

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ИНГУШСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**



УТВЕРЖДАЮ

Проректор по учебной работе

Василий 2018.03.01 Ф.И.О.

2018 2018г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Программирование

(наименование дисциплины)

Основной профессиональной образовательной программы

академического бакалавриата

(академического (ой)/прикладного (ой) бакалавриата/магистратуры)

03.03.02 «Физика»

(код и наименование направления подготовки/специальности)

(наименование профиля подготовки (при наличии))

Квалификация выпускника

бакалавр

Форма обучения

очная

(очная, заочная)

МАГАС, 2018 г.

1. Цели и задачи освоения учебной дисциплины

Цель – выработка у студентов знаний, умений и навыков, связанных с созданием современного программного обеспечения, ознакомление с современными языками программирования, объектно-ориентированной парадигмой программирования, стандартными объектными библиотеками и интегрированными средами разработки. Для достижения поставленных целей идеально подходит платформа .NET Framework и языки, разработанные для этой платформы, в частности, язык Visual C++.

Задачи:

. изучение базовых типов данных и управляющих операторов, имеющиеся в C-подобных языках, и в частности, в языке C++;

•

изучение особенностей традиционных структур данных, связанных с объектной природой стандартных библиотек .NET;

•

освоение объектно-ориентированной парадигмы программирования, в том числе таких ее важнейших концепций, как инкапсуляция, наследование, полиморфизм, и особенностей объектной модели для платформы .NET;

•

овладение дополнительными возможностями, связанными с объектной моделью, а именно механизмом обработки исключений, созданием и использованием свойств и событий, применением обобщенных классов;

•

овладение основами технологии визуального проектирования и освоение связанной с ней библиотеки Microsoft Windows Forms для .NET.

2. Место учебной дисциплины в структуре опов во

Дисциплина «Программирование» реализуется в рамках базовой части Блока 1 «Дисциплины (модули)» программы бакалавриата.

Для освоения данной дисциплины необходимы компетенции, сформированные в рамках освоения дисциплин модуля «Общая физика», «Математика».

Дисциплина изучается на 2 курсе в 3 семестре.

Связь дисциплины «Программирование» с предшествующими дисциплинами и сроки их изучения

Код дисциплины	Дисциплины, предшествующие дисциплине «Программирование»	Семестр
Б1.Б6	Информатика	2

Связь дисциплины «Компьютерные науки» с последующими дисциплинами и сроки их изучения

Код дисциплины	Дисциплины, следующие за дисциплиной «Программирование»	Семестр
Б1.Б11	«Методы вычислений»	5

3. КОМПЕТЕНЦИИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Код компетенции	Результаты освоения ООП Содержание компетенций	Перечень планируемых результатов обучения по дисциплине
ОПК-4	Способность понимать сущность и значение информации в развитии современного общества, сознавать опасности и угрозы, возникающие в этом процессе, соблюдать основные требования информационной безопасности	<p>Знать: 1. Основы функционирования локальных и глобальных компьютерных сетей; 2. Основы функционирования компьютерных вирусов; 3. Основные принципы информационной безопасности.</p> <p>Уметь: 1. Использовать антивирусные программные продукты.</p> <p>Владеть: 1. Навыками обеспечения информационной безопасности.</p>
ОПК-5	Способность использовать основные методы, способы и средства получения, хранения, переработки информации и навыки работы с компьютером как со средством управления информацией	<p>Знать: 1. Виды и типы компьютерных данных; 2. Основные типы алгоритмов; 3. Принципы и приемы использования подпрограмм; 4. Парадигмы программирования; 5. Принципы объектно-ориентированного программирования; 6. Сравнительные характеристики текстового и графического режимов экрана; 7. Принципы построения графических примитивов; 8. Принцип изображения движущихся объектов; 9. Основные положения теории информации; 10. Принципы построения систем обработки и передачи информации; 11. Виды современного программного обеспечения, используемого в профессиональной деятельности; 12. Основы языка программирования Паскаль.</p> <p>Уметь: 1. Создавать собственные программы, реализующие линейные алгоритмы; 2. Создавать собственные программы, реализующие ветвящиеся алгоритмы; 3. Создавать собственные программы, реализующие циклические алгоритмы; 4. Создавать собственные программы, реализующие алгоритмы работы с массивами; 5. Создавать собственные программы, содержащие подпрограммы; 6. Создавать собственные программы, изображающие несложные движущиеся объекты; 7. Использовать информационные технологии для решения физических задач 8. Разрабатывать собственные несложные программы для решения профессиональных задач; 9. Скачивать информацию с ресурсов сети Интернет; 10. Отлаживать программы, используя возможности ИСР.</p>

<i>Код компетенции</i>	Результаты освоения ООП <i>Содержание компетенций</i>	Перечень планируемых результатов обучения по дисциплине
		Владеть: 1. Навыками проектирования и разработки программ на языке программирования Паскаль; 2. Навыками использования информационных технологий для решения физических задач; 3. Навыками программной реализации моделей физических явлений. 4. Приемами работы с ИСР.

3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся

3.1. Объем дисциплины по видам учебных занятий (в часах)

Вид учебной работы	Всего часов
Общая трудоемкость дисциплины	
Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)	74
Аудиторная работа (всего):	56
в том числе:	
лекции	20
практические занятия	
лабораторные работы	
в т.ч. в активной и интерактивной формах	36
Внеаудиторная работа (всего):	–
Самостоятельная работа обучающихся (всего)	18
Вид промежуточной аттестации обучающегося - зачет	–

Содержание дисциплины, структурированное по разделам (темам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах)

№ п/п	Раздел дисциплины	трудоемкость (в часах) Общая	Виды учебных занятий, включая самостоятельную работу обучающихся и трудоемкость (в часах)			Формы текущего контроля успеваемости
			аудиторные учебные занятия		самостоятельная работа обучающихся	
			лекции	лабораторные занятия		
		всего				
1.	Аппаратное и программное обеспечение компьютера		8	4	4	Компьютерный тест
2.	Язык Паскаль		12	32	14	Решение индивидуальных задач Компьютерный тест
			20	36	18	

4.2. Содержание дисциплины, структурированное по темам (разделам)

№	Наименование раздела дисциплины	Содержание
1	Аппаратное и программное обеспечение компьютера	
<i>1.1</i>	<i>Содержание лекционного курса</i>	
1.	Введение в системы обработки и передачи информации (2 часа)	Количество и качество информации. Сообщения и сигналы. Кодирование и квантование сигналов. Информационный процесс в автоматизированных системах. Фазы информационного цикла и их модели. Информационный ресурс и его составляющие. Информационные технологии.
2	Программное обеспечение (ПО) компьютера (2 часа)	Программное обеспечение (ПО) компьютера. Классификация ПО. Операционные системы. Трансляторы. Интегрированные среды разработки программ (ИСР), системы программирования
3	Локальные и глобальные сети. Основы Информационной безопасности(2 часа)	Локальные и глобальные сети. Основы информационной безопасности. Анализ угроз информационной безопасности. Методы и средства обеспечения информационной безопасности. Классификация основных атак и вредоносных программ.
4	Операционная система (ОС) компьютера (2 часа)	Операционные системы. Приемы работы в Интегрированной среде разработки программ (ИСР). Транслятор. Компоновщик. Структура программы на языке Паскаль.
2	Язык Паскаль	
	<i>Содержание лекционного курса</i>	
5	Принципы	Характеристика языка. Структура программы. Принцип

№	Наименование раздела дисциплины	Содержание
	структурного программирования. Алгоритмы (2 часа)	структурного программирования. Алгоритмы, Блок-схемы.
6	Программирование линейных алгоритмов (2 часа)	Виды данных. Типы данных. Операции, выражения. Оператор присваивания. Ввод-вывод данных. Ввод данных с клавиатуры. Форматированный вывод данных простых типов на экран.
7	Программирование ветвящихся алгоритмов (2 часа)	Особенности ветвящихся алгоритмов. Логические выражения. Условный оператор. Составной оператор. Оператор выбора.
8	Программирование циклических алгоритмов (2 часа)	Особенности циклических алгоритмов. Счетный цикл, циклы с пред и пост условием. Программирование вывода таблиц и расчета сумм рядов.
9	Компьютерная графика (2 час)	Модули GRAPH, CRT. Компьютерная анимация. Интерактивная графика.
10	Массивы (2 часа)	Одномерные и двумерные массивы. Ввод-вывод, использование массивов.
2.2 Темы лабораторных занятий		
1	Процедуры и функции Модульность программ	Модульность программ. Процедуры и функции. Параметры подпрограмм: формальные и фактические, параметры-значения и параметры-переменные. Глобальные и локальные переменные.
2	Записи, работа с файлами	Записи. Оператор присоединения. Файлы. Типы файлов. Работа с файлами.
3	Парадигмы программирования	Парадигмы программирования. Понятие о функциональном и логическом программировании.
4	Объекты. Принципы ООП	Введение в объектное программирование. Объекты. Принципы ООП.
5	Программирование линейных алгоритмов	Виды данных. Типы данных. Операции, выражения. Оператор присваивания. Ввод-вывод данных. Ввод данных с клавиатуры. Форматированный вывод данных простых типов на экран.
6	Программирование ветвящихся алгоритмов	Ветвящиеся алгоритмы. Логические выражения. Условный оператор. Составной оператор. Оператор выбора.
7	Программирование циклических алгоритмов	Циклические алгоритмы. Счетный цикл, циклы с пред и пост условием. Программирование вывода таблиц и расчета сумм рядов.
8	Компьютерная графика	Модули GRAPH, CRT. Особенности графического режима экрана. Работа с графическими примитивами.
9	Интерактивная график	Принципы создания компьютерной анимации. Интерактивная графика.
10	Массивы Одномерные и двумерные	Одномерные и двумерные массивы. Ввод-вывод массивов. Поиск в массиве. Обработка элементов массива.
11	Процедуры и функции	Процедуры и функции. Передача данных между программными блоками с помощью параметров и имени функций

12	Параметры подпрограмм.	Параметры подпрограмм: формальные и фактические, параметры-значения и параметры-переменные. Глобальные и локальные переменные.
----	------------------------	--------------------------------------------------------------------------------------------------------------------------------

5. Образовательные технологии

Образовательный процесс по дисциплине организован в форме учебных занятий (контактная работа (аудиторной и внеаудиторной) обучающихся с преподавателем и самостоятельная работа обучающихся). Учебные занятия представлены следующими видами, включая учебные занятия, направленные на проведение текущего контроля успеваемости:

- лекции (занятия лекционного типа);
- семинары, практические занятия (занятия семинарского типа);
- групповые консультации;
- индивидуальные консультации и иные учебные занятия, предусматривающие индивидуальную работу преподавателя с обучающимся;
- самостоятельная работа обучающихся;
- занятия иных видов.

На учебных занятиях обучающиеся выполняют запланированные настоящей программой отдельные виды учебных работ. Учебное задание (работа) считается выполненным, если оно оценено преподавателем положительно.

В рамках самостоятельной работы обучающиеся осуществляют теоретическое изучение дисциплины с учётом лекционного материала, готовятся к практическим занятиям, выполняют домашнее задания, осуществляют подготовку к промежуточной аттестации.

Содержание дисциплины, виды, темы учебных занятий и форм контрольных мероприятий дисциплины представлены в разделе 5 настоящей программы и фонде оценочных средств по дисциплине.

Текущая аттестация по дисциплине (модулю). Оценивание обучающегося на занятиях осуществляется в соответствии с положением о текущей аттестации обучающихся в университете.

По итогам текущей аттестации, ведущий преподаватель (лектор) осуществляет допуск обучающегося к промежуточной аттестации.

Допуск к промежуточной аттестации по дисциплине (модулю). Обучающийся допускается к промежуточной аттестации по дисциплине в случае выполнения им всех заданий и мероприятий, предусмотренных настоящей программой дисциплины в полном объеме. Преподаватель имеет право изменять количество и содержание заданий, выдаваемых обучающимся (обучающемуся), исходя из контингента (уровня подготовленности).

Допуск обучающегося к промежуточной аттестации по дисциплине осуществляет преподаватель, ведущий семинарские (практические) занятия.

Обучающийся, имеющий учебные (академические) задолженности (пропуски учебных занятий, не выполнивший успешно задания(е)) обязан отработать их в полном объеме.

Отработка учебных (академических) задолженностей по дисциплине (модулю). В случае наличия учебной (академической) задолженности по дисциплине, обучающийся отрабатывает пропущенные занятия и выполняет запланированные и выданные преподавателем задания. Отработка проводится в период семестрового обучения или в период сессии согласно графику (расписанию) консультаций преподавателя.

Обучающийся, пропустивший *лекционное занятие*, обязан предоставить преподавателю реферативный конспект соответствующего раздела учебной и монографической литературы (основной и дополнительной) по рассматриваемым вопросам в соответствии с настоящей программой.

Обучающийся, пропустивший *практическое занятие*, отрабатывает его в форме реферативного конспекта соответствующего раздела учебной и монографической литературы (основной и дополнительной) по рассматриваемым на *практическом* занятии вопросам в соответствии с настоящей программой или в форме, предложенной преподавателем. Кроме того, выполняет все учебные задания. Учебное задание считается выполненным, если оно оценено преподавателем положительно.

Преподаватель имеет право снизить бальную (в том числе рейтинговую) оценку обучающемуся за невыполненное в срок задание (по неуважительной причине).

Промежуточная аттестация по дисциплине (модулю). Формой промежуточной аттестации по дисциплине определен Зачет.

Промежуточная аттестация обучающихся осуществляется в соответствии с положением о промежуточной аттестации обучающихся в университете и оценивается: *на зачете – зачтено; незачтено* и рейтинговых баллов, назначаемых в соответствии с принятой в вузе бально-рейтинговой системой.

Зачет принимает преподаватель, ведущий семинарские (практические) занятия по курсу. Оценка знаний обучающегося оценивается по критериям, представленным в фонде оценочных средств по дисциплине.

6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Таблица 5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

№ раздела	Наименование раздела	Содержание средств контроля (вопросы самоконтроля)	Учебно-методическое обеспечение*
1.	Программное обеспечение (ПО) компьютера (2 часа)	Работа с литературой. Решение задач Подготовка к практическим занятиям по вопросам, предложенным преподавателем Подготовка реферата	О: [1-1] Д: [1-3]
2.	Локальные и глобальные сети. Основы Информационной безопасности(2 часа)	Подготовка к практическим занятиям по вопросам, предложенным преподавателем Подготовка реферата Подготовка к вопросам промежуточной аттестации, связанных с темой	О: [1-1] Д: [1-3]
3.	Операционная система (ОС) компьютера (2 часа)	Проверочный тест по предыдущему модулю. Работа с литературой. Подготовка к контрольной работе	О: [1-1] Д: [1-3]
4.	Принципы структурного программирования. Алгоритмы (2 часа)	Работа с литературой. Решение задач Подготовка к практическим занятиям по вопросам, предложенным преподавателем Подготовка к вопросам промежуточной аттестации, связанных с темой. Написание и отладка программ.	О: [1-1] Д: [1-3]
5.	Программирование линейных алгоритмов (2 часа)	Работа с литературой. Решение задач Подготовка к практическим занятиям по вопросам, предложенным преподавателем Подготовка к вопросам промежуточной аттестации, связанных с темой. Написание и отладка программ. Подготовка к вопросам промежуточной аттестации, связанных с темой	О: [1-1] Д: [1-3]
6.	Программирование ветвящихся алгоритмов (2 часа)	Проверочный тест по предыдущему модулю. Работа с литературой. Решение задач. Подготовка к контрольной работе	О: [1-1] Д: [1-3]
7.	Программирование циклических алгоритмов (2 часа)	Работа с литературой. Решение задач Подготовка к практическим занятиям по вопросам, предложенным преподавателем	О: [1-1] Д: [1-3]

		Подготовка реферата	
8.	Компьютерная графика (2 час)	Работа с литературой. Решение задач Подготовка к практическим занятиям по вопросам, предложенным преподавателем Подготовка к вопросам промежуточной аттестации, связанных с темой. Написание и отладка программ.	О: [1-1] Д: [1-3]
9.	Массивы (2 часа)	Работа с литературой. Решение задач Подготовка к практическим занятиям по вопросам, предложенным преподавателем Подготовка реферата	О: [1-1] Д: [1-3]

Примечание: О: – основная литература, Д: – дополнительная литература; в скобках – порядковый номер по списку

Для более углубленного изучения темы задания для самостоятельной работы рекомендуется выполнять параллельно с изучением данной темы. При выполнении заданий по возможности используйте наглядное представление материала. Система накопления результатов выполнения заданий позволит вам создать копилку знаний, умений и навыков, которую можно использовать как при прохождении практики, так и в будущей профессиональной деятельности.

7. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

ФОС приведен ниже

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

а) основная учебная литература:

1. Алексеев Е.Р. Free Pascal и Lazarus: Учебник по программированию / Е.Р. Алексеев, О.В. Чеснокова, Т.В. Кучер. – М.: "ДМК Пресс", - 2010. - 438 с. (URL: http://e.lanbook.com/books/element.php?pl_1_id=1267, дата обращения 17.01.2015)

б) дополнительная учебная литература:

1. Зеленьяк О.П. Практикум программирования на Turbo Pascal. Задачи, алгоритмы и решения / О.П. Зеленьяк, 3-е изд., исправленное и дополненное. – М.: "ДМК Пресс", - 2009. - 320 с. (URL: http://e.lanbook.com/books/element.php?pl_1_id=1249, дата обращения 17.01.2015)
2. Безручко В.Т. Информатика : (курс лекций): учеб. пособие / В. Т. Безручко.- М. : ФОРУМ - ИНФРА-М , 2009.
3. Программирование. Базовый курс : учебный курс.- М. , 2005 - Университетская серия.

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»), необходимых для освоения дисциплины

1. Сайт физического факультета КемГУ, страница Павловой Т.Ю. (URL: http://physic.kemsu.ru/viewpage.php?page_id=108, дата обращения 17.01.2015)
2. URL: <http://www.sibstrin.ru/student/books/programming/>, дата обращения 17.01.2015
3. URL: <http://www.cyberguru.ru/programming/pascal/>, дата обращения 17.01.2015
4. URL: <http://tp7.info/index.php>, дата обращения 17.01.2015

9. Методические указания для обучающихся по освоению дисциплины

В курсе «Программирование» студент-физик должен приобрести теоретические и практические навыки обработки информации с помощью компьютера, написания относительно простых программ для использования в профессиональной деятельности.

На лекциях сначала излагается обзорный материал, посвященный операционным системам (в основном Windows), программному обеспечению компьютера. Затем излагаются основы языка Паскаль в следующем порядке:

- 1) Данные, типы данных
- 2) Простые типы данных
- 3) Виды данных
- 4) Основные операторы языка Паскаль
- 5) Массивы и работа с ними
- 6) Подпрограммы. Виды подпрограмм языка Паскаль
- 7) Другие типы данных
- 8) ООП

Порядок изложения материала на лекциях соответствует логике построения языка программирования, выявлению внутренних взаимосвязей данных и команд языка, которые их обрабатывают. Однако, такое изложение не способствует быстрому получению практических навыков и не соответствует порядку изучения языка на практических занятиях. Поэтому сначала материал лекций «отстает» от нужд практических занятий. Для ускорения темпа подачи лекционного материала часть его, упорядоченная в виде таблиц и примеров, рассматривается бегло и не записывается студентами на лекции. Этот материал размещен в кратких конспектах, выставленных на сайте факультета.

Из-за недостатка времени, на лекции рассматривается достаточно небольшое число примеров использования данных, операторов, реализации алгоритмов на языке Паскаль. Большое число дополнительных примеров можно найти в учебных пособиях «Программирование на языке Паскаль для начинающих», «Структурное программирование В ИСР «Free Pascal» и в электронном учебном пособии «Прикладное программирование на языке Паскаль», размещенных на сайте факультета.

На лабораторных занятиях студент решает индивидуальные задачи по теме занятия (разрабатывает и отлаживает программы). В процессе изучения тем курса, каждый студент получает индивидуальные задачи физического или математического содержания, которые должен решить под руководством преподавателя. На аудиторных лабораторных занятиях преподаватель дает общие указания по решению задач, консультирует студентов. Далее проводится защита решенной задачи.

Для оптимальной разработки полноценной программы, необходимо последовательно пройти несколько этапов. Нарушение порядка выполнения работ, пропуск этапов, приводит обычно к появлению в программе трудно устранимых логических ошибок, потерям времени, необходимости возвращаться к уже проделанным этапам работы. Поэтому абсолютно необходимо вести разработку программы следующим образом:

1 этап – разработка алгоритма

На этом этапе необходимо определить имена и типы данных, которые будут участвовать в программе, порядок действий программы (алгоритм), разработать контрольный пример для программы, который поможет подтвердить правильность ее результатов.

2 этап – разработка программы

Теория программирования предлагает несколько подходов к разработке программ. Небольшие программы лучше писать, руководствуясь принципами структурного программирования.

Структурное программирование - технология разработки программ, основанная на принципах:

1. использование трёх базовых алгоритмических структур: следование, ветвление, цикл;
2. программирование "сверху-вниз";
3. использование модулей.

Способ программирования «сверху-вниз» заключается в том, что задача решается от общего к частному, крупные задачи разбиваются на более мелкие и т.д. пока очередные задачи смогут быть реализованы одним или несколькими простыми операторами языка программирования. При этом, если какие-то фрагменты программы повторяются или представляют собой отдельные, логически цельные задачи, они оформляются в виде отдельных модулей (подпрограмм).

3 этап – набор программы в окне редактора

Разработанную программу необходимо набрать с помощью клавиатуры, а затем записать на диск с помощью меню file/save as в виде имя.pas (вместо слова «имя» нужно набрать имя программы латинскими буквами). Если это не сделано, при запуске программы, ИСР потребует записать ее на диск (откроет соответствующее окно). Но лучше сохранять текст программы периодически, в процессе набора, нажимая клавишу F2.

4 этап – компиляция программы

Компилятор - программа, которая осуществляет перевод исходной программы в объектную программу на языке машинных команд или ассемблера.

Результатом работы компилятора будет являться программа (так называемый «объектный код», имеющий обычно расширение .OBJ) или сообщение об ошибках в исходной программе. Объектный код не является готовой для исполнения программой, требуется еще работа компоновщика.

Также компилятор исправляет некоторые ошибки или делает сообщения об ошибках, которые выводятся в окно сообщений компилятора. По умолчанию это окно появляется под окном редактора вместе с сообщением о наличии ошибок в программе. Нажав два раза клавишу ввода (Enter), в окне редактора можно увидеть сообщение о первой ошибке, и система выставит курсор в то место в программе, где эта ошибка возникает. Если окна компилятора не видно, можно нажать клавишу F6. Если окно компилятора и

в этом случае не появится, вызовите список доступных окон командой Alt+O, выделите Compiler Messages, нажмите кнопки Show и ОК для того, чтобы сделать окно компилятора видимым.

5 этап – компоновка с библиотеками подпрограмм и запуск программы на выполнение

4 и 5 этапы производятся по команде run (ctrl+F9), их можно проводить и отдельно.

6 этап – отладка программы

Синтаксические ошибки выявляются на этапе компиляции. Однако, в программе может содержаться некоторое количество логических ошибок, когда компьютер выдает неправильные результаты, из-за того, что в программе записан неверный алгоритм. Способы выявления логических ошибок описаны далее.

7 этап – анализ результатов, тестирование

На этом этапе необходимо протестировать результаты работы программы. Для этого используется один или несколько контрольных примеров. Если программа имеет ветвления, нужно подобрать контрольные примеры для каждой ветви. Если контрольные примеры выполняются неправильно, придется возвратиться на предыдущие этапы.

Теория и необходимые приемы решения основных задач изложены в пособии «Структурное программирование в ИСП Free Pascal» (в электронном виде URL: http://physic.kemsu.ru/pub/library/learn_pos/FreePascal.PDF, дата

обращения 17.01.2015), в котором имеется большое количество примеров, иллюстрирующих теоретический материал и приемы решения индивидуальных задач. Готовые примеры полезно набрать и запустить на выполнение в Интегрированной среде разработки программ Free Pascal (ИСП), чтобы осознать влияние отдельных переменных и операторов на особенности выполнения программ.

Сложным вопросам программирования посвящены имеющиеся видеофрагменты и интерактивные тренинги, имеющиеся в пособии. Для уяснения деталей, их нужно просмотреть или проделать несколько раз, набрать и отладить пример показанной в фрагменте программы в ИСП самостоятельно. В тренингах также имеются задачи, решения которых автоматически проверяются.

По темам, не отраженным в программе практических занятий проводится компьютерное тестирование во время контрольных точек. Для контроля теоретических вопросов в тесты включены задания и по пройденным на практических занятиях темам.

Кроме того, для студентов, не защитивших индивидуальные задачи в семестре, проводится дополнительное тестирование по соответствующим темам.

Подготовка к тестированию должна состоять из изучения теории, примеров и прохождения пробных тестов в электронном учебном пособии «Прикладное программирование на языке Паскаль». Очень хороший эффект дает самостоятельное воспроизведение разобранных примеров и сравнение их с оригиналом, а также работа с имеющимися в вышеупомянутом пособии тренингами.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем

1. Демонстрация теоретического материала в виде слайд-лекций, выделение наиболее важной части материала с помощью инструментов программного обеспечения интерактивного планшета (перо, экран, лупа);
2. Использование при чтении лекций демонстраций выполнения программ на языке Паскаль, в Интегрированной среде разработки программ Free Pascal;
3. Проведение вычислений с помощью Интегрированной среды разработки программ Free Pascal;
4. Скачивание и (или) изучение учебно-методических материалов сайтов (протокол http); изучение электронных учебно-методических материалов в формате текстовых редакторов пакета офисных программ Open Office.org, пакета MS Office 2007 или пакета MS Office 2003.
5. Отправка разработанных электронных документов на электронную почту преподавателю.

Технологии, используемые при активной и интерактивной формах обучения

1. Технология концентрированного обучения (лекции-беседы, лекции-визуализации с использованием элементов анимации).
2. Технология активного (контекстного) обучения (моделирование предметного и социального содержания будущей профессиональной деятельности).
3. Технология дифференцированного, творческо-репродуктивного обучения (индивидуальное выполнение практических заданий при взаимодействии студента и преподавателя).

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Техническое обеспечение

Компьютер класса IBM PC не ниже Pentium 1800 МГц, оперативная память не менее 512 Мб RAM.

Программное обеспечение

Интегрированная среда разработки программ Free Pascal (свободно распространяемое ПО).

12. Иные сведения и (или) материалы

12.1. Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья

Реализация дисциплины осуществляется в инклюзивных общих группах.

Организационно-педагогическое сопровождение студентов-инвалидов и студентов с ограниченными возможностями здоровья (ОВЗ) направлено на контроль освоения образовательной программы в соответствии с графиком учебного процесса и типовым или индивидуальным учебным планом и включает в себя, при необходимости, контроль за посещаемостью занятий, помощь в организации самостоятельной работы, организацию индивидуальных консультаций, контроль по результатам текущего контроля успеваемости, промежуточной аттестации.

На основе индивидуализированного подхода (индивидуализация содержания, методов, темпа учебной деятельности, внесения, при необходимости, требуемых корректировок в деятельность обучающегося и преподавателя) организуется проведение лекционных и лабораторных занятий для студентов с ОВЗ.

В процессе обучения возможно использование различных форм организации off-line занятий (обсуждение вопросов освоения дисциплины в рамках форумов, блогов, через электронную почту).

В работах задействована Интегрированная среда разработки программ Free Pascal (свободно распространяемое ПО), позволяющая дистанционно выполнять лабораторные работы.

По дисциплине разработан учебно-методический комплекс, включающий мультимедийные презентации лекций, учебно-методические разработки и методические рекомендации для самостоятельного освоения курса.

В перечень основной и дополнительной литературы входят издания, размещенные в электронных библиотечных системах.

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

5.1. Учебно-методические пособия

1. Программирование на языке Паскаль для начинающих (в электронном виде
<http://physic.kemsu.ru/pub/library/recomend/informatika/PosobiePascal.pdf>
(дата обращения 17.01.2015) и на бумажном носителе)
2. Задачи по программированию для физиков (в электронном виде
<http://physic.kemsu.ru/pub/library/recomend/informatika/Zadachi123.doc>,
дата обращения 17.01.2015)
3. Структурное программирование в ИСР «Free Pascal» (в электронном виде
http://physic.kemsu.ru/pub/library/learn_pos/FreePascal.PDF (дата
обращения 17.01.2015) и на бумажном носителе)
4. Прикладное программирование на языке Паскаль (в электронном виде
http://physic.kemsu.ru/pub/library/learn_pos/Free_Pascal/index.htm, дата
обращения 17.01.2015)

5.2. Накопители тестовых заданий

1. НТЗ «Программирование» (пакет программ АСТ)
2. Набор тестов в пособии «Прикладное программирование на языке Паскаль»(вэлектронномвиде
http://physic.kemsu.ru/pub/library/learn_pos/Free_Pascal/index.htm, дата
обращения 17.01.2015)

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

а) основная учебная литература:

2. Алексеев Е.Р. Free Pascal и Lazarus: Учебник по программированию / Е.Р. Алексеев, О.В. Чеснокова, Т.В. Кучер. – М.: "ДМК Пресс", - 2010. - 438 с. (URL: http://e.lanbook.com/books/element.php?pl_1_id=1267, дата обращения 17.01.2015)

б) дополнительная учебная литература:

2. Зеленьяк О.П. Практикум программирования на Turbo Pascal. Задачи, алгоритмы и решения / О.П. Зеленьяк, 3-е изд., исправленное и дополненное. – М.: "ДМК Пресс", - 2009. - 320 с. (URL: http://e.lanbook.com/books/element.php?pl_1_id=1249, дата обращения 17.01.2015)
3. Безручко, Валерия Тимофеевна. Компьютерный практикум по курсу "Информатика" [Текст] : учеб. пособие / В. Т. Безручко. - 3-е изд., перераб.

доп. - М. : ФОРУМ - ИНФРА-М, 2009. - 367 с.

4. Безручко В.Т. Информатика : (курс лекций): учеб. пособие / В. Т. Безручко.- М. : ФОРУМ - ИНФРА-М , 2009.

5. Программирование. Базовый курс : учебный курс.- М. , 2005 - Университетская серия.

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»), необходимых для освоения дисциплины

1. Сайт физического факультета КемГУ, страница Павловой Т.Ю. (URL: http://physic.kemsu.ru/viewpage.php?page_id=108, дата обращения 17.01.2015)
2. URL: <http://www.sibstrin.ru/student/books/programming/>, дата обращения 17.01.2015
3. URL: <http://www.cyberguru.ru/programming/pascal/>, дата обращения 17.01.2015
4. URL: <http://tp7.info/index.php>, дата обращения 17.01.2015

6.1. Паспорт фонда оценочных средств по дисциплине

№ п/п	Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части)	Наименование оценочного средства
1.	Программное обеспечение (ПО) компьютера	ОПК-4, ОПК-5	Компьютерный тест
2.	Локальные и глобальные сети. Основы информационной безопасности	ОПК-4	Компьютерный тест
3.	Программирование линейных алгоритмов	ОПК-5	Набор индивидуальных задач Компьютерный тест
4.	Программирование ветвящихся алгоритмов	ОПК-5	Набор индивидуальных задач Компьютерный тест
5.	Программирование циклических алгоритмов	ОПК-5	Набор индивидуальных задач Компьютерный тест
6.	Компьютерная графика	ОПК-5	Набор индивидуальных задач
7.	Массивы	ОПК-5	Набор индивидуальных

РПД «Программирование»

№ п/п	Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части)	Наименование оценочного средства
			задач Компьютерный тест
8.	Процедуры и функции	ОПК-5	Набор индивидуальных задач Компьютерный тест
9.	Парадигмы программирования	ОПК-5	Компьютерный тест

6.2. Типовые контрольные задания или иные материалы

6.2.1. Примеры индивидуальных задач

По теме «Программирование линейных алгоритмов»

1. Два шарика с зарядами q_1 Кл и q_2 Кл находятся на расстоянии r см друг от друга в среде с диэлектрической проницаемостью ϵ . Написать программу, которая рассчитывает силу, которая действует на каждый шарик.
2. Компьютер должен сообщить космонавту его вес на поверхности Земли, Луны и Марса. Масса космонавта m вводится с клавиатуры. Данные для планет приведены в таблице.

По теме «Программирование ветвящихся алгоритмов»

1. Даны объем и масса тела. Программа должна определить, будет ли это тело плавать в воде.
2. Написать программу, которая находит сопротивление участка цепи, составленного из сопротивлений R_1 и R_2 . Предусмотреть два случая: последовательное и параллельное соединения. R_1 , R_2 и номер случая вводить с клавиатуры.

По теме «Программирование циклических алгоритмов»

1. Написать программу, которая выводит на экран таблицу зависимости изменения относительного числа атомов радона N/N_0 от времени в интервале от $0 < t < m$ суток через каждые сутки. Период полураспада $T=4$ суток, в начальный момент времени $N/N_0=1$. Рассчитать среднее значение относительного числа атомов радона в период от $4x$ по $8x$ сутки.
2. Написать программу для вычисления числа π с помощью ряда:

$$\pi = 4(1 - 1/3 + 1/5 - 1/7 + 1/9 - \dots)$$

По теме «Компьютерная графика»

1. Разработать программу, которая рисует шкалу амперметра. Три клавиши позволяют останавливать его стрелку, смещать ее вправо и влево, соответственно.
2. Разработать программу, которая рисует несколько ионов и электроды внутри жидкости в сосуде. При нажатии на заданную клавишу, ионы приходят в движение.

По теме «Массивы»

1. Написать программу, которая по названию элемента таблицы Менделеева определяет его номер и электронную конфигурацию.

2. В массиве хранятся C_1, C_2, \dots, C_n - емкости n конденсаторов. Определить емкость последовательного и параллельного соединения конденсаторов.

По теме «Процедуры и функции»

1. Написать программу пересчета декартовых координат точки (x, y, z) в цилиндрические (ρ, ϕ, z) . Основная программа вводит и распечатывает координаты точки. Подпрограмма производит пересчет по формулам:

$$\rho = \sqrt{x^2 + y^2} \quad \text{tg } \phi = y / x$$

2. Три заряда находятся в точках с координатами $(x_1, y_1), (x_2, y_2), (x_3, y_3)$. Рассчитать потенциал электрического поля в точке (x, y) .

Критерии оценивания индивидуальных задач

Код контролируемой компетенции (или её части)	Основные показатели оценки результата	Критерии оценки результата	Оценка
ОПК-4	Результативность и скорость работы с антивирусом, проведения проверки созданного документа, сохраненного на флеш-носителе, на наличие компьютерных вирусов.	Знание теории. Правильность, полнота и логичность	зачтено / не зачтено
ОПК-5	Степень соответствия разработанного алгоритма решения и текста программы заданию; Скорость и результативность отладки, анализа сообщений компилятора, внесения исправлений, запуска на выполнение программы на языке Паскаль; Степень соответствия полученных результатов заданию; Результативность работы в ИСР Free Pascal (работа со встроенным редактором, компилятором), использования сети Интернет для поиска информации при решении поставленной задачи.	построения решения задачи. Разработка программы. Правильность работы программы. Умение внести изменение в программу.	

Описание шкалы оценивания

Оценка «зачтено» выставляется обучающемуся, если он

1. знает теорию (т.е. назначение операторов и функций, использованных в программе);
2. разработал и отладил программу (программа должна работать, иметь хороший интерфейс, т.е. понятный для пользователя ввод-вывод);
3. имеет представление о программе (т.е. знает логику решения задачи, назначение отдельных блоков программы);
4. демонстрирует правильность работы программы на контрольных примерах;
5. умеет внести изменения в программу по требованию преподавателя. Оценка

«не зачтено» выставляется обучающемуся, если он

1. не знает теорию;
2. не разработал и не отладил программу;

РПД «Программирование»

3. не имеет представления о программе;
4. не демонстрирует правильность работы программы на контрольных примерах;
5. не умеет внести изменения в программу по требованию преподавателя.

6.2.2 Примеры заданий тестов

По теме «Программное обеспечение (ПО) компьютера.»

1. Система программирования предоставляет программисту возможность
 - : анализа существующих программных продуктов по соответствующей тематике
 - : выбора языка программирования
 - : автоматического построения математической модели, исходя из постановки задачи
 - +: автоматической сборки разработанных модулей в единый проект
2. Программа-компилятор...
 - : обрабатывает структуры данных программы -:
 - оптимизирует код программы -: строит задачу
 - +: переводит исходный текст в машинный код

По теме «Локальные и глобальные сети. Основы информационной безопасности»

1. Информационный запрос - это...
 - : входное сообщение с требованием выделения необходимых ресурсов
 - : необработанное обращение к операционной системе
 - : сигнал или сообщение оператору о необходимости ввода данных
 - +: запрос к информационной системе, содержащий условие на поиск и выборку данных
2. Сетевые вирусы могут попасть на локальный компьютер...
 - : при копировании файла с удаленного компьютера
 - +: при подключении к локальной сети
 - : при вводе логина и пароля
 - : при просмотре web-страницы

По теме «Программирование линейных алгоритмов»

1. Выберите из списка оператор языка Паскаль, который описывает целые переменные b и s
 - : var b,s:char;
 - +: var b, s: integer;
 - : var b, s: real;
 - : read (b,s);
 - : write(b=, s=);
 - : write(b,s);
2. Выбрать оператор, который выводит на экран символы "b=" (без кавычек) и значение целой переменной b, используя 6 позиций
 - : read(b);
 - : write('b=':6);

```
-: read(' b=',b);
+: write('b=', b:6);
-: write('b=', 'b:6:2');
-: write('b=', b:6:2, ' ');
```

По теме «Программирование ветвящихся алгоритмов»

1. Условие $|a|>2$ в языке Паскаль можно записать в виде логического выражения
 - : (a<2) or (a>-2)
 - : (a<2) and (a>-2)
 - +: (a>2) or (a<-2)
 - : (a>2) and (a<-2)
2. Фрагмент программы
if 1<n then begin m:=sqrt(n); write(m); end else m:=2*n; + при n=1 вычислит m=2
 - при n=1 вычислит m=1 и выведет значение m
 - при n=1 вычислит m=2 и выведет значение m
 - выполняться не будет, так как содержит ошибку

По теме «Программирование циклических алгоритмов»

1. Отметьте оператор, который выводит на экран в столбик целые числа от 1 до 10
 - : for i:=1 to 10 do i;
 - : for i:=2 to 11 do write(i-1);
 - +: for i:=2 to 11 do writeln(i-1);
 - : for i:=1 to 10 do read(i);
2. Фрагмент программы: for j:=1 to 10 do begin i:=2*j-1; writeln(j, ' ',i); end;
 - +: Выводит на экран два столбика чисел: в первом целые числа от 1 до 10, во втором нечетные от 1 до 19
 - : Выводит на экран два столбика чисел: в первом целые числа от 1 до 10, во втором нечетные от 3 до 21
 - : Выводит на экран два числа 1 и 1
 - : Выводит на экран два числа 10 и 19

По теме «Компьютерная графика»

1. В программе
x:=100;
for i:=0 to 100 do begin Setcolor(0);
 circle(x,430,20); x:=x+10;
 Setcolor(11); circle(x,430,20);
 { delay(100); } end;
отмеченный фигурными скобками фрагмент служит для
 - стирания объекта
 - рисования объекта
 - повторения единичного перемещения объекта
 - +задержки выполнения программы для рассматривания объекта

2. В операторе putpixel(a,b,c,d);

-: a -координата x точки, b - координата y точки ,c -цвет точки, d-размер

+: неверное число параметров в скобках

-: a -координата x центра, b - координата y центра ,c -радиус, d-цвет

-: a -начальная координата x, b - начальная координата y, c- конечная координата x, d-

конечная координата y

По теме «Массивы»

1. Оператор, который печатает в строку значения номеров и элементов массива nom из n элементов

-:write(nom);

+:for I:=1 to n do write(I,nom[I]);

-:for I:=0 to n do writeln(n,nom[n]); -

:writeln(n,nom[I]);

2. Фрагмент программы for

I:=1 to n do begin

write('nom['I,']=');

writeln(nom[I]); end;

-: Вводит с клавиатуры элементы массива с подсказкой

+: Печатает на экране значения номеров и элементов массива

-: Содержит ошибки

-: Вводит элементы массива с клавиатуры и выводит на экран число элементов

По теме «Процедуры и функции»

1. Программа:

Function Prim(a,b,res:integer)integer; begin

Prim:=a-b; end;

var c,d:integer;

begin

c:=Prim(2,3,d);

write(c); end;

-: выведет на экран число 5

+: выведет на экран число -1

-: работать не будет, так как неправильно описана подпрограмма

-: работать не будет, так как неправильно вызвана подпрограмма

-: работать не будет, так как при вызове подпрограммы неправильно указано число параметров

2. Оператор Prim(a,b,res);

-: описывает подпрограмму-функцию с именем Prim и параметрами-значениями a,b,res

-: описывает подпрограмму-процедуру с именем Prim и параметрами-значениями a,b,res

-: описывает подпрограмму-функцию с именем Prim и параметрами-

переменными a,b,res

-: описывает подпрограмму-процедуру с именем P_{prim} и параметрами-переменными a,b,res

-: написан неверно

-: вызывает подпрограмму-функцию с именем P_{prim} и параметрами a,b,res

+: вызывает подпрограмму-процедуру с именем P_{prim} и параметрами a,b,res

По теме «Парадигмы программирования»

1. При проектировании программного обеспечения используются подходы: а) "сверху-вниз"; б) "снизу-вверх"; в) "слева-направо"; г) "справа-налево" -: в, г -: б, в -: а, г +: а, б

2. Только объектную структуру программы имеют языки: а) C++

б) Fortran

в) Pascal г)

Java

Критерии оценивания результатов тестирования

Код контролируемой компетенции (или её части)	Основные показатели оценки результата	Критерии оценки результата	Оценка
ОПК-4	Выполнение всех запланированных компьютерных тестов.	Результативность тестирования.	зачтено /
ОПК-5			не зачтено

Описание шкалы оценивания

Оценка «зачтено» выставляется обучающемуся, если он правильно ответил на 60% и более заданий теста.

Оценка «не зачтено» выставляется обучающемуся, если он правильно выполнил не более 59% заданий теста.

6.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Для проверки сформированности элементов компетенций используется портфолио.

Тип портфолио: портфолио работ.

Состав портфолио:

1. Текст индивидуальной задачи на линейные алгоритмы с контрольным примером;
2. Текст индивидуальной задачи на ветвящиеся алгоритмы с контрольным примером;
3. Текст индивидуальной задачи на циклические алгоритмы с контрольным примером;
4. Текст индивидуальной задачи на рисование движущихся объектов;
5. Текст индивидуальной задачи на массивы с контрольным примером;

РПД «Программирование»

6. Текст индивидуальной задачи на подпрограммы с контрольным примером;
7. Результаты тестирования.

Критерии оценки:

Код контролируемой компетенции (или её части)	Показатели оценки результата
(ОПК-4) Способность понимать сущность и значение информации в развитии современного общества, сознавать опасности и угрозы, возникающие в этом процессе, соблюдать основные требования информационной безопасности	Результативность и скорость работы с антивирусом, проведения проверки созданного документа, сохраненного на флеш-носителе, на наличие компьютерных вирусов.
(ОПК-5) Способность использовать основные методы, способы и средства получения, хранения, переработки информации и навыки работы с компьютером как со средством управления информацией	Степень соответствия разработанного алгоритма решения и текста программы заданию; Скорость и результативность отладки, анализа сообщений компилятора, внесения исправлений, запуска на выполнение программы на языке Паскаль; Степень соответствия полученных результатов заданию; Результативность тестирования. Результативность работы в ИСП Free Pascal (работа со встроенным редактором, компилятором), использования сети Интернет для поиска информации при решении поставленной задачи.

**8. Перечень основной и дополнительной учебной литературы,
необходимой для освоения дисциплины**

а) основная учебная литература:

3. Алексеев Е.Р. Free Pascal и Lazarus: Учебник по программированию / Е.Р. Алексеев, О.В. Чеснокова, Т.В. Кучер. – М.: "ДМК Пресс", - 2010. - 438 с. (URL: http://e.lanbook.com/books/element.php?pl_1_id=1267, дата обращения 17.01.2015)
4. Могилёв А.В., Пак Н.И., Хеннер Е.К. Информатика : учеб. пособие для вузов. - М. : Академия , 2008 – 841 с.
5. Голицына, Ольга Леонидовна. Языки программирования [Текст] : учеб. пособие / О. Л. Голицына, Т. Л. Партыка, И. И. Попов. - М. : Форум : ИНФРА-М, 2008. - 397 с. : рис., табл. - (Профессиональное образование).

б) дополнительная учебная литература:

4. Зеленьяк О.П. Практикум программирования на Turbo Pascal. Задачи, алгоритмы и решения / О.П. Зеленьяк, 3-е изд., исправленное и дополненное. – М.: "ДМК Пресс", - 2009. - 320 с. (URL: http://e.lanbook.com/books/element.php?pl_1_id=1249, дата обращения 17.01.2015)
5. Безручко, Валерия Тимофеевна. Компьютерный практикум по курсу "Информатика" [Текст] : учеб. пособие / В. Т. Безручко. - 3-е изд., перераб.

- и доп. - М. : ФОРУМ - ИНФРА-М, 2009. - 367 с.
6. Безручко В.Т. Информатика : (курс лекций): учеб. пособие / В. Т. Безручко.- М. : ФОРУМ - ИНФРА-М , 2009.
 7. Программирование. Базовый курс : учебный курс.- М. , 2005 - Университетская серия.
 8. Основы программирования : интерактивный учебный курс.- М. : Равновесие , 2006 .

8. Перечень ~~ресурсов~~ информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»), необходимых для освоения дисциплины

5. Сайт физического факультета КемГУ, страница Павловой Т.Ю. (URL: http://physic.kemsu.ru/viewpage.php?page_id=108, дата обращения 17.01.2015)
6. URL: <http://www.sibstrin.ru/student/books/programming/>, дата обращения 17.01.2015
7. URL: <http://www.cyberguru.ru/programming/pascal/>, дата обращения 17.01.2015
8. URL: <http://tp7.info/index.php>, дата обращения 17.01.2015

9. Методические указания для обучающихся по освоению дисциплины

В курсе «Программирование» студент-физик должен приобрести теоретические и практические навыки обработки информации с помощью компьютера, написания относительно простых программ для использования в профессиональной деятельности.

На лекциях сначала излагается обзорный материал, посвященный операционным системам (в основном Windows), программному обеспечению компьютера. Затем излагаются основы языка Паскаль в следующем порядке:

- 9) Данные, типы данных
- 10) Простые типы данных
- 11) Виды данных
- 12) Основные операторы языка Паскаль
- 13) Массивы и работа с ними
- 14) Подпрограммы. Виды подпрограмм языка Паскаль
- 15) Другие типы данных
- 16) ООП

Порядок изложения материала на лекциях соответствует логике построения языка программирования, выявлению внутренних взаимосвязей данных и команд языка, которые их обрабатывают. Однако, такое изложение не способствует быстрому получению практических навыков и не соответствует порядку изучения языка на практических занятиях. Поэтому сначала материал лекций «отстает» от нужд практических занятий. Для ускорения темпа подачи лекционного материала часть его, упорядоченная в виде таблиц и примеров, рассматривается бегло и не записывается студентами на лекции. Этот материал размещен в кратких конспектах,

РПД «Программирование»

выставленных на сайте факультета.

Из-за недостатка времени, на лекции рассматривается достаточно небольшое число примеров использования данных, операторов, реализации алгоритмов на языке Паскаль. Большое число дополнительных примеров можно найти в учебных пособиях «Программирование на языке Паскаль для начинающих», «Структурное программирование В ИСР «Free Pascal» и в электронном учебном пособии «Прикладное программирование на языке Паскаль», размещенных на сайте факультета.

На лабораторных занятиях студент решает индивидуальные задачи по теме занятия (разрабатывает и отлаживает программы). В процессе изучения тем курса, каждый студент получает индивидуальные задачи физического или математического содержания, которые должен решить под руководством преподавателя. На аудиторных лабораторных занятиях преподаватель дает общие указания по решению задач, консультирует студентов. Далее проводится защита решенной задачи.

Для оптимальной разработки полноценной программы, необходимо последовательно пройти несколько этапов. Нарушение порядка выполнения работ, пропуск этапов, приводит обычно к появлению в программе трудно устранимых логических ошибок, потерям времени, необходимости возвращаться к уже проделанным этапам работы. Поэтому абсолютно необходимо вести разработку программы следующим образом:

1 этап – разработка алгоритма

На этом этапе необходимо определить имена и типы данных, которые будут участвовать в программе, порядок действий программы (алгоритм), разработать контрольный пример для программы, который поможет подтвердить правильность ее результатов.

2 этап – разработка программы

Теория программирования предлагает несколько подходов к разработке программ. Небольшие программы лучше писать, руководствуясь принципами структурного программирования.

Структурное программирование - технология разработки программ, основанная на принципах:

4. использование трёх базовых алгоритмических структур: следование, ветвление, цикл;
5. программирование "сверху-вниз";
6. использование модулей.

Способ программирования «сверху-вниз» заключается в том, что задача решается от общего к частному, крупные задачи разбиваются на более мелкие и т.д. пока очередные задачи смогут быть реализованы одним или несколькими простыми операторами языка программирования. При этом, если какие-то фрагменты программы повторяются или представляют собой отдельные, логически цельные задачи, они оформляются в виде отдельных модулей (подпрограмм).

3 этап – набор программы в окне редактора

Разработанную программу необходимо набрать с помощью клавиатуры, а затем записать на диск с помощью меню file/save as в виде имя.pas (вместо слова «имя» нужно набрать имя программы латинскими буквами). Если это не сделано, при запуске программы, ИСР потребует записать ее на диск (откроет соответствующее окно). Но лучше сохранять текст программы периодически, в процессе набора, нажимая клавишу F2.

4 этап – компиляция программы

Компилятор - программа, которая осуществляет перевод исходной программы в объектную программу на языке машинных команд или ассемблера.

Результатом работы компилятора будет являться программа (так называемый «объектный код», имеющий обычно расширение .OBJ) или сообщение об ошибках в исходной программе. Объектный код не является готовой для исполнения программой, требуется еще работа компоновщика.

Также компилятор исправляет некоторые ошибки или делает сообщения об ошибках, которые выводятся в окно сообщений компилятора. По умолчанию это окно появляется под окном редактора вместе с сообщением о наличии ошибок в программе. Нажав два раза клавишу ввода (Enter), в окне редактора можно увидеть сообщение о первой ошибке, и система выставит курсор в то место в программе, где эта ошибка возникает. Если окна компилятора не видно, можно нажать клавишу F6. Если окно компилятора и

в этом случае не появится, вызовите список доступных окон командой Alt+0, выделите Compiler Messages, нажмите кнопки Show и ОК для того, чтобы сделать окно компилятора видимым.

5 этап – компоновка с библиотеками подпрограмм и запуск программы на выполнение

4 и 5 этапы производятся по команде run (ctrl+F9), их можно проводить и отдельно.

6 этап – отладка программы

Синтаксические ошибки выявляются на этапе компиляции. Однако, в программе может содержаться некоторое количество логических ошибок, когда компьютер выдает неправильные результаты, из-за того, что в программе записан неверный алгоритм. Способы выявления логических ошибок описаны далее.

7 этап – анализ результатов, тестирование

На этом этапе необходимо протестировать результаты работы программы. Для этого используется один или несколько контрольных примеров. Если программа имеет ветвления, нужно подобрать контрольные примеры для каждой ветви. Если контрольные примеры выполняются неправильно, придется возвратиться на предыдущие этапы.

Теория и необходимые приемы решения основных задач изложены в пособии «Структурное программирование в ИСР Free Pascal» (в электронном виде URL: http://physic.kemsu.ru/pub/library/learn_pos/FreePascal.PDF, дата

РПД «Программирование»

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

В процессе освоения образовательной программы компетенции формируются по следующим этапам:

- 1) начальный этап дает общее представление о виде деятельности, основных закономерностях функционирования объектов профессиональной деятельности, методов и алгоритмов решения практических задач;

- 2) основной этап позволяет решать типовые задачи, принимать профессиональные и управленческие решения по известным алгоритмам, правилам и методикам;
- 3) завершающий этап предполагает готовность решать практические задачи повышенной сложности, нетиповые задачи, принимать профессиональные и управленческие решения в условиях неполной определенности, при недостаточном документальном, нормативном и методическом обеспечении.

При освоении дисциплины (модуля) компетенции, закрепленные за ней, реализуются по темам (разделам) дисциплины (модуля), в определенной степени (полностью или в оговоренной части) и на определенном этапе, что приведено в Таблице 1.

Таблица 1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

№	Код компетенции	Номер темы (раздела) дисциплины (модуля)	Степень реализации компетенции при освоении дисциплины (модуля)	Этап формирования компетенции при освоении дисциплины (модуля)
1.	ОПК-4	1-3	Способность понимать сущность и значение информации в развитии современного общества, сознавать опасности и угрозы, возникающие в этом процессе, соблюдать основные требования информационной безопасности	начальный
2.	ОПК-5	1-3	Способность использовать основные методы, способы и средства получения, хранения, переработки информации и навыки работы с компьютером как со средством управления информацией	Начальный

	Пороговый (удовл.) 61-75 баллов	Базовый (хор) 76-90 баллов	Повышенный (отл) 91-100 баллов	(лекции, семинарские, практические, лабораторные)	средства (тесты, творческие работы, проекты и др.)
ОПК-4	<p>Знает: информационные источники поиска, сбора, обработки, анализа и систематизации информации по теме исследования.</p> <p>Умеет: пользоваться информационными базами данных и электронными библиотеками при анализе задач в своей профессиональной области и в смежных областях.</p> <p>Владеет: Владеть способами обеспечения информационной безопасности при работе с информацией ограниченного распространения.</p>	<p>Знает: социальные, экономические и право-вые характеристики используемой в исследовании информации;</p> <p>Умеет: соблюдать требования информационной безопасности при работе с информацией ограниченного распространения, авторского права и др.</p> <p>Владеет: нормативной базой, регламентирующей требования по информационной безопасности, а также иными критериями доступа и открытости информации-</p>	<p>Знает: уровень достоверности эмпирической информации и возможность ее публичной легализации.</p> <p>Умеет: Полностью сформированное умение соблюдать требования информационной безопасности при работе с информацией ограниченного распространения, авторского права и др.</p> <p>Владеет: Успешное и систематическое использование нормативной базы, регламентирующей требования по информационной безопасности, а также иные критерии доступа и открытости информации.</p>	Лекции, семинарские занятия Самостоятельная работа студентов	Опросы; Контрольные работы; собеседования, вопросы для зачета
ОПК-5	<p>Знает: знать принципиальное устройство компьютера, понимать его блочную структуру, базовые принципы его работы, знать один язык программирования;</p> <p>Умеет: уметь провести коррекцию изображения с дефектами, нарисовать с помощью графического редактора схему эксперимента или исследовательской установки;</p> <p>Владеет: базовыми навыками работы с компьютером, как основным средством сохранения и переработки</p>	<p>Знает: современные методы, способы, средства получения, хранения и обработки информации, в том числе – правила и принципы работы в глобальных компьютерных сетях;</p> <p>Умеет: Инсталлировать сложное (и/или оригинальное) программное обеспечение для переработки информации, полученной в физических исследованиях при решении задач профессиональной деятельности;</p> <p>Владеет: навыками проведения расчетов с помощью специализированного математического пакета программ аналитических вычислений иметь</p>	<p>Знает: современные информационные технологии (в том числе – интернет-ресурсы и интернет-технологии), используемые в естественных науках</p> <p>Умеет: уметь соотносить возможности программного обеспечения с целями поставленной задачи профессиональной деятельности.</p> <p>Владеет: методами обработки массивов данных с использованием инструментария универсальных математических пакетов, освоить методики символьного и численного интегрирования и дифференцирования при помощи</p>	Лекции, семинарские занятия Самостоятельная работа студентов	Опросы; Контрольные работы; собеседования, вопросы для зачета

информации по дисциплинам общей и прикладной физики, владеть приемами поиска информации на сайтах научных журналов, владеть основами пользования офисными программными пакетами и программами обработки растровой и векторной графики;	опыт работы с современным профессиональным программным обеспечением, позволяющим выполнить модельный расчет свойств различных материалов, используемых в исследованиях при решении конкретных физических задач.	математических пакетов; □ навыками разработки программ на языке программирования, владеть на базовом уровне языком объектно-ориентированного программирования.		
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Шкала оценивания, показатели и критерии оценивания образовательных результатов обучающегося во время текущей аттестации

Шкала оценивания	Показатели и критерии оценивания
5, «отлично»	Оценка «отлично» ставится, если студент строит ответ логично в соответствии с планом, показывает максимально глубокие знания профессиональных терминов, понятий, категорий, концепций и теорий. Устанавливает содержательные межпредметные связи. Развернуто аргументирует выдвигаемые положения, приводит убедительные примеры.
4, «хорошо»	Оценка «хорошо» ставится, если студент строит свой ответ в соответствии с планом. В ответе представлены различные подходы к проблеме, но их обоснование недостаточно полно. Устанавливает содержательные межпредметные связи. Развернуто аргументирует выдвигаемые положения, приводит необходимые примеры, однако показывает некоторую непоследовательность анализа. Выводы правильны. Речь грамотна, используется профессиональная лексика.
3, «удовлетворительно»	Оценка «удовлетворительно» ставится, если ответ недостаточно логически выстроен, план ответа соблюдается непоследовательно. Студент обнаруживает слабость в развернутом раскрытии профессиональных понятий. Выдвигаемые положения декларируются, но недостаточно аргументированы. Ответ носит преимущественно теоретический характер, примеры ограничены, либо отсутствуют.
2, «неудовлетворительно»	Оценка «неудовлетворительно» ставится при условии недостаточного раскрытия профессиональных понятий, категорий, концепций, теорий. Студент проявляет стремление подменить научное обоснование проблем рассуждениями обыденно-повседневного бытового характера. Ответ содержит ряд серьезных неточностей. Выводы поверхностны

Шкала оценивания, показатели и критерии оценивания образовательных результатов обучающегося на зачете по дисциплине

Результат зачета	Показатели и критерии оценивания образовательных результатов

Результат зачета	Показатели и критерии оценивания образовательных результатов
зачтено	<p>Результат «зачтено» выставляется обучающемуся, если рейтинговая оценка (средний балл) его текущей аттестации по дисциплине входит в данный диапазон.</p> <p>При этом, обучающийся на учебных занятиях и по результатам самостоятельной работы демонстрировал знание материала, грамотно и по существу излагал его, не допускал существенных неточностей в ответе на вопрос, правильно применял использовал в ответах учебно-методический материал исходя из специфики практических вопросов и задач, владел необходимыми навыками и приёмами их выполнения.</p> <p>Учебные достижения в семестровый период и результаты рубежного контроля демонстрируют высокую (15...13) / хорошую (12..10) / достаточную (9...7) степень овладения программным материалом.</p> <p>Рейтинговые баллы назначаются обучающемуся как среднеарифметическое рейтинговых оценок по текущей аттестации (на занятиях и по результатам выполнения контрольных заданий) и промежуточной (экзамен) аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне от достаточного до высокого.</p>
не зачтено	<p>Результат «не зачтено» выставляется обучающемуся, если рейтинговая оценка (средний балл) его текущей аттестации по дисциплине входит в данный диапазон.</p> <p>При этом, обучающийся на учебных занятиях и по результатам самостоятельной работы демонстрирует незнание значительной части программного материала, допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы.</p> <p>Как правило, «не зачтено» ставится обучающимся, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.</p> <p>Учебные достижения в семестровый период и результаты рубежного контроля демонстрируют невысокую (недостаточную) степень овладения программным материалом.</p> <p>Рейтинговые баллы назначаются обучающемуся как среднеарифметическое рейтинговых оценок по текущей аттестации (на занятиях и по результатам выполнения контрольных заданий) и промежуточной (экзамен) аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, не сформированы</p>

3. Типовые контрольные задания или иные материалы, необходимые для оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Для оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций используются следующие типовые контрольные задания:

3.1. Текущий контроль успеваемости

Типовые тесты / задания

Примеры заданий тестов

По теме «Программное обеспечение (ПО) компьютера.»

1. Система программирования предоставляет программисту возможность -: анализа существующих программных продуктов по соответствующей

тематике

- : выбора языка программирования
- : автоматического построения математической модели, исходя из

постановки задачи

- +: автоматической сборки разработанных модулей в единый проект

3. Программа-компилятор...

- : обращает структуры данных программы -
- : оптимизирует код программы -: строит задачу +: переводит исходный текст в машинный код

По теме «Локальные и глобальные сети. Основы информационной безопасности»

1. Информационный запрос - это...

- : входное сообщение с требованием выделения необходимых ресурсов
- : необработанное обращение к операционной системе
- : сигнал или сообщение оператору о необходимости ввода данных
- +: запрос к информационной системе, содержащий условие на поиск и выборку данных

3. Сетевые вирусы могут попасть на локальный компьютер...

- : при копировании файла с удаленного компьютера
- +: при подключении к локальной сети
- : при вводе логина и пароля
- : при просмотре web-страницы

По теме «Программирование линейных алгоритмов»

3. Выберите из списка оператор языка Паскаль, который описывает целые переменные b и s

- : var b,s:char;
- +: var b, s: integer;
- : var b, s: real;
- : read (b,s);
- : write(b=, s=);
- : write(b,s);

4. Выбрать оператор, который выводит на экран символы "b=" (без кавычек) и значение целой переменной b, используя 6 позиций

- : read(b);
- : write('b=:6);

- : read(' b=',b);
- +: write('b=', b:6);
- : write('b=', 'b:6:2');
- : write('b=', b:6:2, '');

По теме «Программирование ветвящихся алгоритмов»

3. Условие $|a|>2$ в языке Паскаль можно записать в виде логического выражения

- : $(a < 2) \text{ or } (a > -2)$
- : $(a < 2) \text{ and } (a > -2)$
- +: $(a > 2) \text{ or } (a < -2)$
- : $(a > 2) \text{ and } (a < -2)$

4. Фрагмент программы

```
if 1 < n then begin m:=sqrt(n); write(m); end else m:=2*n;
```

+ при $n=1$ вычислит $m=2$

- при $n=1$ вычислит $m=1$ и выведет значение m
- при $n=1$ вычислит $m=2$ и выведет значение m
- выполняться не будет, так как содержит ошибку

По теме «Программирование циклических алгоритмов»

2. Отметьте оператор, который выводит на экран в столбик целые числа от 1 до 10

- : for i:=1 to 10 do i;
- : for i:=2 to 11 do write(i-1);
- +: for i:=2 to 11 do writeln(i-1);
- : for i:=1 to 10 do read(i);

2. Фрагмент программы: for j:=1 to 10 do begin i:=2*j-1; writeln(j, ' ', i); end;

- +: Выводит на экран два столбика чисел: в первом целые числа от 1 до 10, во втором нечетные от 1 до 19
- : Выводит на экран два столбика чисел: в первом целые числа от 1 до 10, во втором нечетные от 3 до 21
- : Выводит на экран два числа 1 и 1
- : Выводит на экран два числа 10 и 19

По теме «Компьютерная графика»

2. В программе

```
x:=100;
```

```
for i:=0 to 100 do begin  
  Setcolor(0);  
  circle(x,430,20); x:=x+10;  
  Setcolor(11); circle(x,430,20);  
  { delay(100); } end;
```

отмеченный фигурными скобками фрагмент служит для

-стирания объекта

-рисования объекта

-повторения единичного перемещения объекта

+задержки выполнения программы для рассматривания объекта

3. В операторе putpixel(a,b,c,d);

-: a -координата x точки, b - координата y точки ,c -цвет точки, d-размер

+: неверное число параметров в скобках

-: a -координата x центра, b - координата y центра ,c -радиус, d-цвет

-: a -начальная координата x, b - начальная координата y, c- конечная координата x, d-конечная координата y

По теме «Массивы»

3. Оператор, который печатает в строку значения номеров и элементов массива nom из n элементов

```
 -:write(nom);  
 +:for I:=1 to n do write(I,nom[I]);  
 -:for I:=0 to n do writeln(n,nom[n]);  
 -:writeln(n,nom[I]);
```

4. Фрагмент программы

```
for I:=1 to n do begin  
write('nom[' ,I, '=');  
writeln(nom[I]); end;
```

-: Вводит с клавиатуры элементы массива с подсказкой

+: Печатает на экране значения номеров и элементов массива

-: Содержит ошибки

-: Вводит элементы массива с клавиатуры и выводит на экран число элементов

По теме «Процедуры и функции»

3. Программа:

```
Function Prim(a,b,res:integer)integer;  
begin Prim:=a-b; end;  
var c,d:integer;  
begin  
c:=Prim(2,3,d);  
write(c); end;
```

-: выведет на экран число 5

+: выведет на экран число -1

-: работать не будет, так как неправильно описана подпрограмма

-: работать не будет, так как неправильно вызвана подпрограмма

-: работать не будет, так как при вызове подпрограммы неправильно указано число параметров

4. Оператор Prim(a,b,res);

-: описывает подпрограмму-функцию с именем Prim и параметрами-значениями a,b,res

-: описывает подпрограмму-процедуру с именем Prim и параметрами-значениями a,b,res

-: описывает подпрограмму-функцию с именем Prim и параметрами-переменными a,b,res

-: описывает подпрограмму-процедуру с именем Prim и параметрами-переменными a,b,res

-: написан неверно

-: вызывает подпрограмму-функцию с именем Prim и параметрами a,b,res

+: вызывает подпрограмму-процедуру с именем Prim и параметрами a,b,res

По теме «Парадигмы программирования»

3. При проектировании программного обеспечения используются подходы: а) "сверху-вниз"; б) "снизу-вверх"; в) "слева-направо"; г) "справа-налево" -: в, г -: б, в -: а, г +: а, б

4. Только объектную структуру программы имеют языки: а) C++

б)

Fortran

в) Pascal

г) Java

Типовые контрольные вопросы

1. Основные конструкции алгоритмического языка
2. Жизненный цикл программы
3. Особенности языка C++
4. Структура программы в языке C++
5. Понятие переменной
6. Объявление переменных.
7. Типы данных, определяемые пользователем.
8. Инициализация переменных.
9. Основные типы переменных
10. Синтаксис объявления имени переменной
11. Понятие константы. Объявление.
12. Понятие константы. Инициализация.
13. Простейшие арифметические операции
14. Операция %. Ее особенности.
15. Основные математические функции
16. Принципы структурного программирования
17. Оператор множественного ветвления switch
18. Условный оператор if-else
19. Вложенный оператор if.
20. Классификация циклов
21. Операции инкремента и декремента.
22. Оператор while()
23. Оператор For.
24. Оператор do...while()
25. Принципы модульного программирования
26. Описание функции
27. Вызов функции.
28. Прототипы функций.
29. Передача параметров по имени.
30. Передача параметров по адресу.
31. Данные типа «указатель».
32. Операции для работы с указателями
33. Объявление и инициализация одномерного массива.

34. Объявление и инициализация двумерного массива.
35. Доступ к элементам одномерного массива по индексу.
36. Доступ к элементам одномерного массива по адресу.
37. Доступ к элементам двумерного массива по индексу.
38. Доступ к элементам двумерного массива по адресу.
39. Принципы объектно-ориентированного программирования. Инкапсуляция.
40. Принципы объектно-ориентированного программирования. Полиморфизм.
41. Принципы объектно-ориентированного программирования. Наследование.
42. Структуры. Объявление.
43. Структуры. Инициализация.
44. Прямой доступ к элементам структуры.
45. Косвенный доступ к элементам структуры.
46. Массивы структур. Объявление.
47. Массивы структур. Инициализация.
48. Классы. Объявление открытых переменных.
49. Классы. Инициализация открытых переменных.
50. Классы. Объявление закрытых переменных.
51. Классы. Инициализация закрытых переменных.
52. Спецификаторы доступа.
53. Функции-члены класса.
54. Связывание функций.
55. Встроенные функции.
56. Конструкторы. Основные понятия.
57. Конструкторы по умолчанию.
58. Конструкторы с параметрами.
59. Стандартный класс `string`. Основные понятия.
60. Основные операции класса `string`.

4. Методические материалы, определяющие процедуры оценивания достижения запланированных результатов обучения по дисциплине (модулю)

Текущая аттестация

При оценивании устного опроса и участия в дискуссии на семинаре (практическом занятии) учитываются:

- степень раскрытия содержания материала;
- изложение материала (грамотность речи, точность использования терминологии и символики, логическая последовательность изложения материала);
- знание теории изученных вопросов, сформированность и устойчивость используемых при ответе умений и навыков.

Для оценивания результатов обучения в виде знаний используются такие процедуры и технологии как тестирование и опрос на семинарах (практических занятиях).

Для оценивания результатов обучения в виде умений и владений используются следующие процедуры и технологии:

- практические контрольные задания (далее – ПКЗ), включающих одну или несколько задач (вопросов) в виде краткой формулировки действий (комплекса действий), которые следует выполнить, или описание результата, который нужно получить.

По сложности ПКЗ разделяются на простые и комплексные задания.

Простые ПКЗ предполагают решение в одно или два действия. К ним можно отнести: простые ситуационные задачи с коротким ответом или простым действием; несложные задания по выполнению конкретных действий. Простые задания применяются для оценки умений. Комплексные задания требуют многоходовых решений как в типичной, так и в нестандартной ситуациях. Это задания в открытой форме, требующие поэтапного решения и развернутого ответа, в т.ч. задания на индивидуальное или коллективное выполнение проектов, на выполнение практических действий или лабораторных работ. Комплексные практические задания применяются для оценки владений.

Типы практических контрольных заданий:

- задания на установление правильной последовательности, взаимосвязанности действий, выяснения влияния различных факторов на результаты выполнения задания;
- установление последовательности (описать алгоритм выполнения действия),
- нахождение ошибок в последовательности (определить правильный вариант последовательности действий);
- указать возможное влияние факторов на последствия реализации умения и т.д.
- задания на принятие решения в нестандартной ситуации (ситуации выбора, многоальтернативности решений, проблемной ситуации).

Оценивание обучающегося на текущей аттестации осуществляется в соответствии с критериями, представленными в п. 2, и носит балльный характер.

Промежуточная аттестация

Форма промежуточной аттестации: Зачет.

При проведении промежуточной аттестации студент должен ответить на вопросы теоретического характера и практического характера.

При оценивании ответа на вопрос теоретического характера учитывается:

- теоретическое содержание не освоено, знание материала носит фрагментарный характер, наличие грубых ошибок в ответе;
- теоретическое содержание освоено частично, допущено не более двух-трех недочетов;
- теоретическое содержание освоено почти полностью, допущено не более одного-двух недочетов, но обучающийся смог бы их исправить самостоятельно;
- теоретическое содержание освоено полностью, ответ построен по собственному плану.

При оценивании ответа на вопрос практического характера учитывается объем правильного решения.

Основой для определения оценки служит уровень усвоения студентами материала, предусмотренного рабочей программой дисциплины.

