

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФИЛИАЛ ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО  
ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«БАЙКАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ЭКОНОМИКИ И ПРАВА»  
(Филиал ФГБОУ ВПО «БГУЭП» в г. Усть-Илимске)



ПРОГРАММА ДИСЦИПЛИНЫ

ВЫСОКОУРОВНЕВЫЕ МЕТОДЫ ИНФОРМАТИКИ  
И ПРОГРАММИРОВАНИЯ БЗ.В.3

Направление подготовки 230700 Прикладная информатика

Профиль подготовки:

Информационные системы и технологии в управлении

Квалификация (степень) выпускника Бакалавр

Форма обучения Очная

	Очное обучение
Курс	2
Семестр	4
Лекции	36
Практические (семинарские, лабораторные) занятия	18
Самостоятельная работа	90
Всего часов	144
Курсовая работа	4
Зачет	-
Экзамен	4

Усть-Илимск 2012

## СОДЕРЖАНИЕ

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	3
2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП БАКАЛАВРИАТА .....	4
3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ .....	5
4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ.....	7
4.1. Содержание разделов дисциплины .....	7
4.2. Лекционные занятия, их содержание.....	8
4.3. Семинарские, практические, лабораторные занятия, их содержание.....	11
4.4. Вид и форма промежуточной аттестации.....	12
5. ИСПОЛЬЗУЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ.....	13
6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ.....	14
6.1. Текущий контроль .....	14
6.2. Образцы тестовых и контрольных заданий текущего контроля.....	14
6.3. Тематика рефератов, эссе, докладов .....	19
6.4. Темы курсовых работ, критерии оценивания.....	19
6.5. Методические указания по организации самостоятельной работы.....	20
6.6. Промежуточный контроль .....	24
7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	28
8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	30

## 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Жизненный цикл программы практически в каждый момент времени имеет отношение к некоторому технологическому процессу, некоторому языку и некоторому инструменту системы программирования. Поэтому необходимо не только знание каждого из этих направлений, но и понимание теснейших связей и зависимостей между ними. Для предметной области программирования группы знаний выглядят так:

- фундаментальные знания, математические основы программирования;
- знание методологий программирования, группы языков, классов систем и технологий;
- знание особенностей конкретных программных средств.

Цель изучения дисциплины «Высокоуровневые методы информатики и программирования» – развить способность рассмотреть основные методологии, технологии программирования, заложить фундамент понимания основ программирования на примере языке высокого уровня Object Pascal и современной интегрированной среды визуальной разработки программ Delphi.

Задачи изучения дисциплины: дать студентам теоретические знания о программных конструкциях, применяемых во всех языках высокого уровня, научить студентов разрабатывать программы для решения широкого круга задач с использованием интегрированных сред быстрой разработки, научить создавать эргономичный пользовательский интерфейс с использованием стандартных компонент среды.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП БАКАЛАВРИАТА

Дисциплина «Высокоуровневые методы информатики и программирования» относится к вариативной части профессионального цикла БЗ.

Для изучения данной дисциплины обучающийся должен обладать знаниями, умениями и готовностями, приобретенными в результате освоения следующих дисциплин: Информатика и программирование, Информационные системы и технологии.

Общая трудоемкость дисциплины составляет 4 зачетные единицы.

### 3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс изучения дисциплины направлен на формирование следующих компетенций.

Код компетенции	Компетенция
ПК-4	Способен ставить и решать прикладные задачи с использованием современных информационно-коммуникационных технологий
ПК-5	Способен осуществлять и обосновывать выбор проектных решений по видам обеспечения информационных систем
ПК-7	Способен использовать технологические и функциональные стандарты, современные модели и методы оценки качества и надежности при проектировании, конструировании и отладке программных средств

Ключевыми компетенциями, формируемыми в процессе изучения дисциплины являются ПК-5.

Уровневое описание признаков компетенции ПК-5:

Способен осуществлять и обосновывать выбор проектных решений по видам обеспечения информационных систем

Уровень освоения	Признаки проявления
Продвинутый (91 – 100 баллов)	Умеет использовать инструментальные средства, поддерживающие разработку программного обеспечения профессионально-ориентированных информационных систем; иметь опыт современного программирования и использования возможностей вычислительной техники и программного обеспечения; классифицировать языки программирования, проводить декомпозицию системы на объекты и процессы, выделять этапы объектно-ориентированного проектирования
Базовый (71– 90 баллов)	Иметь представление: о особенностях программирования в средах современных информационных систем, о создании модульных программ, об элементах теории модульного программирования, об основах объектно-ориентированного проектирования и программирования, визуального программирования, о способах размещения нового компонента, реакции на события, о назначении компонентов, способах использования компонентов, обработки и накопления информации
Минимальный (41 – 70 баллов)	Иметь представление: о тенденциях развития компьютерной техники и программных средств, технических средств информатизации современные достижения вычислительной техники (вычислительные машины, системы и сети телекоммуникаций); об общей характеристике процессов сбора, передачи, обработки и накопления информации

В результате освоения дисциплины обучающийся должен:

Знать: объектно-ориентированный подход к проектированию и разработке программ; особенности программирования в оконных операционных средах; компоненты; использование компонентов.

Уметь: применять объектно-ориентированный подход к проектированию и разработке программ; создавать приложения на Delphi и др.; выполнять тестирование и отладку программ; создавать справочную систему; оформлять программную документацию.

Владеть: работы в интегрированной среде разработки Delphi и др.; работы с научно-технической литературой и технической документацией по программному обеспечению ПЭВМ.

Уровень «знать» достигается проведением лекционных и практических занятий и самостоятельной работой. Уровень «уметь» достигается проведением практических и лабораторных занятий.

#### 4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 4 зачетные единицы (144 часа).

##### 4.1. Содержание разделов дисциплины

№ п/п	Раздел и тема дисциплины	Семестр	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)			Формы текущего контроля успеваемости
			Лекции	Семинар Лаборат. Практич.	Самост. раб.	
1	Происхождение, история и эволюция методологий программирования	4	8	4	27	Контрольные работы, тестирование
1.1	Тема 1. Происхождение, история и эволюция методологий программирования		2	1	9	
1.2	Тема 2. Основные понятия из области программирования		2	1	9	
1.3	Тема 3. Языки и системы программирования		4	2	9	
2	Программирование в средах современных информационных систем		28	14	63	Контрольные работы, лабораторные работы, тестирование
2.1	Тема 4. Методы проектирования алгоритмов и программ		4	2	9	
2.2	Тема 5. Элементы теории модульного программирования. Создание модульных программ		4	2	9	
2.3	Тема 6. Объектно-ориентированное проектирование и программирование		4	2	9	
2.4	Тема 7. Объектно-ориентированный подход к проектированию и разработке программ. Конструкторы и деструкторы	4	2	9		
2.5	Тема 8. Особенности программирования в оконных операционных средах. Среда разработки. Система окон	4	2	9		

	разработки. Система меню					
2.6	Тема 9. Основы визуального программирования. Размещение нового компонента. Реакция на события. Компоненты. Использование компонентов.		4	2	9	
2.7	Тема 10. Отладка и тестирование программ		4	2	9	
	Итого		36	18	90	

#### 4.2. Лекционные занятия, их содержание

№ п/п	Наименование разделов и тем	Содержание
1	Происхождение, история и эволюция методологий программирования	
1.1	Тема 1. Происхождение, история и эволюция методологий программирования	<p>Законы эволюции программного обеспечения. Основные методологии программирования. Атрибуты методологий. Краткая характеристика методологий: структурное программирование, объектно-ориентированное программирование, функциональное программирование, логическое программирование.</p> <p>Введение в технологии программирования. Основные понятия и определения. Технология создания ПО. Процесс. Стадия. Технологический подход.</p> <p>Новейшие направления в области создания технологий программирования. Case-системы. Индустрия искусственного интеллекта. Экспертные системы.</p>
1.2	Тема 2. Основные понятия из области программирования	<p>Программирование как вид деятельности. Взаимосвязь программирования и других областей знания.</p> <p>Понятия алгоритм, исполнитель алгоритма. Пять важных свойств алгоритмов. Величины в алгоритмах. Базовые структуры алгоритмов: линейный алгоритм, разветвленный алгоритм, циклический алгоритм. Формы представления алгоритма: словесная, блок-схема, псевдокод, программная. ГОСТ 19.701-90. Схемы алгоритмов, программ, данных и систем. Термин задача. Решение задачи. Постановка задачи. Классификация задач: технологические, функциональные. Понятие вычислителя.</p> <p>Программа как формализованное описание процесса обработки данных. Программное средство. Жизненный цикл программного средства. Основная задача профессионального программирования. Понятие качества программного средства. Характеристики качества программного</p>



		обеспечения.
1.3	Тема 3. Языки и системы программирования	<p>Элементы языка: алфавит, лексемы, синтаксис и семантика языка. Особенности языков программирования. Характеристики языков программирования: уровень, мощность, концептуальная целостность, надежность, удобочитаемость, полнота, гибкость, мобильность, эффективность. Уровни языков программирования: машинный язык, язык ассемблера, язык высокого уровня. Взаимосвязь уровня и мощности языка.</p> <p>Основные понятия и определение системы программирования. Структура современной системы программирования. Классификация систем программирования: по ориентации на поддержку технологических процессов, по категориям, по предоставляемому интерфейсу, по функциональной ориентации в технологическом процессе.</p> <p>Инструментальные системы программирования (инструментальные среды программирования, средства автоматизации разработки программ, интегрированные среды).</p> <p>Трансляция программ. Трансляторы: интерпретаторы, компиляторы. Компилируемые и интерпретируемые языки высокого уровня. Выбор языка программирования. Критерии и факторы выбора.</p>
2	Программирование в средах современных информационных систем	
2.1	Тема 4. Методы проектирования алгоритмов и программ	<p>Классификация методов проектирования алгоритмов и программ. По степени автоматизации проектных работ: неавтоматизированное (традиционное) проектирование и автоматизированное проектирование (Case-технология и ее элементы). По принятой методологии процесса разработки различают подходы к построению программ: метод функциональной декомпозиции (структурное проектирование программных средств), информационное моделирование предметной области и связанных с ней приложений, объектно-ориентированное проектирование программных средств. Методы, ориентированные на обработку: модульное программирование, функциональная декомпозиция, проектирование с использованием потока данных, технология структурного анализа проекта.</p>
2.2	Тема 5. Элементы теории модульного программирования. Создание модульных программ	<p>Метод структурного проектирования и программирования: нисходящее проектирование, модульное программирование, структурное программирование.</p> <p>Понятие модуля. Характеристика модуля. Принцип модульного программирования.</p> <p>Общая структура программы на языке Object Pascal. Структура модуля. Заголовок модуля и связь моду-</p>

		<p>лей друг с другом. Интерфейсная часть. Исполняемая часть. Инициализирующая часть. Компиляция модулей. Доступ к объявленным в модуле объектам. Стандартные модули. Использование стандартных модулей.</p> <p>Подпрограммы. Процедуры и функции. Формальный параметр. Фактический параметр. Передача параметров. Организация и использование подпрограмм. События. Обработчики событий.</p>
2.3	Тема 6. Объектно-ориентированное проектирование и программирование	<p>Основные теоретические положения ООП. Три принципа ООП: инкапсуляция, наследование, полиморфизм. Понятия класса, объекта. Свойство. Метод. Событие. Соотношение основных понятий ООП.</p>
2.4	Тема 7. Объектно-ориентированный подход к проектированию и разработке программ. Конструкторы и деструкторы	<p>Объектно-ориентированное проектирование программ. Объектная декомпозиция. Работа с объектами. Создание объектов. Использование объектов. Конструкторы и деструкторы. Методы построения классов. Объектный тип данных. Переменные объектного типа.</p>
2.5	Тема 8. Особенности программирования в оконных операционных средах. Среда разработки. Система окон разработки. Система меню	<p>Понятие IDE. Структура интегрированной среды разработки Delphi. Среда разработки и ее составляющие. Система окон разработки. Окно конструктора форм, окно кода программы, окно инспектора объектов. Проводник по коду.</p> <p>Система меню. Панели инструментов. Палитра компонентов. Справочник.</p> <p>Настройка интегрированной среды разработки Delphi. Проект. Характеристика проекта. Файл проекта. Файлы формы. Файлы модулей. Файл ресурсов. Параметры проекта. Создание и сохранение нового проекта. Особенности сохранения проектов. Компиляция и выполнение программы. Ошибки компиляции.</p>
2.6	Тема 9. Основы визуального программирования. Размещение нового компонента. Реакция на события. Компоненты. Использование компонентов.	<p>Основы визуального программирования. Определяющие элементы процесса визуализации: визуализируемая модель (окно, форма, диалог), окно инструментов, окно свойств. Принципы визуального программирования. Элементы управления. Проектирование формы. Визуальные и объектные аспекты форм. Архитектура приложения. Разработка приложений. Компонентный подход системы Delphi. Палитра компонентов. Общая характеристика компонентов. Визуальные и не визуальные компоненты. Использование компонентов. Размещение компонентов. Элементы управления. Определение функциональности приложения. Событие и процедура обработки события.</p> <p>Язык программирования Object Pascal. Элементы языка: алфавит, лексемы, лексические единицы, синтаксис, оформление. Константы и переменные,</p>

		<p>именование, объявление и использование. Данные и типы данных языка. Стандартные и определяемые пользователем. Простые и сложные типы данных. Организация действий над данными. Ввод/вывод данных. Обработка данных: операции, приоритеты операций, выражения. Операторы языка: классификация и форматы операторов.</p> <p>Формы в Delphi. Характеристики формы. Организация взаимодействия окон. Шаблоны форм. Классы для системы меню. Создание интерфейса приложения. Меню, подсказки. Характеристика главного меню. Контекстное меню. Конструктор меню. Компоненты для отображения текста. Ввод и редактирование информации. Общие элементы компонентов редактирования. Однострочные редакторы. Многострочный редактор. Работа со списками. Общая характеристика списков. Простой список. Комбинированный список. Особенности комбинированного списка. Работа с кнопками. Стандартная кнопка. Кнопка с рисунком. Кнопка быстрого доступа. Использование переключателей. Переключатель с независимой фиксацией. Переключатель с зависимой фиксацией. Объединение элементов управления. Графические компоненты. Геометрическая фигура. Графический образ. Окно рисования. Рисование при выполнении программы. Поверхность рисования. Анимация. Выполнение практических заданий лабораторных практикумов.</p>
2.7	Тема 10. Отладка и тестирование программ	<p>Введение в тестирование. Типичные ошибки (синтаксические, логические, времени исполнения). Защита от ошибок. Тест и тестирование. Этапы процесса тестирования. Стратегии тестирования. Подходы к тестированию: сверху вниз, снизу вверх. Отладка. Стандартная техника отладки</p>

#### 4.3. Семинарские, практические, лабораторные занятия, их содержание

Лабораторная работа №1. Знакомство с элементами интегрированной среды разработки Delphi.

Лабораторная работа №2. Знакомство с компонентами среды Delphi (вкладка Standard палитры компонентов). Использование компонентов.

Лабораторная работа №3, №4. Знакомство со структурой программ среды разработки Delphi. Знакомство с элементами языка Object Pascal.

Лабораторная работа №5, №6. Знакомство с типами данных.

Лабораторная работа №7. Создание меню (главного, контекстного).

Лабораторная работа №8. Знакомство с дополнительными компонентами Delphi страницы Standard. Элементы управления. Интерактивный ввод и вывод.

Лабораторная работа №9. Создание пользовательских функций и процедур.

Лабораторная работа №10. Работа с графикой. Графические компоненты. Окно рисования. Геометрическая фигура.

Лабораторная работа №11. Создание движущих изображений (анимация).

Лабораторная работа №12. Стандартная техника отладки. Обработчики исключений.

#### 4.4. Вид и форма промежуточной аттестации

Промежуточный контроль проводится в виде экзамена (по всему курсу, включая темы, изученные самостоятельно) в 4 семестре.

## 5. ИСПОЛЬЗУЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Изучение данного курса предполагает проведение лекций, практических занятий, индивидуальную работу преподавателя со студентами и самостоятельную работу студентов.

Лекции с проблемным изложением проводятся с применением мультимедийного оборудования в виде презентаций. Данные лекции становятся доступными для обучающихся при подготовке к разного вида контролю и СРС. Лекции-дискуссии, деловые игры (выбор конкретного оборудования и его обоснование), написание рефератов.

Лабораторные работы предусматриваются для более глубокого закрепления теоретических знаний, углубления навыков студентов.

При проведении лабораторных работ учебные группы разбиваются на подгруппы. Количество подгрупп определяется наличием и возможностями демонстрационных систем, лабораторных установок и автоматизированных рабочих мест в учебном классе.

Любой практическое занятие включает самостоятельную проработку теоретического материала и изучение методики решения типичных задач. Некоторые задачи содержат элементы научных исследований, которые могут потребовать углубленной самостоятельной проработки теоретического материала.

Для оказания помощи студентам в освоении учебного материала в часы самостоятельной работы регулярно проводятся групповые и индивидуальные консультации.

Доля занятий с использованием активных и интерактивных методов составляет 50%.

## 6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

### 6.1. Текущий контроль

Текущий контроль рекомендуется осуществлять в соответствии с разработанной рейтинговой системой по дисциплине:

Раздел 1 30 баллов	1. Тестирование по теме «Основные понятия из области программирования» (10 баллов) 2. Контрольная работа № 1 по теме «Базовые алгоритмические структуры» (10 баллов) 3. Контрольная работа № 2 по теме «Способы записи алгоритмов» (10 баллов)
Раздел 2 70 баллов	1. Тестирование (10 баллов) 2. Контрольная работа № 3 по теме «Пользовательские процедуры и функции» (10 баллов) 3. Защита 10 лабораторных практикумов, за каждый 5 баллов (50 баллов)

### 6.2. Образцы тестовых и контрольных заданий текущего контроля

#### Пример тестового задания

##### Вариант 1. Выберите правильный ответ из списка

1. Независимая программная единица, служащая для выполнения некоторой определенной функции программы и для связи с остальной частью программы

- модуль
- оператор
- обработчик
- алгоритм

2. Тип String предназначен для хранения:

- строк
- целых чисел
- вещественных чисел
- массивов

3. Тип Char предназначен для хранения:

- строк
- целых чисел
- вещественных чисел
- отдельных символов

4. С помощью оператора WHILE .. DO:

- три пути работы программы
- циклическую обработку по заданному условию
- не циклическую обработку
- условие для выборки значений

5. Последовательность действий или метод преобразования входных данных в правильный результат – это...

- решение задачи

- алгоритм
6. Программная конструкция, используемая для многократного выполнения группы операторов – это ...
- условный оператор
  - оператор цикла
  - оператор присваивания
7. Верное описание глобальной переменной Kolvo для хранения значений о количестве и весе продаваемых товаров:
- Kolvo: Byte
  - Kolvo: Real
  - Kolvo: Integer
  - Kolvo: Double
8. С помощью одного оператора IF...THEN можно создать:
- три пути работы программы
  - два пути работы программы
  - четыре пути работы программы
  - неограниченное количество путей работы программы
9. Оператор FOR..DO используется для:
- выбора значений из списка значений
  - создания счетного цикла
  - создания цикла с условием
  - обработки исключительных ситуаций
10. Узнать длину строки sSource (тип String) можно:
- Length(sSource)
  - sSource.Length
  - Max(sSource)
11. Инstrukция WriteLn(F, S) выполняет:
- запись строки S в файл
  - чтение строки S из файла
  - удаление всех символов S из файла
12. Инstrukция ReadLn(F, S) выполняет:
- запись строки S в файл
  - чтение строки S из файла
  - удаление всех символов S из файла
13. Функция EOF(F) возвращает:
- номер текущего символа в файле
  - истину, если достигнут конец файла
  - ложь, если достигнут конец файла
  - истину, если достигнут конец строки
14. Базовые канонические структуры алгоритмов – это ...
- выбор
  - повторение
  - переход
  - развилка
  - следование
  - ввод
15. При выполнении группы операторов K:=0; I:=1; while I<=10 do K:=K+1; I:= I+1; в Pascal зацикливание возникает из-за:
- начального значения I
  - начального значения K
  - условия в заголовке цикла while
  - того, что величина I в теле цикла не меняется

16. В системе программирования после обнаружения факта ошибки начинается процесс ее поиска и исправления, который называют ...

- отладкой
- компиляцией
- тестированием
- трассировка

17. Этапы разработки программной системы с использованием объектно-ориентированного подхода:

- постановка задачи
- анализ
- катализ
- синтез
- проектирование
- реализация
- модификация

18. Этапы разработки программной системы с использованием структурного подхода:

- постановка задачи
- анализ
- катализ
- синтез
- проектирование
- реализация
- модификация

19. Если тело цикла содержит другой оператор цикла, то такие циклы называются

...

- встроенными
- вложенными
- внутренними
- внешними

20. Объекты программы (константы, типы, переменные, процедуры, функции), описанные во внешних блоках, но не описанные во внутренних блоках, по отношению к внутренним блокам являются ...

- пользовательскими
- личными
- глобальными
- локальными

## Образцы контрольных работ

### Контрольная работа № 1.

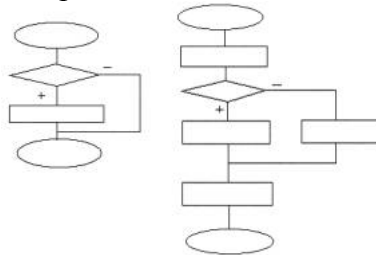
#### Вариант 1

Ответить на вопросы в письменной форме:

1. Как записывается и выполняется команда выбора?
2. Какая разница между условиями, записанными после слова while и repeat для одной и той же задачи?
3. Как организовать цикл с уменьшением счетчика?
4. Какого типа выражения допустимы в качестве начального и конечного значений?
5. Как необходимо записать несколько операторов, чтобы они выполнялись в цикле for?



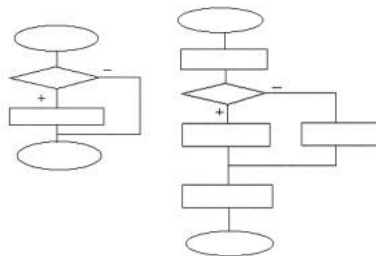
6. Как описывается одномерный массив?
7. Как описывается двумерный массив?
8. Что можем использовать в качестве индекса?
9. Что фактически указывает тип индекса?
10. Какой тип может быть типом индекса?
11. Придумать задания под приведенные блок-схемы.



### Вариант 2

Ответить на вопросы в письменной форме:

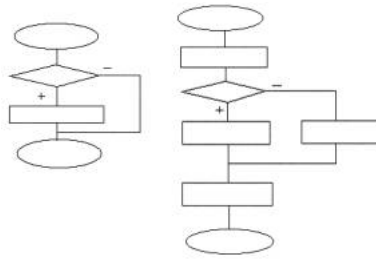
1. Как выполняется команда ветвления в неполной форме?
2. Какие логические операции используются в команде ветвления?
3. Как оформляется цикл while?
4. Когда завершается выполнение цикла while?
5. Как организовать выполнение нескольких операторов в цикле while?
6. Управляющие переменные какого типа допустимы в операторе for?
7. Как обращаться к элементам массива?
8. Чем однозначно определяется значение элемента массива?
9. Назовите простые типы используемые в массиве?
10. Как называем элемент массива?
11. Придумать задания под приведенные блок-схемы.



### Вариант 3

Ответить на вопросы в письменной форме:

1. Как выполняется команда ветвления в полной форме?
2. Какие правила оформления команды ветвления существуют?
3. Какие алгоритмы называются циклическими?
4. Какой из операторов цикла является самым универсальным?
5. Для управления циклом Repeat какие функции удобно использовать?
6. Когда используется оператор цикла FOR?
7. Какого типа выражения допустимы в качестве начального и конечного значений?
8. Что такое массив?
9. Где должен быть описан, используемый в программе массив?
10. Какого типа должен быть индекс?
11. Придумать задания под приведенные блок-схемы.



## Контрольная работа № 2.

### Вариант 1

1. Чему будет равно значение переменной  $x$  после выполнения фрагмента программы?

```
x:=0;
for i:=1 to 5 do
x:=x+1;
```

2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы на Object Pascal для вычисления функции при заданном значении  $X$

$$Z = \begin{cases} x - 0.5 & \text{при } x < 0 \\ x & \text{при } 0 \leq x < 4 \\ \sin x + \cos x & \text{при } x \geq 4 \end{cases}$$

### Вариант 2

1. Чему будет равно значение переменной  $f$  после выполнения фрагмента программы?

```
f:=2; x:=0;
for i:=1 to 5 do
begin
x:=x+1;
f:= f * x;
end;
```

2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы на Object Pascal для вычисления размера подоходного налога ( $N$ ). Подоходный налог определяется: 1) зарплата ( $Z$ ) меньше минимальной заработной платы ( $M$ ) – не облагается; 2) если выше, то берется 13% от суммы, превышающей минимальную.

### Вариант 3

1. Чему будет равно значение переменной  $S$  после выполнения фрагмента программы?

```
x:=0; S:=0;
while x<= 2 do
begin
S:= S + x;
x:=x+2;
end;
```

2. Составить алгоритм решения задачи в форме блок-схемы, псевдокода, текста программы на Object Pascal для вывода на экран агрегатного состояния воды (лед, жидкость, пар) в зависимости от введенной температуры.

### 6.3. Тематика рефератов, эссе, докладов

Не предусмотрено

### 6.4. Темы курсовых работ, критерии оценивания

Целью выполнения курсовой работы является практическое использование полученных базовых знаний о современных методах проектирования и разработки прикладных программ и информационных систем на основе концепций объектно-ориентированного подхода к проектированию и разработке программ, а также практическое освоение современных технологий программирования и концепций создания программных приложений, развитие навыков самостоятельной работы, связанных с анализом, детализацией, выбором методов решения поставленных задач, планированием использования возможностей сред программирования, а также различных источников информации для реализации задач курсовой работы.

В ходе выполнения курсовой работы по данной дисциплине у студента должно расширяться и углубляться представление о принципах и возможностях использования инструментальных систем программирования.

В ходе достижения цели решаются следующие задачи:

- дальнейшего развития логического и алгоритмического мышления;
- углубленного изучения принципов работы прикладного программного обеспечения; выработки умения использовать современные инструментальные средства для разработки, отладки и тестирования создаваемых прикладных программ;
- выработки умения самостоятельного анализа, детализации, выбора методов решения задач, планированием использования возможностей сред программирования, а также различных источников информации для реализации этих задач.

При выполнении курсовой работы студент должен ознакомиться с предметной областью по теме выполняемой работы и в необходимой мере исследовать эту область. В соответствии с темой курсовой работы студент должен отобрать и проанализировать информацию, необходимую для реализации задач курсовой работы, и выбрать способы решения этих задач с использованием возможностей соответствующих сред программирования.

Выполнение курсовых работ по дисциплине "Высокоуровневые методы информатики и программирования" может быть связано: а) с реализацией возможностей, предоставляемых средами программирования, в виде программных приложений, ориентированных на конечного пользователя (см. ниже темы 1-16 примерного перечня тем курсовых работ); б) с изучением возможностей использования компонентов сред программирования и реализацией этих возможностей в виде программных приложений демонстрационного характера (см. там же темы 17-25); в) с изучением информационных и образовательных ресурсов, имеющих отношение к использованию сред программирования (см. там же темы 26-28).

Примерный перечень тем курсовых работ:

1. Программа: Решение алгебраических уравнений 1-й, 2-й, 3-й и 4-й степени.
2. Программа: Решение системы  $N$  линейных алгебраических уравнений с  $N$  неизвестными.
3. Программа: Операции с алгебраическими матрицами.
4. Программа: Двоичный калькулятор.
5. Программа: Восьмеричный калькулятор.
6. Программа: Шестнадцатеричный калькулятор.

7. Программа: Электронный кассовый аппарат.
8. Программа: Переводы единиц измерения.
9. Программа: Оболочка для тестирования
10. Программа: Оболочка для анкетирования.
11. Программа: Календарь с часами.
12. Программа: Многофункциональные электронные часы.
13. Программа: Многофункциональный секундомер.
14. Программа: Протокол соревнований.
15. Программа: Таблица чемпионата.
16. Программа: Поиск сбойных участков на диске.
17. Форма, ее свойства и примеры использования.
18. Возможности и примеры использования окон редактирования и меток.
19. Возможности и примеры использования кнопок, флажков, переключателей.
20. Возможности и примеры использования списков.
21. Возможности и примеры использования панелей.
22. Изобразительные возможности среды программирования.
23. Возможности среды программирования для работы с файлами.
24. Сравнение возможностей сред программирования на примере создания и использования простого приложения.
25. Сравнение возможностей сред программирования на примерах использования компонентов.
26. Обзор литературы и CD по среде программирования.
27. Internet-поддержка среды программирования.
28. Образовательные Internet-ресурсы дисциплины "Высокоуровневые методы информатики и программирования".

#### 6.5. Методические указания по организации самостоятельной работы

Самостоятельная работа имеет своей целью углубление знаний студентов по изучаемой дисциплине.

Текущая самостоятельная работа предусматривает следующие виды:

- работа с лекционным материалом;
- подготовка к лабораторным работам, оформление отчетов по выполненным лабораторным работам, подготовка к защите;
- подготовка к различным формам промежуточной аттестации (к тестированию, контрольным работам);
- изучение рекомендованной литературы (основной и дополнительной), работа с библиотечным каталогом, самостоятельный подбор необходимой литературы;
- поиск необходимой информации через Интернет;
- изучение тем, вынесенных на самостоятельную проработку;
- работа со встроенными справочными системами программных продуктов;
- работа с техническими справочниками (англо-русский);
- выполнение тестовых заданий, выполнение контрольных работ;
- подготовка к экзамену.

Творческая проблемно-ориентированная самостоятельная работа включает следующие виды:

- поиск, анализ, структурирование информации по темам, выносимым на самостоятельное изучение;
- составление и разработка словаря (гlossария);
- выполнение творческих заданий.

## Содержание самостоятельной работы по темам курса

Тема 1. Законы эволюции программного обеспечения. Основные методологии и технологии программирования. Новейшие направления в области создания технологий программирования.

1. Работа с лекционным материалом, изучение рекомендованной литературы, самостоятельный подбор необходимой литературы, поиск необходимой информации через Интернет. Оформить конспект в электронном виде.

2. Составление словаря (глоссария) по списку ключевых слов, понятий, терминов для тестирования.

3. Подготовка к тестированию и выполнение тестовых заданий.

Классификация видов программирования. Происхождение, история и эволюция методологий программирования. История и эволюция технологий программирования. Классификация технологических подходов. Краткая характеристика каждого. Проблемы и перспективы развития технологических подходов. Стандартный набор технологических процессов, основанный на стандарте ISO 12207:1995. Основные процессы (приобретение, поставка, разработка, эксплуатация, сопровождение). Вспомогательные процессы (документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, совместная оценка, аудит, разрешение проблем). Организационные процессы (управление, создание инфраструктуры, усовершенствование, обучение). Взаимосвязь между процессами. Инструментарий технологии программирования. Состав и назначение.

Тема 2. Основные понятия из области программирования

1. Работа с лекционным материалом, изучение рекомендованной литературы, самостоятельный подбор необходимой литературы, поиск необходимой информации через Интернет. Оформить конспект в электронном виде.

2. Составление словаря (глоссария) по списку ключевых слов, понятий, терминов для тестирования.

3. Подготовка к тестированию и выполнение тестовых заданий.

4. Подготовка и выполнение контрольной работы № 2.

Этапы процесса разработки программ для структурного подхода и для объектно-ориентированного подхода. Краткая характеристика этапов создания программного обеспечения: постановка задачи, анализ и исследование задачи, проектирование программы (разработка алгоритма, структуры программы, разработка пользовательского интерфейса), реализация (написание кода программы на выбранном языке программирования, тестирование и отладка кода), составление документации, сопровождение программы, модификация программы. Критерии качества программ. Рекомендации по созданию программ.

Тема 3. Языки и системы программирования

1. Работа с лекционным материалом, изучение рекомендованной литературы, самостоятельный подбор необходимой литературы, поиск необходимой информации через Интернет. Оформить конспект в электронном виде.

2. Составление словаря (глоссария) по списку ключевых слов, понятий, терминов для тестирования.

3. Подготовка к тестированию и выполнение тестовых заданий.

4. Подготовка и выполнение контрольной работы №2.

История развития языков программирования. Поколения языков программирования. Обзор языков программирования высокого уровня. Классификация языков программирования: по поддерживаемым методологиям, по принадлежности к семействам, по ориентации на предметные области. Проблемы и перспективы развития языков. История возникновения систем программирования. Принципы функционирования систем программирования: функции текстовых редакторов в системах программирования, компилятор как составная часть системы программирования, назначение и функции компоновщика, загрузчики и отладчики, библиотеки подпрограмм как составная часть систем программирования. Примеры современных систем программирования. Системы программирования компании Borland/Inprise (Turbo Pascal, Borland Delphi, Borland C++ Builder). Системы программирования фирмы Microsoft (Microsoft Visual Basic, Microsoft Visual C++). Концепция .NET. Разработка программного обеспечения для сети Интернет (Язык HTML, языки программирования Java и Java Script). Эволюция Delphi. Разработка приложений. Компонентный подход системы Delphi. Палитра компонентов. Общая характеристика компонентов. Визуальные и не визуальные компоненты. Использование компонентов. Размещение компонентов. Элементы управления. Определение функциональности приложения. Стиль программирования. Правила хорошего стиля программирования в Object Pascal. Комментарии. Выполнение заданий самостоятельной работы лабораторных практикумов.

#### Тема 4. Методы проектирования алгоритмов и программ

1. Работа с лекционным материалом, изучение рекомендованной литературы, самостоятельный подбор необходимой литературы, поиск необходимой информации через Интернет. Оформить конспект в электронном виде.

2. Составление словаря (глоссария) по списку ключевых слов, понятий, терминов для тестирования.

3. Подготовка к тестированию и выполнение тестовых заданий.

Этапы создания программных продуктов при неавтоматизированной разработке программ. Структура программных продуктов. Проектирование интерфейса пользователя: диалоговый режим, графический интерфейс пользователя. Метод структурного проектирования и программирования: нисходящее проектирование, модульное программирование, структурное программирование. Объектно-ориентированное проектирование программ. Основные понятия и методика ООП. Объект. Свойство. Метод. Событие. Класс. Соотношение основных понятий ООП.

Тема 5. Элементы теории модульного программирования. Создание модульных программ.

1. Работа с лекционным материалом, изучение рекомендованной литературы, самостоятельный подбор необходимой литературы, поиск необходимой информации через Интернет. Оформить конспект в электронном виде.

2. Составление словаря (глоссария) по списку ключевых слов, понятий, терминов для тестирования.

3. Подготовка к тестированию и выполнение тестовых заданий.

4. Доработка лабораторных работ, оформление отчетов по выполненным лабораторным работам, подготовка к защите.

Набор стандартных подпрограмм: математические и статистические процедуры и функции, процедуры и функции для работы с датой/временем, процедуры и функции преобразования типов, процедуры и функции для работы со строками и символами.

## Тема 6. Объектно-ориентированное проектирование и программирование

1. Работа с лекционным материалом, изучение рекомендованной литературы, самостоятельный подбор необходимой литературы, поиск необходимой информации через Интернет. Оформить конспект в электронном виде.

2. Составление словаря (глоссария) по списку ключевых слов, понятий, терминов для тестирования.

3. Подготовка к тестированию и выполнение тестовых заданий.

Реализация принципов ООП. Этапы реализации объектно-ориентированного подхода. Создание пользовательских интерфейсов. Реализация интерфейсных элементов.

Тема 7. Объектно-ориентированный подход к проектированию и разработке программ. Конструкторы и деструкторы.

1. Работа с лекционным материалом, изучение рекомендованной литературы, самостоятельный подбор необходимой литературы, поиск необходимой информации через Интернет.

2. Составление словаря (глоссария) по списку ключевых слов, понятий, терминов для тестирования.

3. Подготовка к тестированию и выполнение тестовых заданий.

Тема 8. Особенности программирования в оконных операционных средах. Среда разработки. Система окон разработки. Система меню.

Работа с лекционным материалом, изучение рекомендованной литературы, самостоятельный подбор необходимой литературы, поиск необходимой информации через Интернет.

1. Составление словаря (глоссария) по списку ключевых слов, понятий, терминов для тестирования.

2. Подготовка к тестированию и выполнение тестовых заданий.

3. Выполнение творческих заданий.

4. Работа со встроенными справочными системами программных продуктов.

5. Доработка лабораторных работ, оформление отчетов по выполненным лабораторным работам, подготовка к защите.

Тема 9. Основы визуального программирования. размещение нового компонента. Реакция на события. Компоненты. Использование компонентов.

Работа с лекционным материалом, изучение рекомендованной литературы, самостоятельный подбор необходимой литературы, поиск необходимой информации через Интернет. Оформить конспект в электронном виде.

Составление словаря (глоссария) по списку ключевых слов, понятий, терминов для тестирования.

Подготовка к тестированию и выполнение тестовых заданий.

Подготовка и выполнение контрольной работы № 3.

Выполнение творческих заданий.

Работа со встроенными справочными системами программных продуктов.

Доработка лабораторных работ, оформление отчетов по выполненным лабораторным работам, подготовка к защите.

Обзор палитры компонентов. Стил ь программирования. Правила хорошего ст иля программирования в Object Pascal. Комментарии. Особенности модальных форм. Особенности немодальных форм. Диалоговые окна. Стандартные диалоги. Процедуры и функции, реализующие диалоги. Общая характеристика визуальных компонентов. Свойства, события, методы. Класс TString. Группа. Панель. Область прокрутки. Фреймы.

## Тема 10. Отладка и тестирование программ

Работа с лекционным материалом, изучение рекомендованной литературы, самостоятельный подбор необходимой литературы, поиск необходимой информации через Интернет.

Составление словаря (глоссария) по списку ключевых слов, понятий, терминов для тестирования.

Подготовка к тестированию и выполнение тестовых заданий.

Работа со встроенными справочными системами программных продуктов.

Доработка лабораторных работ, оформление отчетов по выполненным лабораторным работам, подготовка к защите.

Подготовка к экзамену.

### 6.6. Промежуточный контроль

Промежуточный контроль проводится в экзамена (по всему курсу, включая темы, изученные самостоятельно) в 4 семестре.

Условием допуска студента к экзамену является выполнение всех лабораторных работ и их своевременная защита, а также своевременная сдача промежуточного контроля в виде тестирования и контрольных работ по всем дидактическим единицам.

Экзамен проводится в устной форме. Экзаменационные билеты содержат теоретическую и практическую части.

Критерии оценки знаний студентов в целом по дисциплине:

«отлично» - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений; ответ на экзамене характеризуется научной терминологией, четкостью, логичностью, умением самостоятельно мыслить и делать выводы.

«хорошо» - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

«удовлетворительно» - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

«неудовлетворительно» - выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

### Вопросы к экзамену

1. Основные понятия программирования. Программа и программное обеспечение. Программный продукт. Его характеристика.



2. Жизненный цикл программного обеспечения. Содержание этапов классического ЖЦ. Качество ПО. Критерии качества.
3. Основные области в программировании: методологии и технологии программирования; языки и системы программирования; операционные и архитектурные платформы. Краткая характеристика каждой группы.
4. Методология программирования. Атрибуты методологии. Ядра методологий. Основные методологии: императивного программирования, ООП, функционального программирования, логического программирования, активного программирования (экстремального). Краткая характеристика.
5. Технологии программирования. Основные понятия и определения.
6. Технологический подход. Классификация подходов.
7. Три стратегии конструирования программного обеспечения.
8. Водопадная модель разработки приложений.
9. Инкрементная модель разработки приложений.
10. Спиральная модель разработки приложений.
11. Тяжеловесные и облегченные процессы разработки приложений.
12. Характеристики XP-процесса.
13. Технологический процесс. Классификация технологических процессов.
14. История и эволюция технологий программирования. Проблемы и перспективы развития.
15. Алгоритмы. Понятие исполнителя алгоритма. Важные свойства алгоритмов. Формы представления алгоритма: словесная, блок-схема, псевдокод, программная. Виды структур алгоритма.
16. Языки программирования. Уровни языков программирования. Машинный язык. Язык ассемблера. Язык высокого уровня.
17. Эволюция языков программирования.
18. Классификация языков программирования.
19. Обзор языков программирования высокого уровня.
20. Компилируемые и интерпретируемые языки высокого уровня. Компилятор. Интерпретатор.
21. Определение системы программирования. Традиционный состав системы программирования.
22. Характеристика структуры современной системы программирования. Функции текстовых редакторов в системах программирования.
23. Характеристика структуры современной системы программирования. Компилятор как составная часть системы программирования.
24. Характеристика структуры современной системы программирования. Компоновщик. Назначение и функции.
25. Характеристика структуры современной системы программирования. Загрузчики и отладчики. Функции.
26. Характеристика структуры современной системы программирования. Библиотеки подпрограмм.
27. Примеры современных систем программирования.
28. Методы проектирования программ.
29. Модульное программирование. Преимущества модульного программирования. Определение модуля. Свойства модуля. Модульная структура приложения.
30. Два типа подпрограмм. Синтаксис определения процедуры и функции. Синтаксис вызовов процедуры и функции. Отличия процедур и функций.
31. Использование функций и процедур. Определение фактических и формальных параметров. Передача параметров по ссылке и по значению.
32. Объектно-ориентированное программирование. Основное понятие ООП. Объекты и классы. Определения, назначение.

33. Принцип инкапсуляции.
34. Принцип наследования.
35. Принцип полиморфизма.
36. Методы, события и свойства. Определения и назначение.
37. Конструкторы и деструкторы.
38. Стил программирования. Правила хорошего стили в Object Pascal.
39. Object Pascal. Типы данных. Структура типов данных.
40. Object Pascal. Типы данных. Простые типы.
41. Object Pascal. Типы данных. Структурированные типы.
42. Object Pascal. Типы данных. Массивы. Записи.
43. Object Pascal. Массивы. Статические и динамические массивы. Работа с ними.
44. Object Pascal. Типы данных. Множества и строки.
45. Object Pascal. Типы данных. Целые, логические, символьные типы.
46. Object Pascal. Типы данных. Вещественные типы.
47. Object Pascal. Типы данных. Файлы.
48. Сложные структуры данных. Тип Record и тип File (TextFile).
49. Object Pascal. Синтаксис языка. Алфавит. Словарь языка. Комментарии.
50. Object Pascal. Структура программы, раздел описаний.
51. Object Pascal. Структура программы. Модули, главный файл проекта.
52. Object Pascal. Структура программы. Переменные и константы.
53. Object Pascal. Структура программы. Раздел операторов.
54. Object Pascal. Условный оператор IF. Оператор ветвлений CASE.
55. Object Pascal. Циклы. FOR, WHILE, REPEAT.
56. Object Pascal. Функции и процедуры.
57. Object Pascal. Примеры типов данных.
58. Основы визуального программирования. Элементы процесса визуализации. Принципы визуального программирования. Элементы управления. Проектирование форм. Визуальные и объектные аспекты форм. Архитектура приложения.
59. Среда визуального программирования. Основные понятия и принципы системы Delphi.
60. Среда разработки и ее составляющие. Окно формы, кода программы, инспектора объектов.
61. Понятие проекта. Характеристика. Состав проекта. Параметры проекта.
62. Файл проекта.
63. Файлы формы
64. Файлы модулей
65. Файл ресурсов.
66. Создание и сохранение нового проекта. Особенности сохранения проектов.
67. События. Обработчик событий.
68. Компиляция и выполнение проекта.
69. Тестирование программ. Методы тестирования.
70. Отладка программ. Причины ошибок Типы ошибок в программе: времени выполнения, синтаксические, логические. Защита от ошибок.
71. Компонентный подход системы Delphi. Использование компонентов. Палитра компонентов.
72. Библиотека визуальных компонентов.
73. Общая характеристика визуальных компонентов. Свойства, события, методы.
- Класс TString.
74. Компоненты для отображения текста.
75. Ввод и редактирование информации. Общие элементы компонентов редактирования. Однострочный редактор. Многострочный редактор.

- 76. Работа со списками. Общая характеристика списков. Простой список. Комбинированный список. Особенности комбинированного списка.
  - 77. Работа с кнопками. Стандартная кнопка. Кнопка с рисунком. Кнопка быстрого доступа.
  - 78. Использование переключателей. Переключатель с независимой фиксацией. Переключатель с зависимой фиксацией.
  - 79. Объединение элементов управления. Группа. Панель. Область прокрутки.
  - 80. Графические компоненты. Геометрическая фигура. Графический образ. Окно рисования.
  - 81. Рисование при выполнении программы. Поверхность рисования. Анимация.
  - 82. Разработка приложений. Определение функциональности приложения.
  - 83. Формы в Delphi. Характеристики формы. Организация взаимодействия окон.
- Шаблоны форм.
- 84. Особенности модальных форм. Особенности немодальных форм.
  - 85. Диалоговые окна. Стандартные диалоги. Процедуры и функции, реализующие диалоги.
  - 86. Создание интерфейса приложения. Меню, подсказки. Характеристика главного меню. Контекстное меню. Конструктор меню.

## 7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

### а) основная литература:

1. Бабушкина, И.А. Практикум по объектно-ориентированному программированию / И.А. Бабушкина, С.М. Окулов. – М.: БИНОМ. Лаборатория знаний, 2009 – 366 с.
2. Бескоровайный, И. В. Азбука Delphi: программирование с нуля. – Новосибирск: Сибирское университетское издательство, 2008. – 112 с.
3. Давыдова Н.А. Программирование. Учебное пособие / Н.А.Давыдова. – М.:БИНОМ, Лаборатория знаний, 2012 – 239с.
4. Молочков В.П. От Delphi 7 к Delphi 2006. Для начинающих. – М.: Диалог-МИФИ, 2007.
5. Программирование: Учебно-методическое пособие / Авт.-сост.О.В.Рязанова. – Барнаул; Рубцовск: АлтГУ, 2007 – 255с.
6. Стивенс, Р. Delphi. Готовые алгоритмы. Визуальный язык программирования. Rad для программистов. – Издательство «ДМКПресс», 2007. – 384 с.

### б) дополнительная литература:

1. Иванова, Г.С. Технология программирования: Учебник / Г.С. Иванова. - М.: КноРус, 2011 – 336с.– 640 с.
2. Истомина, Е.П. Высокоуровневые методы информатики и программирования: учебник / Е.П. Истомина, В.В. Новиков, М.В. Новикова. - 3-е изд.- СПб.: ООО "Андреевский издательский дом", 2010 – 228с.
3. Климова, Л.М. Delphi 7: Основы программирования. Решение типовых задач: Самоучитель – М.: Кудиц-Образ, 2006 – 480 с.
4. Князева, М.Д. Алгоритмика: от алгоритма к программе: учебное пособие / М.Д. Князева. – М.: КУДИЦ-ОБРАЗ, 2006. – 192 с.
5. Котляров, В.П. Основы тестирования программного обеспечения: Учебное пособие / В.П. Котляров. – М.: Интернет-Университет Информационных Технологий, 2009 – 285с.
6. Меньшиков, Ф.В. Олимпиадные задачи по программированию / Ф.В. Меньшиков. – СПб: Питер, 2007 – 315с.
7. Мишенин, А.И. Сборник задач по программированию: Учебное пособие / А.И. Мишенин. – М.: Финансы и статистика, 2009 – 224с.
8. Непейвода, Н.Н. Стили и методы программирования: Курс лекций. Учебное пособие / Н.Н. Непейвода. – М.: Интернет-Университет Информационных Технологий, 2009 – 320с.
9. Рудаков, А.В. Технология разработки программных продуктов: Практикум: Учебное пособие. – М.: АКАДЕМА, 2010. – 192 с.
10. Рудаков, А.В. Технология разработки программных продуктов: Учебник / А.В. Рудакова. - 6-е изд., стереотип.- М.: Академия, 2011 – 208с.
11. Сеницын, С.В. Программирование на языке высокого уровня : учебник для студ. высш. учеб. заведений / С.В. Сеницын, А.С. Михайлов, О.И. Хлытчиев. – М.: Издательский центр "Академия", 2010 – 400с.
12. Свердлов, С.З. Языки программирования и методы трансляции: Учебное пособие / С.З. Свердлов. – СПб: Питер, 2007 – 638с.
13. Сухарев, М. Золотая книга Delphi: С обновлениями до версии 2010 / М. Сухарев. – СПб.: Наука и Техника, 2010 – 1040с.

### в) программное обеспечение и Интернет-ресурсы:

1. Информационная система "Единое окно доступа к образовательным ресурсам" [Электронный ресурс] – Режим доступа: [http://window.edu.ru/resource/192/77192/Афанасьева, Т. В. Алгоритмы и программы: учебное пособие / Т.В. Афанасьева, Ю.Е. Кувайскова, В.А. Фасхутдинова. - Ульяновск: УлГТУ, 2011. - 227 с.](http://window.edu.ru/resource/192/77192/Афанасьева,Т.В.Алгоритмыипрограммы:учебноепособие/Т.В.Афанасьева,Ю.Е.Кувайскова,В.А.Фасхутдинова.-Ульяновск:УлГТУ,2011.-227с)
2. Научно-технический и научно-производственный журнал «Информационные технологии», раздел Программирование и CASE-технологии // <http://novtex.ru/IT/>
3. Поисковые системы: Google, Yandex, Rambler.
4. Программирование Realcoding.Net - Программирование линейное, С++, Delphi, С#, .NET, 1с, системы, языки, обучение: Программирование для чайников и начинающих // [www.realcoding.net](http://www.realcoding.net).
5. Центр Информационных технологий / Библиотека on-line // <http://www.citforum.ru>.

## 8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Мультимедийные средства и другая техника для презентаций учебного материала, слайды, программы для анализа данных и обработки результатов эмпирических исследований. Практические занятия проводятся в компьютерных классах. Выполнение лабораторных работ опирается на использование пакета прикладного программного обеспечения Borland Delphi 7.0

Для оформления отчетов по лабораторным работам, пояснительной записки к контрольным работам используется пакет прикладных программ: Microsoft Office: PowerPoint 2007, Microsoft PowerPoint 2010, Microsoft Word 2007, Microsoft Word 2010, Excel 2007, Excel 2010, Visio Professional 2003, Visio Professional 2007, Visio Professional 2010.

Также для самостоятельной работы студенты могут по своему усмотрению использовать дополнительно свободно-распространяемое ПО или демо-версии.