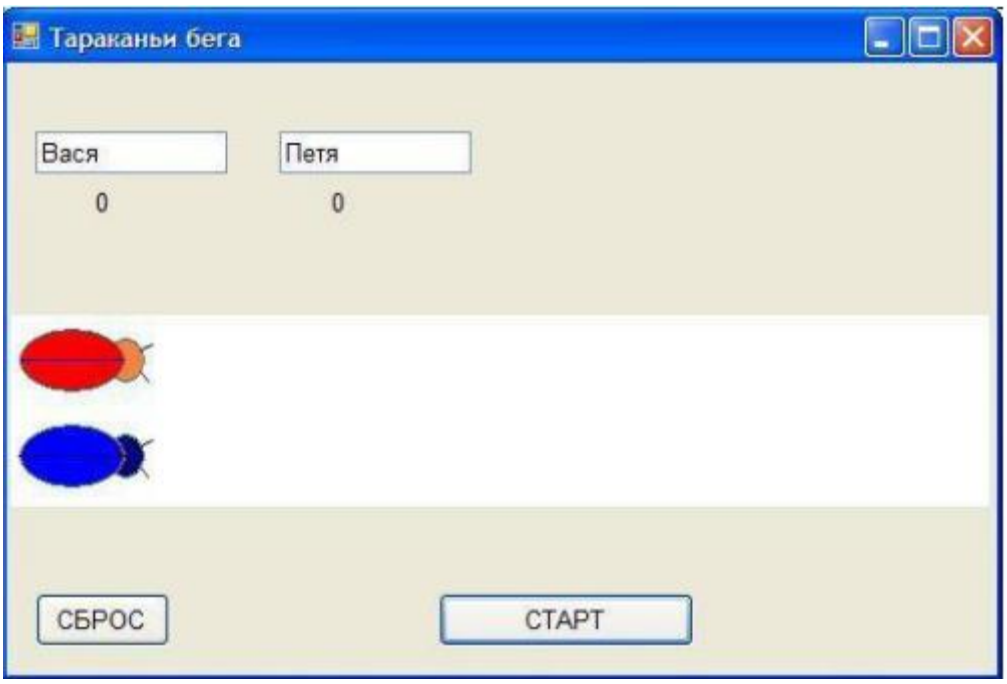


Рисунок 3 – Игровая консоль приложения

Создать в коде программы несколько глобальных переменных:

```
Public partial class Form1 : Form
{
Int flag1;
Int x1, x2;
Public Form1()
{
...
```

Переменная flag1 - предназначена для контроля игровой ситуации, если flag1 равна 0, то жучки стоят на месте, а если flag1 не равна 0, то жучки начинают двигаться. Переменные x1 и x2 предназначены для хранения горизонтальных координат положения жучков на беговой дорожке.

Создать событие для кнопки "Сброс", это событие очищает все переменные и устанавливает жучков в стартовую позицию.

```
Private void button2_Click(object sender, eventargs e)
{
Label1.Text = "0";
Label2.Text = "0";
X1 = 1; x2 = 1;
Picturebox1.Left = x1;
Picturebox2.Left = x2;
Flag1 = 0;
}
```

Для кнопки "СТАРТ" тоже создадим событие, оно будет очень простое, нужно присвоить переменной flag1 любое значение не равное 0.

```
Private void button1_Click(object sender, eventargs e)
{
Flag1 = 1;
}
```

Теперь осталось создать событие, которое будет перемещать наших жучков. Для этого понадобится компонент таймер timer1. Установить компонент на форме и произвести некоторые настройки.

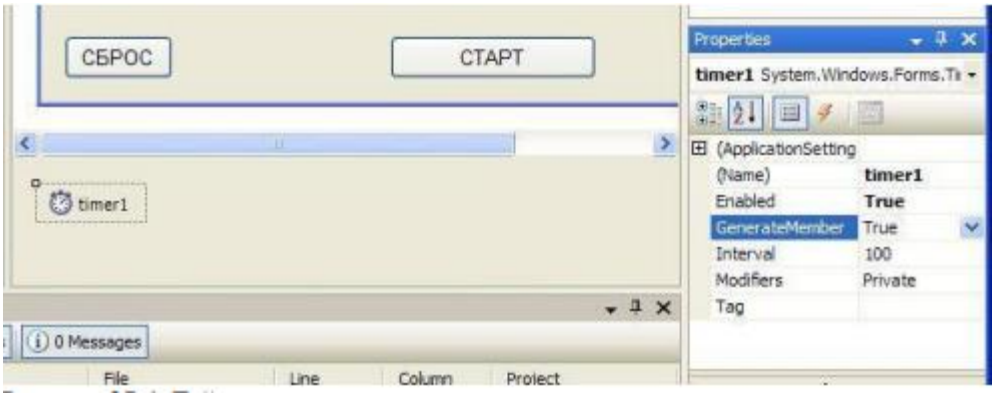


Рисунок 4 – Таймер

Свойство generatemember и Enabled должны быть True. Свойство Interval равно 100 - это один тик таймера в секунду. Если жучки будут двигаться медленно, то это значение можно уменьшить, например до 50.

Для определения координаты финиша следует одного из жучков сдвинуть до конца вправо.

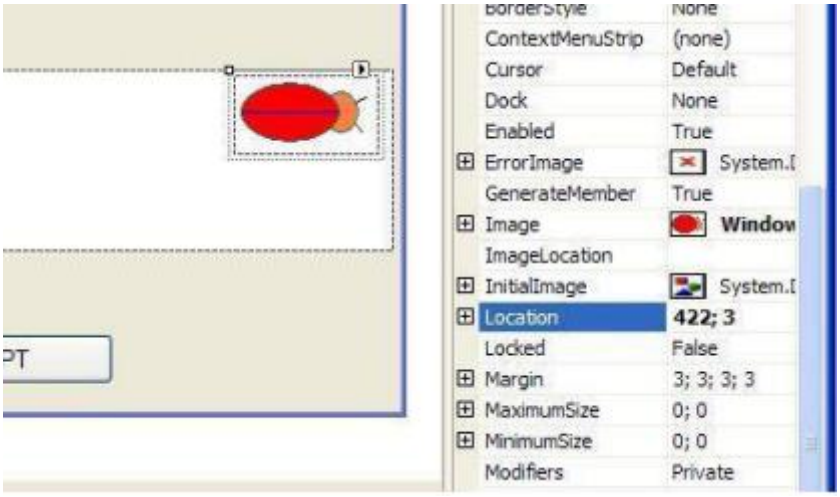


Рисунок 5 – Настройна координат

Первая цифра в свойстве Location означает финишную координату (это значение может быть другим, его нужно запомнить для вставки в код программы).

Двойным щелчком по иконке timer1 создать событие движения жучков:

```
Private void timer1_Tick(object sender, eventargs e)
{
If (flag1 != 0) // Если дан старт :
{
Random a = new Random(); // Включаем генератор случайных чисел
//    В переменную count записываем случайное число в диапазоне от 0 до 8
Int count = a.Next(8);
//    Наращиваем значение координаты x1 на случайное число count
X1 += count;
//    Выводим значение пройденного пути для первого игрока label1.Text =
t.tostring(x1);
//    Смещаем первого жучка на случайную величину picturebox1.Left = x1;

//    Создаём случайное число для второго игрока и повторяем всё то же самое
Count = a.Next(8); x2 += count;
Label2.Text = Convert.tostring(x2); picturebox2.Left = x2;
}
//    Проверяем какой из игроков дошёл до финиша и останавливаем процесс
If ((x1 >= 422) || (x2 >= 422))

{
}
```

Запустить программу на исполнения и, нажав кнопку "СТАРТ", сыграть несколько партий, чтобы убедиться, что игроки приходят к финишу с переменным успехом.

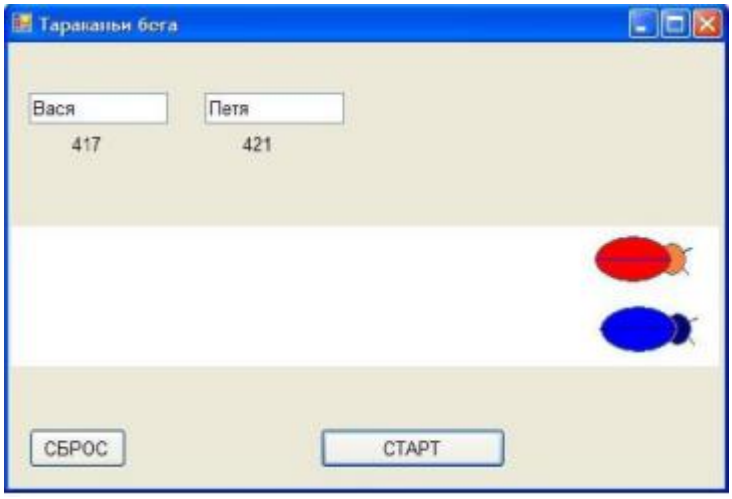


Рисунок 6 – Готовое приложение

Лабораторное занятие № 28

Создание процедур обработки событий. Компиляция и запуск приложения

Разработать приложение с использованием компонентов: компонента Shape и других необходимых элементов интерфейса для создания приложения, выполняющего операции с графической фигурой. Приложение должно настраивать отдельные графической фигуры, задавать свойства кисти и карандаша. Осуществить перетаскивание фигуры и изменение ее размера с помощью событий от мыши.

На форме расположить компоненты: Shap – основной элемент интерфейса, необходимое количество дополнительных элементов интерфейса, а так же восемь

маленьких фигур (Shap) для задания маски плоскости фигуры.

1. Тип фигуры (свойство Shape) выбирать из выпадающего списка.
2. Левой кнопкой мыши выполнять перемещение фигуры.
3. Правой кнопкой мыши выполнять изменение размера фигуры.

Настроить свойства карандаша для контура фигуры (цвет, толщину, стиль), используя для этого необходимые элементы интерфейса.

Настроить свойства кисти (цвет, маску, цвет маски). Маску выбирать из восьми маленьких фигур, заполненных стандартными масками. Для этого создать событие onmousedown, в котором заполняется плоскость основной фигуры выбранной маской.

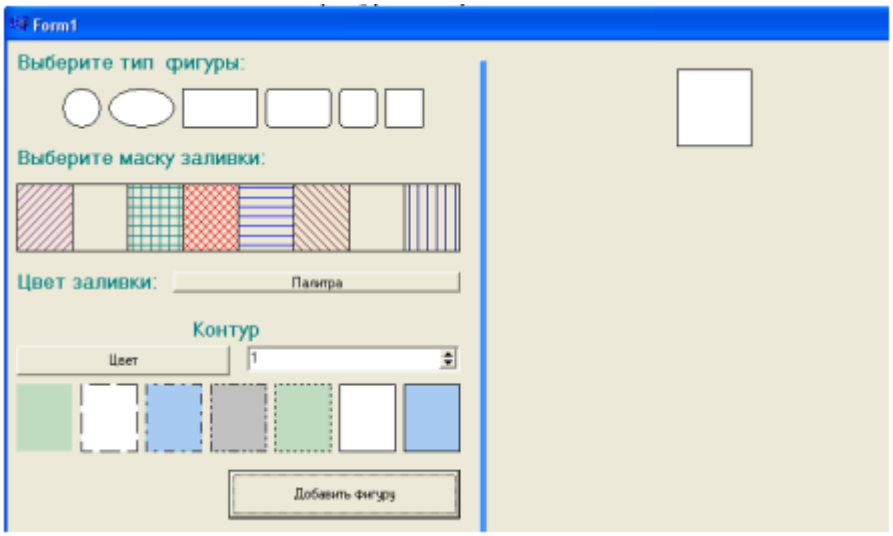


Рисунок 1 – Фрагмент работы приложения «Редактор фигур»

Лабораторное занятие № 29

Разработка интерфейса приложения

Разработать приложение с использованием компонентов: paintbox, Timer и других необходимых элементов интерфейса для создания приложения, выполняющего построение графических примитивов с заданными условиями отображения. Для построения графических примитивов использовать спецификацию класса tcanvas компонента paintbox.

1. На форме разместить компонент paintbox, который должен занимать большую часть формы и являться областью отображения графических примитивов.
2. В выпадающем списке выбрать тип примитива: точки, линии, прямоугольники, круговые фигуры, дуги, хорды, сектора, шрифты. Выбрать нужный примитив из выпадающего списка.
3. Атрибуты для кажого примитива задаются случайными значениями с учетом возможного диапазона для данного свойства.
4. В контейнере radiogroup выбрать режим отображения примитивов: по одному, количество (указать нужное количество), бесконечно.
5. Обеспечить скорость смены рисования примитива, выбрав необходимое значение для скорости таймера.
6. Обеспечить работу кнопок: «Пуск», «Пауза», «Очистить».
7. Использовать четыре компонента paintbox одного размера.
8. Для каждого компонента выбрать изображаемые в нем графические примитивы с

помощью компонента checklistbox.

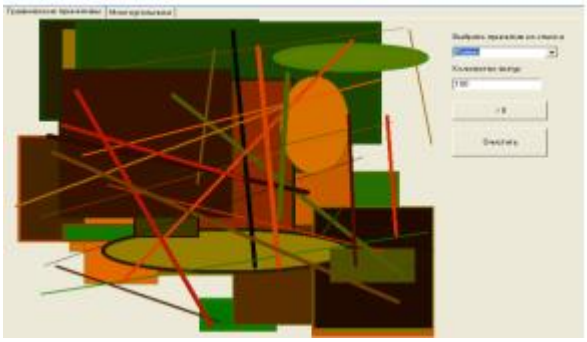


Рисунок 1 – Фрагмент работы приложения «Графические примитивы»

Лабораторное занятие № 30

Тестирование, отладка приложения

Разработать приложение с использованием компонентов: paintbox и других необходимых элементов интерфейса для создания приложения, выполняющего построение графических фигур правильной формы с заданными условиями отображения и расчетными координатами вершин.

Для построения графических примитивов использовать спецификацию класса tcanvas для компонента paintbox.

1. На форме разместить компонент paintbox, который должен занимать большую часть формы и являться областью отображения графических примитивов.

2. Задать количество вершин многоугольника.

3. Задать величину радиуса многоугольника.

4. Расчитать координаты вершин многоугольника по формулам вращения.

5. Построить многоугольник.

6. Установить флажок, с помощью которого задается режим соединения вершин правильного многоугольника.

7. Вводя коэффициент смещения, выполнить сдвиг вложенного многоугольника на заданное значение коэффициента.

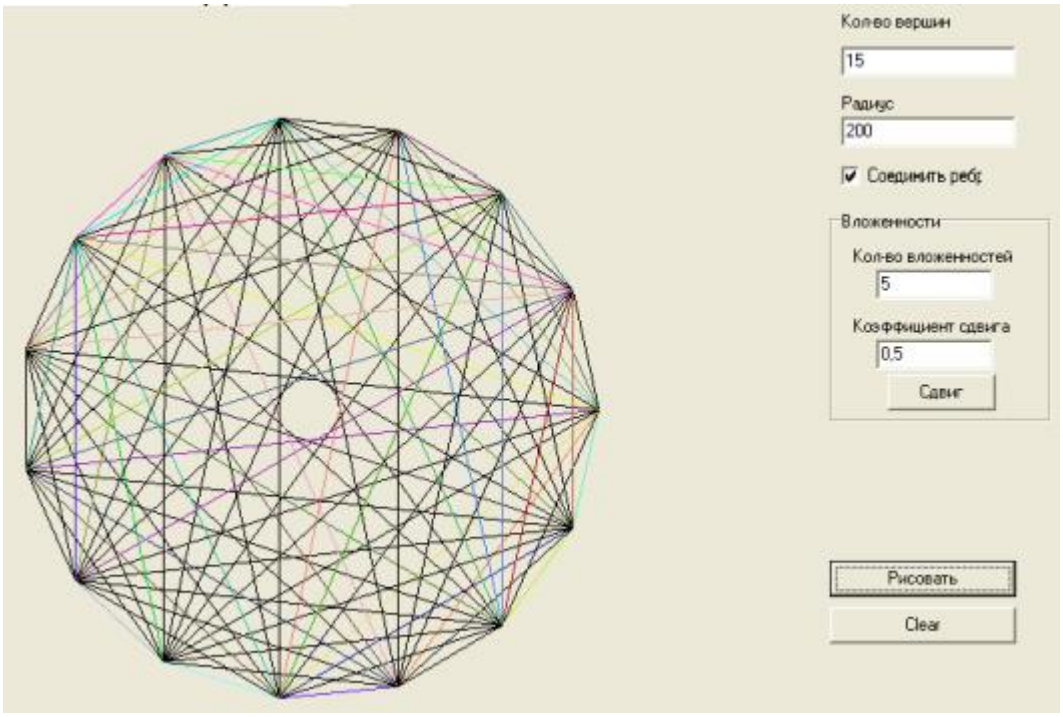


Рисунок 1 – Фрагмент работы приложения «Построение геометрических фигур»

Лабораторные занятия № 31-32

Классы ООП: виды, назначение, свойства, методы, события.

Написать программу с использование объектов, выполнив следующую последовательность действий:

1. Создать новый тип - класс, включив в его состав необходимые свойства и методы. Описать класс – овал со свойствами:

a. Радиусы по горизонтали и вертикали в пикселях;

b. Методы:

c. Инициализация свойств объекта;

d. Изменить значения радиусов овала;

e. Расчитать плщадь овала

f. Расчмтать длину дуги овала.

g. Реализовать методы созданного класса.

h. Описать объекты созданного класса.

2. Создать один экземпляр класса – одиночный объект, и к нему применить запрограммированные методы.

3. Создать несколько объектов созданного класса, и к каждому объекту применить методы.

4. Сформировать массив из 5 объектов. Применить к каждому элементу такого массива методы.

Лабораторное занятие № 33

Создание наследованного класса

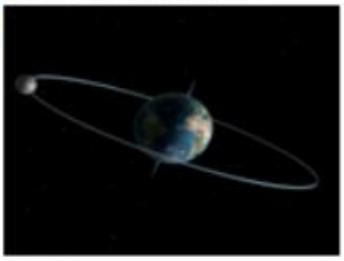
1. На основе базового класса «Автомобиль», создать производный класс «Общественный транспорт».

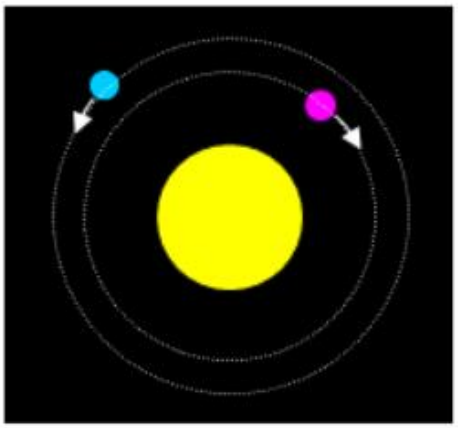
2. На основе базового класса «Общественный транспорт», создать производный класс «Автобус».

3. На основе базового класса «Общественный транспорт», создать производный класс «Трамвай».

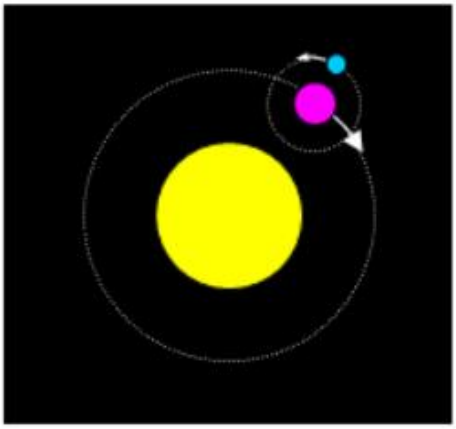
Лабораторное занятие № 34
Программирование приложений

«3»: Изобразить модель Солнца с одной планетой





«4»: Изобразить модель Солнца с двумя планетами, которые вращаются в противоположные стороны:



«5»: Изобразить модель системы Солнце-Земля-Луна:

Лабораторное занятие № 35
Перегрузка методов

1. Разработайте программу, содержащую класс с двумя открытыми и двумя закрытыми целочисленными переменными. Создайте два объекта этого класса. Используйте для инициализации закрытых переменных класса параметризованные конструкторы. Дополните программу функциями расчета суммы всех переменных объекта.

2. Создайте функции, определяющие максимальное и минимальное значение переменных каждого объекта. Используйте коструктор.

3. Определите, в каком объекте сумма переменных максимальна (минимальна). Используйте перегрузку конструкторов.

Критерии оценивания практических работ

Если практическая работа выполнена в полном объеме и правильно оформлена, то ставится оценка «5».

Если практическая работа выполнена более чем на 75%, ставится оценка «4».

Если практическая работа выполнена более чем на 60%, ставится оценка «3».

В противном случае работа не засчитывается.

3.2.2 Самостоятельная работа

№	Название разделов и тем	Виды заданий
1	**Тема 1.2** Типы данных	Анализ понятий по теме
2	**Тема 2.1** Операторы языка программирования	Анализ понятий по теме
3	**Тема 3.3** Модульное программирование	Анализ понятий по теме
4	**Тема 4.1** Указатели	Анализ понятий по теме
5	**Тема 5.6** Иерархия классов	Анализ понятий по теме

Критерии оценивания анализа понятий

Параметр	Оценка
– представлено не менее 5 определений; – определения взяты из различных источников; – источники оформлены в соответствии с правила библиографического описания; – содержание словарных статей представлено развернуто; – перечень определений оформлен в соответствии с требованиями; – задание выполнено в срок.	5
– представлено менее 5 определений; – определения взяты из различных источников (менее 5); – источники оформлены с отступлениями от правил библиографического описания; – содержание словарных статей представлено развернуто; – перечень определений оформлен в соответствии с требованиями; – задание выполнено в срок.	4
– представлено менее 3 определений; – определения взяты из различных источников (менее 3); – источники оформлены с отступлениями от правил библиографического описания; – содержание словарных статей представлено не развернуто; – перечень определений оформлен не в соответствии с требованиями; – задание выполнено не в срок.	3
– задание обучающимся не выполнено.	2

4 КОНТРОЛЬНО-ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ИТОГОВОЙ АТТЕСТАЦИИ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Предметом оценки являются умения и знания. Контроль и оценка осуществляются с использованием следующих форм и методов: практическая работа, самостоятельная работа, устный опрос.

Оценка освоения дисциплины предусматривает использование экзамена.

4.1 Паспорт контрольно-оценочных материалов

Назначение:

КОМ предназначен для контроля и оценки результатов освоения учебной дисциплины ОП.04 Основы алгоритмизации и программирования по специальности СПО 09.02.07 Информационные системы и программирование.

Умения

У1 – разрабатывать алгоритмы для конкретных задач;

У2 – использовать программы для графического отображения алгоритмов;

У3 – определять сложность работы алгоритмов;

У4 – работать в среде программирования;

У5 – реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;

У6 – оформлять код программы в соответствии со стандартом кодирования;

У7 – выполнять проверку, отладку кода программы.

Знания:

З1 – понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;

З2 – эволюцию языков программирования, их классификацию, понятие системы программирования;

З3 – основные элементы языка, структуру про граммы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;

З4 – подпрограммы, составление библиотек подпрограмм;

З5 – объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на при мере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Общие компетенции:

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

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

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде

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

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках

Профессиональные компетенции:

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

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

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

4.2 Задания экзаменующегося

Теоретические вопросы для подготовки к экзамену

1. Области применения языков программирования. Стандарты языков программирования.
2. Среда проектирования.
3. Компиляторы и интерпретаторы.
4. Жизненный цикл программы.
5. Программа. Программный продукт и его характеристики.
6. Основные этапы решения задач на компьютере.
7. Простые типы данных.
8. Производные типы данных.
9. Структурированные типы данных.
10. Операции и выражения.
11. Правила формирования и вычисления выражений.
12. Структура программы.
13. Ввод и вывод данных.
14. Оператор присваивания.
15. Составной оператор.
16. Условный оператор.
17. Оператор выбора.
18. Цикл с постусловием.
19. Цикл с предусловием.
20. Цикл с параметром.
21. Вложенные циклы.
22. Массивы. Двумерные массивы.

23. Строки. Стандартные процедуры и функции для работы со строками.
24. Структурированный тип данных – множество. Операции над множествами.
25. Комбинированный тип данных – запись.
26. Файлы последовательного доступа. Файлы прямого доступа.
27. Общие сведения о подпрограммах. Определение и вызов подпрограмм.
28. Область видимости и время жизни переменной.
29. Механизм передачи параметров.
30. Организация функций.
31. Рекурсия. Программирование рекурсивных алгоритмов.
32. Основы структурного программирования. Методы структурного программирования.
33. Модульное программирование. Понятие модуля. Структура модуля.
34. Компиляция и компоновка программы.
35. Стандартные модули.
36. Указатели. Описание указателей.
37. Основные понятия и применение динамически распределяемой памяти.
38. Создание и удаление динамических переменных.
39. Структуры данных на основе указателей. Задача о стеке.
40. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.
41. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.
42. Классы объектов.
43. Компоненты и их свойства.
44. Событийно-управляемая модель программирования.
45. Компонентно-ориентированный подход.
46. Требования к аппаратным и программным средствам интегрированной среды разработчика.
47. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов.
48. Панель компонентов и их свойства.
49. Окно кода проекта.
50. Состав и характеристика проекта. Выполнение проекта.
51. Настройка среды и параметров проекта.
52. Панель компонентов и их свойства. Окно кода проекта.
53. Состав и характеристика проекта. Выполнение проекта.
54. Настройка среды и параметров проекта.
55. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.
56. Дополнительные элементы управления.
57. Свойства компонентов. Виды свойств.
58. Синтаксис определения свойств. Назначения свойств и их влияние на результат.
59. Управление объектом через свойства.
60. События компонентов (элементов управления), их сущность и назначение.
61. Создание процедур на основе событий.
62. Разработка функционального интерфейса приложения. Создание интерфейса приложения.

63. Разработка функциональной схемы работы приложения.
64. Разработка игрового приложения. Разработка приложения.
65. Проектирование объектно-ориентированного приложения.
66. Создание интерфейса пользователя.
67. Тестирование, отладка приложения.
68. Классы ООП: виды, назначение, свойства, методы, события.
69. Перегрузка методов.

ПРИМЕР ЗАДАНИЯ
ВАРИАНТ № 1

Инструкция для обучающихся

1. В задании – 2 теоретических вопроса и 1 задача.
2. Время выполнения задания – 40 мин.

Задания.

1. Настройка среды и параметров проекта.
2. Компиляция и компоновка программы.
3. Задача. Составить программу вычисления для заданных значений x, y, z арифметического выражения $u = \mathrm{tg}^2(x+y) - \mathrm{e}^{y-z}\sqrt{\cos x^2 + \sin z^2}$

ПАКЕТ ЭКЗАМЕНАТОРА

УСЛОВИЯ.

Для проведения экзамена в кабинет приглашаются по 5 студентов. Каждый студент выбирает вариант, и в течение 40 минут выполняет задания.

Количество вариантов задания для экзаменующегося – 20 вариантов.

Время выполнения задания – 60 минут.

Оборудование: ручка, компьютер.

Экзаменационная ведомость – оценочный лист и экзаменационная ведомость

КРИТЕРИИ ОЦЕНКИ.

Шкала оценивания теоретических вопросов (задания № 1, 2)[1]:

Баллы	Критерии оценивания	Показатели оценивания
3	Степень усвоения учебного материала	– свободное владение учебным материалом
	Полнота освещения каждого из предложенных вопросов	– глубоко и полно раскрыты теоретические и практические аспекты вопроса
	Логика изложения	– изложение общих и частных аспектов вопроса логично;

[1] Каждое теоретическое задание оценивается отдельно

Баллы	Критерии оценивания	Показатели оценивания
		– ответы четкие
	Самостоятельность изложения	– творческий подход к изложению материала; – продемонстрирована дискуссионность проблематики; – использован дополнительный материал
	Полнота освещения дополнительных вопросов	– дополнительные вопросы глубоко и полно раскрыты; – продемонстрировано знание пограничных вопросов
	Характеристика устной речи	– речь ясная, четкая, рассудительная
2	Степень усвоения учебного материала	– владение учебным материалом в рамках лекционного курса и обсуждения на семинарских занятиях
	Полнота освещения каждого из предложенных вопросов	– раскрыты основные аспекты вопроса
	Логика изложения	– имеются логические неточности; – ответы по существу вопроса
	Самостоятельность изложения	– чувствуется потенциал, который может быть реализован при дополнительной проработке отдельных тем дисциплины
	Полнота освещения дополнительных вопросов	– дополнительные вопросы раскрыты в общих моментах
	Характеристика устной речи	– имеются «мелкие» оговорки
1	Степень усвоения учебного материала	– учебный материал усвоен слабо
	Полнота освещения каждого из предложенных вопросов	– недостаточно полное освещение узловых моментов вопроса; – раскрыты отдельные аспекты вопроса
	Логика изложения	– логика нарушена; – ответы не всегда конкретны
	Самостоятельность изложения	– сделана попытка самостоятельного осмысления и изложения изучаемого материала
	Полнота освещения дополнительных вопросов	– раскрыты отдельные дополнительные вопросы
	Характеристика устной речи	– имеются затруднения при объяснении основных категорий

Баллы	Критерии оценивания	Показатели оценивания
		дисциплины
0	Степень усвоения учебного материала	– учебный материал не усвоен
	Полнота освещения каждого из предложенных вопросов	– в ответах показано слабое понимание теоретических аспектов вопроса, – не раскрыт ни один из аспектов вопроса, – ответы поверхностны
	Логика изложения	– логика в изложении не наблюдается; – ответы не по существу вопроса
	Самостоятельность изложения	– материал излагается только на основе учебника без ссылок на дополнительную литературу
	Полнота освещения дополнительных вопросов	– знание пограничных вопросов практически отсутствует; – студент затрудняется более глубоко обосновать те или иные положения
	Характеристика устной речи	– речь путанная, сбивчивая, страдает оговорками

Шкала оценивания практических заданий (задание № 3):

Критерии	Баллы
Правильный ход решения, все его шаги выполнены верно	3
Правильный ход решения, не все его шаги выполнены верно, допущено 1-2 логических ошибки	2
Решение начато логически верно, но допущены ошибки, либо решение не доведено до конца, при этом результат неверный или отсутствует.	1
Неверное решение, неверный результат или отсутствие решения.	0

Шкала соответствия набранных баллов оценкам:

Общее количество баллов	Соответствие пятибалльной системе
0 – 3	Неудовлетворительно
4 – 5	Удовлетворительно
6 – 7	Хорошо
8– 9	Отлично