

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Рубцовский институт (филиал) федерального государственного
бюджетного образовательного учреждения высшего профессионального
образования «Алтайский государственный университет»**



УТВЕРЖДАЮ

Директор Рубцовского
института (филиала) АлтГУ

М.П.

25 мая 2016 г.

А.И. Анисимов
А.И. Анисимов

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
ВЫСОКОУРОВНЕВЫЕ МЕТОДЫ ИНФОРМАТИКИ И
ПРОГРАММИРОВАНИЯ**

Уровень основной образовательной программы: бакалавриат

Направление подготовки: 09.03.03 Прикладная информатика

Форма обучения: очная, заочная, заочная (ускоренная) на базе ВО,
заочная (ускоренная) на базе СПО

Кафедра: Математики и прикладной информатики

**Рубцовск
2016**

При разработке рабочей программы учебной дисциплины в основу положены:

1) ФГОС ВО по направлению подготовки 09.03.03 Прикладная информатика утвержденный Министерством образования и науки РФ 12 марта 2015 г. (рег. № 207)

2) Учебный план по направлению подготовки 09.03.03 Прикладная информатика, утвержденный решением Ученого совета Рубцовского института (филиала) АлтГУ от 23 мая 2016 г., протокол № 10.

3) Рабочая программа одобрена на заседании кафедры математики и прикладной информатики от 23 мая 2016 г., протокол № 11.

Заведующий кафедрой математики
и прикладной информатики, к.т.н. _____



Е.А. Жданова

Разработчик:

старший преподаватель кафедры
математики и прикладной информатики _____



О.В. Рязанова

Работодатель:

Начальник отдела
информационно-технического
обеспечения Администрации г.
Рубцовска



Д.П. Рева

СОДЕРЖАНИЕ

I. РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ	4
1. ВВОДНАЯ ЧАСТЬ.....	4
1.1. ЦЕЛИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	4
1.2. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП УНИВЕРСИТЕТА	4
1.3 ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	4
2. ОСНОВНАЯ ЧАСТЬ.....	6
2.1. ОБЪЕМ УЧЕБНОЙ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ.....	6
2.2. СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	14
2.3. САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТА	20
2.4. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	21
2.5. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	23
2.6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ	24
2.7. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОСВОЕНИЮ УЧЕБНОЙ ДИСЦИПЛИНЫ	25
II. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ	34
3. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ УСПЕВАЕМОСТИ И РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ	35
3.1. ВИДЫ КОНТРОЛЯ И АТТЕСТАЦИИ, ФОРМЫ ОЦЕНОЧНЫХ СРЕДСТВ	35
3.2 КАРТА КОМПЕТЕНЦИЙ ДИСЦИПЛИНЫ.....	36
3.3. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ОП.....	42
3.4. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ (ФОРМЫ ОЦЕНОЧНЫХ СРЕДСТВ).....	44

I. РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

1. ВВОДНАЯ ЧАСТЬ

1.1. ЦЕЛИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Целями освоения учебной дисциплины Высокоуровневые методы информатики и программирования являются: приобретение умений и навыков методики объектно-ориентированного программирования и владения соответствующим инструментарием, приобретение умений и навыков работы с интегрированной средой delphi, которая позволяет создавать различные программы: от простейших однооконных приложений до программ управления базами данных.

1.2. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП УНИВЕРСИТЕТА

1.2.1. Учебная дисциплина «Высокоуровневые методы информатики и программирования» относится к вариативная часть.

1.2.2. Для изучения данной учебной дисциплины необходимы следующие знания, умения и навыки, формируемые предшествующими дисциплинами:

- Информатика и программирование

1.2.3. Перечень последующих учебных дисциплин, для которых необходимы знания, умения и навыки, формируемые данной учебной дисциплиной:

- Программирование на Delphi
- Проектирование информационных систем
- Проектирование обучающих систем

1.3 ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих общекультурных (ОК), общепрофессиональных (ОПК) и профессиональных (ПК) компетенций:

- способен проводить описание прикладных процессов и информационного обеспечения решения прикладных задач - ПК-7
- способен программировать приложения и создавать программные

прототипы решения прикладных задач - ПК-8

Примечание: Детальное описание компетенций – перечень компонентов, то есть требования к знаниям, умениям и навыкам, полученным в ходе изучения дисциплины, технологии формирования компетенций, формы оценочных средств, уровни освоения компетенций – должно быть изложено в Карте компетенций дисциплины.

Карта компетенций дисциплины является обязательным приложением рабочей программы дисциплины.

2. ОСНОВНАЯ ЧАСТЬ

2.1. ОБЪЕМ УЧЕБНОЙ ДИСЦИПЛИНЫ И ВИДЫ УЧЕБНОЙ РАБОТЫ

2.1.1. Тематический план учебной дисциплины (очная форма)

Дидактические единицы (ДЕ)	Наименование разделов и тем	Максимальная нагрузка студентов, час.	Количество аудиторных часов при очной форме обучения			Самостоятельная работа студентов, час.
			Лекции	Практические (семинарские) занятия	Лабораторные работы	
1	2	3	4	5	6	7
3 семестр						
<i>Происхождение, история и эволюция методологий программирования</i>						
ДЕ 1 (30 б.)	1. Основные понятия из области программирования	4	2			2
	2. Законы эволюции программного обеспечения. Основные методологии и технологии программирования. Новейшие направления в области создания технологий программирования.	6	2			4
	3. Языки и системы программирования	8	4			4
<i>Текущий контроль</i>		<i>тестирование, контрольная работа</i>				
<i>Программирование в средах современных информационных систем</i>						
ДЕ 2 (70 б.)	4. Методы проектирования алгоритмов и программ	8	4			4
	5. Элементы теории модульного программирования. Создание модульных программ.	10	2		4	4
	6. Объектно-ориентированное проектирование и	6	2			4

программирование					
7. Объектно-ориентированный подход к проектированию и разработке программ. Конструкторы и деструкторы.	4	2			2
8. Особенности программирования в оконных операционных средах. Среда разработки. Система окон разработки. Система меню.	16	4		6	6
9. Основы визуального программирования. Размещение нового компонента. Реакция на события. Компоненты. Использование компонентов.	15			10	5
10. Отладка и тестирование программ	4				4
Текущий контроль	тестирование, контрольная работа, защита лабораторных работ				
Промежуточная аттестация	экзамен				
Итого за семестр часов	108*	22		20	39
Итого за весь курс часов	108*	22		20	39
Итого за весь курс з.е.	3				

2.1.2. Тематический план учебной дисциплины (заочная форма)

Дидактические единицы (ДЕ)	Наименование разделов и тем	Максимальная нагрузка студентов, час.	Количество аудиторных часов при заочной форме обучения			Самостоятельная работа студентов, час.
			Лекции	Практические (семинарские) занятия	Лабораторные работы	
1	2	3	4	5	6	7
2 курс						
<i>Происхождение, история и эволюция методологий программирования</i>						
ДЕ 1 (30 б.)	1. Основные понятия из области программирования	12	2			10
	2. Законы эволюции программного обеспечения. Основные методологии и технологии программирования. Новейшие направления в области создания технологий программирования.	10				10
	3. Языки и системы программирования	10				10
<i>Текущий контроль</i>		<i>тестирование, контрольная работа</i>				
<i>Программирование в средах современных информационных систем</i>						
ДЕ 2 (70 б.)	4. Методы проектирования алгоритмов и программ	10				10
	5. Элементы теории модульного программирования. Создание модульных программ.	12	2			10
	6. Объектно-ориентированное проектирование и программирование	12			2	10
	7. Объектно-ориентированный подход к проектированию и разработке программ. Конструкторы и деструкторы.	12	2			10
	8. Особенности	7			2	5

	программирования в оконных операционных средах. Среда разработки. Система окон разработки. Система меню.					
	9. Основы визуального программирования. Размещение нового компонента. Реакция на события. Компоненты. Использование компонентов.	5				5
	10. Отладка и тестирование программ	9			4	5
Текущий контроль		контрольная работа, защита лабораторных работ, тестирование				
Промежуточная аттестация		экзамен				
Итого за курс часов		108*	6		8	85
Итого за весь курс часов		108*	6		8	85
Итого за весь курс з.е.		3				

2.1.3. Тематический план учебной дисциплины (заочная (ускоренная) на базе ВО форма)

Дидактические единицы (ДЕ)	Наименование разделов и тем	Максимальная нагрузка студентов, час.	Количество аудиторных часов при заочной (ускоренной) на базе ВО форме обучения			Самостоятельная работа студентов, час.
			Лекции	Практические (семинарские) занятия	Лабораторные работы	
1	2	3	4	5	6	7
1 курс						
<i>Происхождение, история и эволюция методологий программирования</i>						
ДЕ 1 (30 б.)	1. Основные понятия из области программирования	12	2			10
	2. Законы эволюции программного обеспечения. Основные методологии и технологии программирования. Новейшие направления в области создания технологий программирования.	9				9
	3. Языки и системы программирования	10				10
<i>Текущий контроль</i>		<i>тестирование, контрольная работа</i>				
<i>Программирование в средах современных информационных систем</i>						
ДЕ 2 (70 б.)	4. Методы проектирования алгоритмов и программ	8				8
	5. Элементы теории модульного программирования. Создание модульных программ.	10			2	8
	6. Объектно-ориентированное проектирование и программирование	8				8
	7. Объектно-ориентированный подход к проектированию и разработке программ.	10	2			8

Конструкторы и деструкторы.					
8. Особенности программирования в оконных операционных средах. Среда разработки. Система окон разработки. Система меню.	10			2	8
9. Основы визуального программирования. Размещение нового компонента. Реакция на события. Компоненты. Использование компонентов.	10			2	8
10. Отладка и тестирование программ	12	2		2	8
Текущий контроль	тестирование, контрольная работа, защита лабораторных работ				
Промежуточная аттестация	экзамен				
Итого за курс часов	108*	6		8	85
Итого за весь курс часов	108*	6		8	85
Итого за весь курс з.е.	3				

2.1.4. Тематический план учебной дисциплины (заочная (ускоренная) на базе СПО форма)

Дидактические единицы (ДЕ)	Наименование разделов и тем	Максимальная нагрузка студентов, час.	Количество аудиторных часов при заочной (ускоренной) на базе СПО форме обучения			Самостоятельная работа студентов, час.
			Лекции	Практические (семинарские) занятия	Лабораторные работы	
1	2	3	4	5	6	7
1 курс						
Происхождение, история и эволюция методологий программирования						
ДЕ 1 (30 б.)	1. Основные понятия из области программирования	12	2			10
	2. Законы эволюции программного обеспечения. Основные методологии и технологии программирования. Новейшие направления в области создания технологий программирования.	9				9
	3. Языки и системы программирования	10				10
Текущий контроль		тестирование, контрольная работа				
Программирование в средах современных информационных систем						
ДЕ 2 (70 б.)	4. Методы проектирования алгоритмов и программ	8				8
	5. Элементы теории модульного программирования. Создание модульных программ.	10			2	8
	6. Объектно-ориентированное проектирование и программирование	8				8
	7. Объектно-ориентированный подход к проектированию и разработке программ.	10	2			8

Конструкторы и деструкторы.					
8. Особенности программирования в оконных операционных средах. Среда разработки. Система окон разработки. Система меню.	10			2	8
9. Основы визуального программирования. Размещение нового компонента. Реакция на события. Компоненты. Использование компонентов.	10			2	8
10. Отладка и тестирование программ	12	2		2	8
Текущий контроль	тестирование, контрольная работа, защита лабораторных работ				
Промежуточная аттестация	экзамен				
Итого за курс часов	108*	6		8	85
Итого за весь курс часов	108*	6		8	85
Итого за весь курс з.е.	3				

2.2. СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.2.1. Содержание разделов учебной дисциплины

Происхождение, история и эволюция методологий программирования

Тема 1. Основные понятия из области программирования

Аудиторное изучение: Программирование как вид деятельности. Взаимосвязь программирования и других областей знания.

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.

Программа как формализованное описание процесса обработки данных. Программное средство. Жизненный цикл программного средства. Основная задача профессионального программирования. Понятие качества программного средства. Характеристики качества программного обеспечения.

Самостоятельное изучение: 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 2. Законы эволюции программного обеспечения. Основные методологии и технологии программирования. Новейшие направления в области создания технологий программирования.

Аудиторное изучение: Законы эволюции программного обеспечения. Основные методологии программирования. Атрибуты методологий. Краткая характеристика методологий: структурное программирование, объектно-ориентированное программирование, функциональное программирование, логическое программирование.

Введение в технологии программирования. Основные понятия и определения. Технология создания ПО. Процесс. Стадия. Технологический подход.

Новейшие направления в области создания технологий программирования. Case-системы. Индустрия искусственного интеллекта. Экспертные системы.

Самостоятельное изучение: 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 3. Языки и системы программирования

Аудиторное изучение: 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.

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.

Инструментальные системы программирования (инструментальные среды программирования, средства автоматизации разработки программ, интегрированные среды).

Трансляция программ. Трансляторы: интерпретаторы, компиляторы. Компилируемые и интерпретируемые языки высокого уровня. Выбор языка программирования. Критерии и факторы выбора.

Самостоятельное изучение: 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.

Программирование в средах современных информационных систем

Тема 4. Методы проектирования алгоритмов и программ

Аудиторное изучение: 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.

Самостоятельное изучение: Этапы создания программных продуктов при неавтоматизированной разработке программ. Структура программных продуктов. Проектирование интерфейса пользователя: диалоговый режим, графический интерфейс пользователя.

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

Аудиторное изучение: Метод структурного проектирования и программирования: нисходящее проектирование, модульное программирование, структурное программирование.

Понятие модуля. Характеристика модуля. Принцип модульного программирования.

Общая структура программы. Структура модуля. Заголовок модуля и связь модулей друг с другом. Интерфейсная часть. Исполняемая часть. Инициализирующая часть. Компиляция модулей. Доступ к объявленным в модуле объектам. Стандартные модули. Использование стандартных модулей.

Подпрограммы. Процедуры и функции. Формальный параметр. Фактический параметр. Передача параметров. Организация и использование подпрограмм.

Самостоятельное изучение: Функционально-модульная схема алгоритма приложения.

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

Аудиторное изучение: Основные теоретические положения ООП. Принципы ООП. Понятия класса, подкласса. Методы и классы.

Самостоятельное изучение: Достоинства и недостатки ООП. Реализация принципов ООП.

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

Аудиторное изучение: Объектно-ориентированное проектирование программ. Этапы реализации объектно-ориентированного подхода. Введение в классы.

Самостоятельное изучение: Синтаксис описания класса и подкласса. Объявление объекта. Операция new. Синтаксис описания метода. Передача параметров, перегрузка и переопределение методов. Конструкторы класса. Параметризованные конструкторы. Перегрузка конструкторов. Вложенные и внутренние классы. Абстрактные методы и классы. Класс Object. Динамическая диспетчеризация методов. Использование ключевого слова super.

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

Аудиторное изучение: Интегрированные среды Java. Знакомство с интегрированной средой разработки Eclipse, NetBeans.

Настройка интегрированной среды разработки. Проект. Характеристика проекта. Файл проекта. Параметры проекта. Создание и сохранение нового проекта. Особенности сохранения проектов. Компиляция и выполнение программы. Ошибки компиляции.

Самостоятельное изучение: Java в Интернете.

Основные стандартные модули, обеспечивающие работу в среде. Меню команд.

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

Аудиторное изучение: Введение в технологию Java. Выполнение Java программ. Особенности языка Java. Лексические основы (зарезервированные ключевые слова, идентификаторы, литеры, разделители). Синтаксис языка. Управляющие структуры. Типы данных языка Java. Понятие переменной. Объявление переменной. Целые типы данных. Приведение типов. Автоматическое преобразование типов в выражениях. Вещественные типы данных. Логический тип данных. Арифметические операции. Целочисленные битовые операции. Операции отношений. Логические операции. Операции присваивания. Условная операция. Приоритет операций. Условные операторы If-else, switch. Операторы повторения While, do-While, For. Операторы перехода Continue, Break.

Понятие и описание массива. Ссылки на массив. Элементы и длина массива. Одномерные массивы. Многомерные массивы.

Классы. Работа со строками.

Самостоятельное изучение: История создания Java. Основные области использования языка.

Тема 10. Отладка и тестирование программ

Самостоятельное изучение: 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. Тестовые данные. Обработчики исключений.

2.2.2. Лабораторный практикум

Лабораторная работа №1. Знакомство с интегрированной средой

разработки Eclipse.

Лабораторная работа №2. Знакомство с интегрированной средой разработки NetBeans.

Лабораторная работа №3,4. Прimitives типы данных и их операции

Создание первого приложения на Java. Типы данных и операции над ними. Преобразование типов данных.

Лабораторная работа № 5, 6. Управляющие операторы

Условные операторы.

Лабораторная работа № 7. Циклы

Операторы повторения и перехода.

Лабораторная работа № 8. Массивы

Одномерные и многомерные массивы.

Лабораторная работа №9,10. ООП в Java

Описание классов, методов. Конструкторы класса.

2.3. САМОСТОЯТЕЛЬНАЯ РАБОТА СТУДЕНТА

2.3.1. Виды самостоятельной работы студента

Номер ДЕ	Виды самостоятельной работы студента	Часы			
		очная	заочная	заочная (ускоренная) на базе ВО	заочная (ускоренная) на базе СПО
1	чтение студентами основной и дополнительной литературы , подготовка к тесту, подготовка к контрольной работе	10	30	29	29
2	чтение студентами основной и дополнительной литературы , подготовка к лабораторным работам, подготовка к контрольной работе, подготовка к экзамену	29	55	56	56

2.4. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.4.1 Основная литература

- 1) Давыдова Н.А. Программирование. Учебное пособие / Н.А.Давыдова. – М.:БИНОМ, Лаборатория знаний, 2012 – 239с.
- 2) Кауфман, В.Ш. Языки программирования. Концепции и принципы /В.Ш.Кауфман. – Издательство «ДМКПресс», 2010 – 464 с.
- 3) Потопахин, В. Современное программирование с нуля! /В.Потопахин.- Издательство «ДМКПресс», 2010 – 240 с.

2.4.2 Дополнительная литература

- 4) Иванова, Г.С. Технология программирования: Учебник / Г.С. Иванова. - М.: КноРус, 2011 – 336с.– 640 с.
- 5) Истомин, Е.П. Высокоуровневые методы информатики и программирования: учебник / Е.П. Истомин, В.В. Новиков, М.В. Новикова. - 3-е изд.- СПб.: ООО "Андреевский издательский дом", 2010 – 228с.
- 6) Кингсли, Х.Э. JavaScript в примерах / Х.Э.Кингсли. - Издательство: "ДМК Пресс", 2009. – 272 с.
- 7) Князева, М.Д. Алгоритмика: от алгоритма к программе: учебное пособие / М.Д. Князева. – М.: КУДИЦ-ОБРАЗ, 2006. – 192 с.
- 8) Котляров, В.П. Основы тестирования программного обеспечения: Учебное пособие / В.П. Котляров. – М.: Интернет-Университет Информационных Технологий, 2009 – 285с.
- 9) Непейвода, Н.Н. Стили и методы программирования: Курс лекций. Учебное пособие / Н.Н. Непейвода. – М.: Интернет-Университет Информационных Технологий, 2009 – 320с.
- 10) Программирование: Учебно-методическое пособие / Авт.-сост.О.В.Рязанова. – Барнаул; Рубцовск: АлтГУ, 2007 – 255с.
- 11) Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL: Джентельменский набор Web-мастера / Н.А. Прохоренок. - 3-е изд.- СПб: БХВ - Петербург, 2011 – 912с.
- 12) Свердлов, С.З. Языки программирования и методы трансляции: Учебное пособие / С.З. Свердлов. – СПб: Питер, 2007 – 638с.

13) 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.

14) Хеффельфингер, Д. Разработка приложений Java EE 6 в NetBeans 7 /Д.Хеффельфингер. - Издательство: "ДМК Пресс", 2013. – 330 с.

2.4.3 Базы данных, интернет-ресурсы, информационно-справочные и поисковые системы

15) Единое окно доступа к образовательным ресурсам. Электронная библиотека [Электронный ресурс]: инф. система. – М.: ФГАУ ГНИИ ИТТ "Информика", 2005-2016. – Режим доступа: <http://www.window.edu.ru>, свободный. – Загл. с экрана (дата обращения 27.04.2016)

16) Интернет-университет информационных технологий – дистанционное образование – INTUIT.ru [Электронный ресурс]: офиц. сайт. – М.: Открытые системы, 2003-2016. - Режим доступа: <http://www.intuit.ru>, свободный. - Загл. с экрана (дата обращения: 27.04.2016).

17) Поисковые системы: Google, Yandex, Rambler.

18) Университетская библиотека On-line [Электронный ресурс], М.: Издательство «Директ-Медиа», 2001-2016. Режим доступа: <http://www.biblioclub.ru>. – Загл. с экрана (дата обращения 27.04.2016).

19) Электронно-библиотечная система Издательство «Лань» [Электронный ресурс], СПб.: Издательство Лань, 2016. Режим доступа: <http://e.lanbook.com>. – Загл. с экрана (дата обращения 27.04.2016).

2.5. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.5.1. Требования к аудиториям (помещениям, местам) для проведения занятий:

Аудитории для проведения интерактивных занятий: видеопроектор, экран настенный, др. оборудование и компьютерный класс для проведения лабораторно-практических занятий.

2.5.2. Требования к оборудованию рабочих мест преподавателя и обучающихся:

Рабочее место преподавателя должно быть оснащено видеопроектором подключённым к компьютеру с установленным программным обеспечением указанным в пункте 2.5.4. Рабочие места обучающихся должны быть оборудованы компьютерами с установленным программным обеспечением указанным в пункте 2.5.4.

2.5.3. Требования к специализированному оборудованию отсутствуют.

2.5.4. Требования к программному обеспечению учебного процесса

Необходимое программное обеспечение
Windows 7 Professional Service Pack 1
Microsoft Excel 2010
Microsoft PowerPoint 2010
Microsoft Word 2010

2.6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

30 % – интерактивных занятий от объема аудиторных занятий (в соответствии с требованиями ФГОС)

№ ДЕ	Виды учебной работы	Образовательные технологии	Особенности проведения занятий
1	лекции	лекция-визуализация, публичная презентация	групповые
2	лекции, лабораторные работы	публичная презентация, разработка проекта, творческое задание	групповые, индивидуальные

2.7. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОСВОЕНИЮ УЧЕБНОЙ ДИСЦИПЛИНЫ

В связи с тем, что объемы информации в различных сферах деятельности человека, подлежащей хранению, растут высокими темпами, производители программного обеспечения вынуждены разрабатывать новые гибкие подходы к созданию программ и их модификации. В настоящее время при разработке сложного программного обеспечения обычно используют одну из двух технологий: структурное программирование или объектно-ориентированное программирование. Первая технология рекомендует декомпозировать программу на подпрограммы (процедуры), решающие отдельные подзадачи, т.е. базируется на процедурной декомпозиции. Вторая технология использует подход, при котором в предметной области выделяют отдельно функционирующие элементы. Поведение этих объектов программно моделируется с использованием специальных средств, а затем уже из готовых объектов собирается сложная программа. В основе второй технологии лежит объектная декомпозиция.

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 Java и сред разработки программ Eclipse и NetBeans.

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.

В процессе проведения занятий используются активные методы обучения, которые подразумевают периодическое проведение консультаций, активное участие студентов в учебном процессе в ходе выполнения практических работ, иллюстрация изучаемого теоретического материала практическими задачами и примерами, которые выдаются каждому студенту на занятии в качестве раздаточного материала.

Вся дисциплина разбита на две дидактические единицы, по итогам которой имеется промежуточная аттестация. Курс изучается в течение одного семестра, завершается – экзаменом.

Текущий контроль знаний студентов – теоретических и практических – производится в процессе защиты лабораторных работ, по результатам тестирования, выполнения контрольных работ.

Для тестирования используются тесты, размещенные в банке системы Moodle.

Критериями для оценки контрольной работы служит точность ответа на поставленные вопросы, формулировка целей и задач, раскрытие рассматриваемых понятий, четкость структуры работы, логичность изложения, наличие выводов.

Отчет по лабораторной работе оформляется в виде пояснительной записки на бумажном носителе и должен содержать название и цели работы, описание технологии работы с практическими заданиями, с заданиями для самостоятельного выполнения, творческие задания, ответы на контрольные вопросы. При защите отчета результаты, которые должны быть получены в ходе выполнения лабораторной работы, демонстрируются студентом на компьютере.

Аттестация проводится в форме экзамена (с учетом набранных баллов). Условием допуска студента к экзамену является выполнение всех лабораторных работ и их своевременная защита, а также своевременная сдача текущего контроля в виде тестирования, выполнения контрольных работ.

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.

Текущий контроль позволяет оценить знания студента по балльно-рейтинговой системе (максимальный рейтинг 100 баллов).

Оценке «отлично» соответствует рейтинг более 90 баллов, оценке «хорошо» соответствует рейтинг в диапазоне от 76 до 90 баллов, оценке «удовлетворительно» соответствует рейтинг в диапазоне от 61 до 75 баллов, оценке «неудовлетворительно» соответствует рейтинг не более 60 баллов.

Дополнительно баллы можно получить за творческие успехи и индивидуальный подход при выполнении лабораторных работ. Баллы могут быть **сняты** за пропуски занятий без уважительной причины.

Материалы текущего контроля для оценки знаний студентов по каждой дидактической единице с указанием **максимального количества баллов** представлены в виде следующей таблицы.

<p>ДЕ 1 30 баллов</p>	<p>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)</p>
<p>ДЕ 2 70 баллов</p>	<p>1. Тестирование (10 баллов) 2. Контрольная работа № 3 по теме «Обзор языка Java» (10 баллов) 3. Защита лабораторных работ (50 баллов)</p>

ОРГАНИЗАЦИЯ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

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 текущего контроля (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 (глоссария);
- выполнение творческих заданий.

Содержание самостоятельной работы студентов по дисциплине в разрезе дидактических единиц:

ДЕ 1

Содержание самостоятельной работы студентов по дисциплине в разрезе дидактических единиц:

ДЕ 1

Тема 1. 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 № 1.

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 2. 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 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. Выполнение заданий самостоятельной работы лабораторных практикумов.

ДЕ 2

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 (гlossария) 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 (гlossария) 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 (гlossария) по списку ключевых слов, понятий, терминов для тестирования.
3. Подготовка к тестированию и выполнение тестовых заданий.

Реализация принципов ООП. Этапы реализации объектно-ориентированного подхода.

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 (гlossария) 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.*

1. Работа с лекционным материалом, изучение рекомендованной литературы, самостоятельный подбор необходимой литературы, поиск необходимой информации через Интернет.
2. Составление словаря (гlossария) по списку ключевых слов, понятий, терминов для тестирования.
3. Подготовка к тестированию и выполнение тестовых заданий.
4. Выполнение творческих заданий.
5. Работа со встроенными справочными системами программных продуктов.
6. 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.*

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 (гlossария) по списку ключевых слов, понятий, терминов для тестирования.
3. Подготовка к тестированию и выполнение тестовых заданий.
4. Подготовка и выполнение контрольной работы № 3.
5. Выполнение творческих заданий.
6. Работа со встроенными справочными системами программных продуктов.
7. Доработка лабораторных работ, оформление отчетов по выполненным лабораторным работам, подготовка к защите.

Тема 10. Отладка и тестирование программ

1. Работа с лекционным материалом, изучение рекомендованной литературы, самостоятельный подбор необходимой литературы, поиск необходимой информации через Интернет.
2. Составление словаря (гlossария) по списку ключевых слов, понятий, терминов для тестирования.
3. Подготовка к тестированию и выполнение тестовых заданий.
4. Работа со встроенными справочными системами программных продуктов.
5. Доработка лабораторных работ, оформление отчетов по выполненным лабораторным работам, подготовка к защите.
6. Подготовка к экзамену.

Учебно-методическое обеспечение самостоятельной работы студентов

Для самостоятельной работы студентов используются:

- сетевые образовательные ресурсы (файл-сервер Rfagu\Public\Учебные программы), доступные только из локальной сети института;
- сеть Интернет;
- система Moodle;
- ЭБС Библиотека РИ (филиала) АлтГУ, ЭБС «Университетская библиотека он-лайн», ЭБС Центральная библиотека, ЭБС издательство «Лань», доступные из локальной сети АлтГУ;
- информационная система "Единое окно доступа к образовательным ресурсам" (window.edu.ru), которая предоставляет свободный доступ к каталогу образовательных интернет-ресурсов и полнотекстовой электронной учебно-методической библиотеке для общего и профессионального образования.

II. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине

«Высокоуровневые методы информатики и программирования»

Направление подготовки

09.03.03 Прикладная информатика

Квалификация выпускника: бакалавр

Форма обучения: очная, заочная, заочная (ускоренная) на базе ВО, заочная (ускоренная) на базе СПО

3. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ УСПЕВАЕМОСТИ И РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1. ВИДЫ КОНТРОЛЯ И АТТЕСТАЦИИ, ФОРМЫ ОЦЕНОЧНЫХ СРЕДСТВ

№ ДЕ	№ семестра	Виды контроля и аттестации	Форма оценочного средства
1	3	Текущий контроль	тестирование, контрольная работа
2		Текущий контроль	тестирование, контрольная работа, защита лабораторных работ
<i>Промежуточная аттестация: экзамен</i>			

3.2 КАРТА КОМПЕТЕНЦИЙ ДИСЦИПЛИНЫ

КАРТА КОМПЕТЕНЦИЙ ДИСЦИПЛИНЫ					
Наименование дисциплины: Высокоуровневые методы информатики и программирования					
Цель дисциплины	приобретение умений и навыков методики объектно-ориентированного программирования и владения соответствующим инструментарием, приобретение умений и навыков работы с интегрированной средой Delphi, которая позволяет создавать различные программы: от простейших однооконных приложений до программ управления базами данных.				
Задачи	<ul style="list-style-type: none"> – создание у студентов упорядоченной системы знаний о реальных возможностях программирования на языках высокого уровня, о направлениях профессионального программирования, тенденциях развития объектной технологии построения программных приложений; – формирование базы для работы в интегрированных системах программирования; – ознакомление студентов с практикой применения инструментария интегрированных сред программирования для решения различных прикладных задач. 				
В процессе освоения данной дисциплины студент формирует и демонстрирует следующие компетенции					
Компетенции					
Компетенции		Перечень компонентов	Технологии формирования	Форма оценочного средства	Уровни освоения компетенций
Индекс компетенции	Формулировка				

ПК-7	способен проводить описание прикладных процессов и информационного обеспечения решения прикладных задач	<p>Знать: законы эволюции программного обеспечения, основные методологии проектирования программ, основные понятия и определение системы программирования, структуру современной системы программирования, инструментальные системы программирования</p> <p>Уметь: выбирать методы проектирования программного обеспечения; разрабатывать программные приложения; проводить обзор, анализ и</p>	лек., лаб., сам.	ЛР, ТС,	<p>Пороговый уровень Знать: основные понятия и определение системы программирования, структуру современной системы программирования, классификацию систем программирования Уметь: разрабатывать программные приложения; проводить обзор, анализ и обоснование выбора ИКТ для решения прикладных задач. Владеть: навыками применения инструментария интегрированных сред программирования для решения различных прикладных задач</p> <p>Повышенный уровень Знать: этапы процесса разработки программных комплексов для структурного программирования и объектно-ориентированного анализа Уметь: выбирать методы проектирования программного обеспечения Владеть: навыками использования современных технологий программирования, тестирования и документирования программных комплексов, работой с</p>
------	---	---	------------------	---------	---

		<p>обоснование выбора ИКТ для решения прикладных задач</p> <p>Владеть:</p> <p>навыками оценки сложности алгоритмов и программ, использования современных технологий программирования, тестирования и документирования программных комплексов; работы с инструментальными средствами моделирования предметной области, прикладных и информационных процессов</p>			<p>инструментальными средствами моделирования предметной области, прикладных и информационных процессов</p>
--	--	---	--	--	---

ПК-8	способен программировать приложения и создавать программные прототипы решения прикладных задач	<p>Знать: законы эволюции программно о обеспечения; основные методологии проектирования программ; этапы процесса разработки программных комплексов для структурного программирования и объектно-ориентированного анализа;</p> <p>основные понятия и определение системы программирования, структуру современной системы программирования; классификацию систем программирования; инструментальные</p>	лек., лаб., сам.	ЛР, ТС,	<p>Пороговый уровень Знать: основные понятия и определение системы программирования, структуру современной системы программирования; классификацию систем программирования; инструментальные системы программирования. Уметь: выбирать методы проектирования программного обеспечения Владеть: навыками применения инструментария интегрированных сред программирования для решения различных прикладных задач</p> <p>Повышенный уровень Знать: законы эволюции программного обеспечения; основные методологии проектирования программ; этапы процесса разработки программных комплексов для структурного программирования и объектно-ориентированного анализа. Уметь: проводить анализ выбора ИКТ для решения прикладных задач Владеть: навыками оценки сложности алгоритмов и программ, использования <i>современных технологий программирования</i></p>
------	--	--	------------------	---------	--

		<p>системы программирования</p> <p>Уметь: выбирать методы проектирования программного обеспечения, разрабатывать программные приложения, проводить обзор, анализ и обоснование выбора ИКТ для решения прикладных задач</p> <p>Владеть: тестирования и документирования программных комплексов; работой с инструментальными средствами моделирования предметной области, прикладных и информационных процессов</p>			<p>современных технологий программирования</p>
--	--	--	--	--	--

		нных процессов			
--	--	-------------------	--	--	--

** Индекс и формулировка компетенции из ФГОС*

***Технологии формирования: лекция, самостоятельная работа, семинар, лабораторные работы, практические занятия, производственная практика, преддипломная практика, выполнение ВКР*

**** Форма оценочного средства: коллоквиум Кл; контрольная работа Кнр; собеседование Сб; тестирование ТС; деловая игра ДИ; ролевая игра РИ; кейс-задача КЗ; реферат Реф; эссе Э; защита лабораторные работы ЛР; портфолио Порт; круглый стол КС; дискуссия Дис; дебаты Деб; диспут Дисп; полемика Пол; разноуровневые задачи РЗ; доклад Док; сообщение Сообц; творческое задание ТЗ; курсовая работа КР; курсовой проект КП; зачет Зач; экзамен Экз;*

3.3. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ОПОП

Код и название компетенций	Наименование ДЕ и тем	Часы		Уровень освоения
		Ауди-торные	Самосто-ятельные	
ПК-7 способен проводить описание прикладных процессов и информационного обеспечения решения прикладных задач	ДЕ 1, Тема 1 Основные понятия программирования	2	2	3
	ДЕ 1, Тема 2 Законы эволюции программного обеспечения. Основные методологии и технологии программирования. Новейшие направления в области создания технологий программирования	2	4	1
	ДЕ 2, Тема 4 Методы проектирования алгоритмов и программ	2	3	1
ПК-8 способен программировать приложения и создавать программные прототипы решения прикладных задач	ДЕ 1, Тема 3 Языки и системы программирования	4	4	2
	ДЕ 2, Тема 5 Элементы теории модульного программирования. Создание модульных программ.	6	4	2
	ДЕ 2, Тема 6 Объектно-ориентированное проектирование и программирование	2	4	2
	ДЕ 2, Тема 4 Методы проектирования алгоритмов и программ	2	1	2

	ДЕ 2, Тема 7 Объектно-ориентированный подход к проектированию и разработке программ. Конструкторы и деструкторы.	2	2	3
	ДЕ 2, Тема 8 Особенности программирования в оконных операционных средах. Среда разработки. Система окон разработки. Система меню	10	6	3
	ДЕ 2, Тема 9 Основы визуального программирования. Размещение нового компонента. Реакция на события. Компоненты. Использование компонентов	10	5	3
	ДЕ 2, Тема 10 Отладка и тестирование программ		4	3

Для характеристики уровня освоения учебного материала используются следующие обозначения:

- 1. – ознакомительный (узнавание ранее изученных объектов, свойств);*
- 2. – репродуктивный (выполнение деятельности по образцу, инструкции или под руководством)*
- 3. – продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач).*

3.4. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ (ФОРМЫ ОЦЕНОЧНЫХ СРЕДСТВ)

3.4.1. Перечень оценочных средств

Вопросы для подготовки к экзамену по дисциплине «Высокоуровневые методы информатики и программирования»

Вопросы для проверки уровня обученности ЗНАТЬ

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) Объектно-ориентированное программирование. Основное понятие ООП. Объекты и классы. Определения, назначение.
- 32) Принцип инкапсуляции.
- 33) Принцип наследования.
- 34) Принцип полиморфизма.
- 35) Методы, события и свойства. Определения и назначение.
- 36) Конструкторы и деструкторы.
- 37) Типы данных. Структура типов данных.
- 38) Типы данных. Простые типы.
- 39) Типы данных. Структурированные типы.
- 40) Типы данных. Массивы.
- 41) Типы данных. Строки.
- 42) Типы данных. Целые, логические, символьные типы.
- 43) Типы данных. Вещественные типы.
- 44) Java. Синтаксис языка. Алфавит. Словарь языка. Комментарии.
- 45) Java. Условный оператор.
- 46) Java. Циклы.

Вопросы для проверки уровня обученности УМЕТЬ

47) Создание и сохранение нового проекта. Особенности сохранения проектов.

48) Компиляция и выполнение проекта.

49) Тестирование программ. Методы тестирования.

50) Отладка программ. Причины ошибок. Типы ошибок в программе: времени выполнения, синтаксические, логические. Защита от ошибок.

51) Разработка приложений. Определение функциональности приложения.

Задания для проверки уровня обученности ВЛАДЕТЬ

1) Написать программу, которая выводит на экран первую часть таблицы кодировки символов (от 0 до 127).

2) Написать программу, которая для массива $M(k)$ выполняет сдвиг элементов на D позиций влево и получает новый массив.

3) Создать приложение с использованием пользовательской процедуры, которая выводит на экран строку, состоящую из звездочек. Длина строки (количество звездочек) является параметром пользовательской функции.

4) Создать приложение, позволяющее пользователю ввести любое слово. Программа должна переставить первую букву слова на место последней буквы. При этом вторую, третью и т.д. сдвинуть влево на одну позицию.

5) Дана строка. Написать программу, которая разбивает строку на подстроки, размер которых не превышает заданного значения n .

Критерии оценивания

– **оценка «отлично»** выставляется студентам, успешно сдавшим экзамен, и показавшим глубокое знание теоретической части курса, умение проиллюстрировать изложение практическими приемами и расчетами, проявившим творческие способности в понимании, изложении и использовании учебно-программного материала, полно, подробно ответившим на вопросы билета и экзаменатора;

– **оценка «хорошо»** выставляется студентам, сдавшим экзамен с незначительными замечаниями, и показавшим глубокое знание теоретической части курса, умение проиллюстрировать изложение практическими приемами и расчетами, освоившим основную литературу, рекомендованную программой курса, обнаружившим стабильный характер знаний и способность к их самостоятельному восполнению и обновлению в ходе практической деятельности, полностью ответившим на вопросы билета и вопросы

экзаменатора, но допустившим при ответах незначительные ошибки, указывающие на наличие несистематичности и пробелов в знаниях;

– **оценка «удовлетворительно»** выставляется студентам, сдавшим экзамен со значительными замечаниями, показавшим знание основных положений теории при наличии существенных пробелов в деталях, испытывающим затруднения при практическом применении теории, допустившим существенные ошибки при ответах на вопросы билетов и вопросы экзаменатора, но показавшим знания основного учебно-программного материала в объеме, необходимом для предстоящей работы;

– **оценка «неудовлетворительно»** выставляется, если студент показал существенные пробелы в знаниях основных положений теории, которые не позволяют ему приступить к практической работе без дополнительной подготовки, не ответил на вопросы билеты или членов экзаменационной комиссии.

Комплект заданий для проведения тестирования по дисциплине «Высокоуровневые методы информатики и программирования»

Пример тестового задания №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. *Tun String предназначен для хранения:*

- строк
- целых чисел
- вещественных чисел
- массивов

3. *Tun Char предназначен для хранения:*

- строк
- целых чисел
- вещественных чисел

- отдельных символов
 - 4. С помощью оператора *WHILE .. DO*:
- три пути работы программы
- цикличную обработку по заданному условию
- не цикличную 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. Инструкция $ReadLn(F, S)$ 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. В системе программирования после обнаружения факта ошибки начинается процесс ее поиска и исправления, который называют ...

- отладкой
- компиляцией
- тестированием
- трассировка

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

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

17. Этапы разработки программной системы с использованием структурного подхода:

- постановка задачи
- анализ
- катализ
- синтез

- проектирование
- реализация
- модификация

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

- встроенными
- вложенными
- внутренними
- внешними

19. 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

Пример тестового задания №2

Требуется выбрать один ответ из предложенных вариантов

ВОПРОСЫ	ВАРИАНТЫ ОТВЕТОВ	ВЫБОР
1. Ошибка в структуре кода – это ...	1. синтаксическая ошибка 2. логическая ошибка	<input type="checkbox"/>
2. Метод, удаляющий объект – это ...	1. конструктор 2. деструктор	<input type="checkbox"/>
3. Объект – это ...	1. режим ввода информации посредством элемента управления 2. переменная, в которой хранится передаваемая в подпрограмму информация 3. структура, в которой сгруппированы данные и поведение некоторой сущности	<input type="checkbox"/>
4. Подпрограмма, которая не обязательно возвращает что-либо в вызывающую подпрограмму – это ...	1. обработчик события 2. функция 3. процедура	<input type="checkbox"/>

5. Переменная, в которой хранится передаваемая в подпрограмму информация – это ...	1. формальный параметр 2. фактический параметр	<input type="checkbox"/>
6. Фокус – это ...	1. метод, создающий новый экземпляр объекта некоторого класса 2. свойство объекта, состоящее в том, что он может принимать входную информацию 3. режим ввода информации посредством элемента управления	<input type="checkbox"/>
7. С помощью флажков (Checkbox) пользователь может выбрать ...	1. один элемент из предложенного набора 2. несколько элементов из предложенного набора	<input type="checkbox"/>
8. Именованная область памяти, в которой хранится некоторое значение, изменяемое во время выполнения программы – это ...	1. константа 2. переменная	<input type="checkbox"/>
9. Транслятор – это ...	1. часть базового ПО, поддерживающая процесс программирования 2. программный инструмент, предназначенный для перевода программ с одного формального языка на другой	<input type="checkbox"/>
10. Язык имеет машиноподобные команды и осуществляет прямой доступ к памяти	1. низкого уровня 2. алгоритмический 3. высокого уровня	<input type="checkbox"/>
11. Ключевое слово, определяющее размер и диапазон допустимых значений переменных – это ...	1. директива компилятора 2. тип данных 3. константа 4. синтаксис	<input type="checkbox"/>
12. Инициализация – это ...	1. операция объединения строк в предписанном порядке слева направо 2. присвоение переменной	<input type="checkbox"/>

	начального значения	
13. Внутренняя документация программы – это ...	1. блок-схемы 2. комментарии 3. техническое задание 4. руководство пользователя	<input type="checkbox"/>
14. Компонент, с помощью которого пользователь может инициировать выполнение некоторого фрагмента кода или целевой программы – это ...	1. надпись 2. кнопка	<input type="checkbox"/>
15. Пиктограмма – это ...	1. точка на экране, являющаяся наименьшим элементом изображения 2. прямоугольная область с рисунком	<input type="checkbox"/>
16. Соглашения об именах компонентов следующие ...	1. использовать имена, предлагаемые средой разработки 2. использовать префиксы из трех букв нижнего регистра, указывающие на тип компонента	<input type="checkbox"/>
17. Способность программы обрабатывать объект по-разному в зависимости от его класса – это ...	а) полиморфизм в) инкапсуляция с) наследование	<input type="checkbox"/>
18. Резервированное ключевое слово для инициализации объектов – это ...	а) nil в) null	<input type="checkbox"/>
19. Область видимости подпрограммы ограничена модулем, в котором она определена – это ...	а) private в) public	<input type="checkbox"/>

Критерии оценки:

- **оценка «отлично»** выставляется студенту, если он правильно отвечает более чем на 90% заданий.
- **оценка «хорошо»** выставляется студенту, если он правильно отвечает от 75% и до 90% заданий;

- **оценка «удовлетворительно»** выставляется студенту, если он правильно отвечает от 60% и до 74% заданий;
- **оценка «неудовлетворительно»** выставляется студенту, если он правильно отвечает менее чем на 60% заданий.