



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего профессионального образования

«Дальневосточный федеральный университет»  
(ДВФУ)

### ШКОЛА ЕСТЕСТВЕННЫХ НАУК

«СОГЛАСОВАНО»  
Руководитель ОП

Добжинский Ю.В.

«01» сентября 2017 г.



«УТВЕРЖДАЮ»

Заведующий кафедрой

«Информационные системы управления»

Документов

А.И. Сухомлинов

«01» сентября 2017 г.

### РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Введение в программирование

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Образовательная программа «Автоматизированные системы обработки информации и управления»

Форма подготовки очная

Курс 1 семестр 1,2

лекции 36 / 36 час.

практические занятия 18 / 36 час.

лабораторные работы 36 / 36 час.

в том числе с использованием МАО лек. \_\_\_ час. / пр 12 час. / лаб. 12 час.

всего часов аудиторной нагрузки 198 час.

в том числе с использованием МАО 24 час.

самостоятельная работа 126 час.

в том числе на подготовку к экзамену 36 / 37 час.

курсовая работа / курсовой проект \_\_\_ семестр

зачет \_\_\_ семестр

экзамен 1 / 2 семестр

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.01 «Информатика и вычислительная техника», утвержденного приказом Министерства образования и науки Российской Федерации от 04. 04. 2016 № 12-13-593

Рабочая программа обсуждена на заседании кафедры «Информационные системы управления», протокол № 5 от «22» января 2016 г.

Заведующий кафедрой ИСУ А.И. Сухомлинов

Составитель: ст. преп. кафедры ИСУ Е.А.Елсукова

**I. Рабочая программа пересмотрена на заседании кафедры:**

Протокол от « \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_  
(подпись) (И.О. Фамилия)

**II. Рабочая программа пересмотрена на заседании кафедры:**

Протокол от « \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Заведующий кафедрой \_\_\_\_\_  
(подпись) (И.О. Фамилия)

## ABSTRACT

**Bachelor's degree in 09.03.01 "Computer Science"**

**Study profile "Automated data processing and management"**

**Course title:** Introduction to Programming

**Variable part of Block 1, 9 credits**

**Instructor:** Elsukova Elena

**At the beginning of the course a student should be able to:** encode different types of information, to represent algorithm visualization, apply the basic data structures and the basic design of high-level language; algorithms to implement high-level programming language.

**Learning outcomes:** general professional competences (**GPC**) – **GPC2** ability to master the methods of using software to solve practical problems; **GPC4** ability to participate in setting up and adjustment of software and hardware systems.

**Course description:** The first section covers the basics of computing processes algorithmization; introduce the concept of high level programming language. In the second section detailed study of all the data used in Pascal. In part "Basic programming control structures" discusses features of the implementation structures "following", "alternative", "iteration". The fourth section is devoted to the basic algorithms: consider algorithms for sorting, searching, sorting organization; evaluate their complexity and efficiency. In the section "Dynamic data structures" is considered the organization of work with dynamic memory using linear lists, graphs. The last section is devoted to the implementation of structured programming on Pascal.

### **Main course literature:**

1. Pavlovskaya T.A. Pascal. Programmirovaniye na yazyke vysokogo urovnya. Uchebnik dlya vuzov [Pascal. Programming in a high level language. Textbook for high schools] – SPb:Peter, 2010. – 464p. (rus)
2. Nemnyugin S. Programmirovaniye na yazyke vysokogo urovnya. Uchebnik dlya vuzov. [Pascal. Programming in a high level language. Textbook for high schools. ] – SPb:Peter, 2008. – 544p. (rus)
3. Okulov S. M. Osnovy programmirovaniya [Programming algorithms] – M: Binom., 2014.— 386p. (rus)

**Form of final knowledge control:** Exam / Exam

## **Аннотация к рабочей программе дисциплины «Введение в программирование»**

Рабочая программа учебной дисциплины «Введение в программирование» предназначена для направления 09.03.01 Информатика и вычислительная техника, профиль Автоматизированные системы обработки информации и управления, и разработана в соответствии с ФГОС ВО по данному направлению.

Дисциплина «Введение в программирование» относится к вариативной части блока обязательных дисциплин (Б1.В.ОД.2.2) федерального государственного образовательного стандарта высшего образования.

Общая трудоемкость освоения дисциплины составляет 9 зачетных единиц, 324 часа. Учебным планом предусмотрены следующие виды учебной работы – аудиторные занятия (198 часов): лекции (72 часа), лабораторные работы (72 часа), практические занятия (54 часа) и самостоятельная работа (126 часов, в том числе 63 часа на подготовку к экзамену). Дисциплина реализуется в первом, втором семестрах I курса бакалавриата.

Для изучения дисциплины необходимы знания школьной программы по математике, физике и информатике. Студенты должны знать основы устройства ЭВМ, принципы кодирования и хранения информации, основы алгоритмизации.

**Цель дисциплины “Введение в программирование”** – получение студентами базовых теоретических знаний по алгоритмизации и программированию и приобретение практических навыков программирования на языке высокого уровня (ЯВУ).

### **Задачи:**

- изучить приемы алгоритмизации и программирования при решении задач на ЭВМ;
- дать представление информации в ЭВМ и различных структур данных;
- рассмотреть типовые задачи программирования и методы их решения, оценить сложность рассмотренных алгоритмов;
- освоить системы программирования Pascal и Delphi;
- получить навыки оформления программной документации.

Для успешного изучения дисциплины «Введение в программирование» у обучающихся должны быть сформированы следующие предварительные компетенции:

Должен знать:

- основы устройства ЭВМ;

- арифметические и логические основы работы компьютера;
- принципы кодирования и хранения информации на ЭВМ;
- основы алгоритмизации (понятие, свойства и способы описания алгоритмов, основные алгоритмические структуры).

Должен уметь:

- работать в среде исполнителя;
- читать алгоритмы, представленные в виде блок-схем;
- программно реализовывать типовые (несложные) алгоритмы;
- тестировать готовые программы.

Планируемые результаты обучения по данной дисциплине (знания, умения, владения), соотнесенные с планируемыми результатами освоения образовательной программы, характеризуют этапы формирования следующих компетенций (общекультурные/ общепрофессиональные/ профессиональные компетенции (элементы компетенций)):

Код и формулировка компетенции	Этапы формирования компетенции	
<b>ОПК-2</b> способностью осваивать методики использования программных средств для решения практических задач	Знает	основные понятия языка программирования высокого уровня; основные структуры данных Pascal и способы их реализации; основные конструкции Pascal.
	Умеет	ставить задачу и разрабатывать алгоритм ее решения.
	Владеет	навыками алгоритмизации и программирования при решении задач на ЭВМ.
<b>ОПК-4</b> способностью участвовать в настройке и наладке программно-аппаратных комплексов	Знает	основные этапы решения задач на ЭВМ; стандарты программной документации.
	Умеет	использовать технологию и средства структурного программирования.
	Владеет	методами отладки и решения задач на ЭВМ в различных режимах.

Для формирования вышеуказанных компетенций в рамках дисциплины «Введение в программирование» применяются следующие методы активного обучения:

- Практические занятия (12 час.) - Работа в малых группах.
- Лабораторные занятия (12 час.) - Коллективные решения творческих задач.

# **I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА**

## **Раздел I. Введение в дисциплину (10 час.)**

### **Тема 1. Введение в дисциплину (2 час.)**

Цели и задачи курса. Роль дисциплины в образовательной программе. Литература: обязательная, дополнительная, ресурсы Интернет. Место программирования в современном IT-обществе. Основные этапы решения задач на ЭВМ; Модель «Семь +».

### **Тема 2. Основы алгоритмизации вычислительных процессов (4 час.)**

Понятие алгоритма; его свойства. Способы представления алгоритмов. Преимущества и недостатки способов. Правила построения схем алгоритмов. Системные методы разработки алгоритмов. Метод структурной алгоритмизации. Алгоритмизация линейных, разветвляющихся и циклических вычислительных процессов. Понятие сложности алгоритма. Алгоритмически неразрешимые задачи.

### **Тема 3. Языки программирования: эволюция, основные понятия (2 час.)**

Эволюция языков программирования (ЯП). Типы ЯП: языки программирования низкого уровня, языки программирования высокого уровня. Связь между языком высокого уровня и машинным языком. Системы программирования. Развитие систем программирования: визуальные среды быстрого проектирования. Структура языка программирования. Синтаксис и семантика.

### **Тема 4. Pascal - ЯП высокого уровня. Программа на языке высокого уровня (2 час.)**

Алфавит языка Pascal. Элементарные конструкции. Имена, правила выбора имени. Числа: целые, действительные. Структура программы. Опции транслятора. Комментарии: назначение и правила составления.

## **Раздел II. Данные языка программирования Pascal (26 час.)**

### **Тема 1. Основные положения концепции данных языка программирования высокого уровня Pascal (4 час.)**

Концепция данных: описание, перечень допустимых операций. Базовые типы данных (Integer, Real, Boolean, Char) - описание, диапазон значений, размещение в оперативной памяти, перечень допустимых операций; функции для работы с данным типом. Тип данных определяемый пользователем. Порядковый тип данных. Функции преобразования типов, идентичность и совместимость типов.

### **Тема 2. Структурированный тип данных (2 час.)**

Классификация данных по структурному признаку. Структурированные типы данных - диапазон, перечисляемый тип. Структурированный тип данных массивы: способы описания, виды массивов (одномерные, многомерные), способы формирования массивов.

### **Тема 3. Обработка массивов (2 час.)**

Типовые задачи обработки одномерных массивов. Типовые задачи обработки матриц: задача минимакса, вычисление следа матрицы, учет диагональных элементов.

### **Тема 4. Строки (2 час.)**

Способы описания строковых данных. Таблица ASCII, UNICODE. Подпрограммы обработки строк. Обработка длинных строк.

### **Тема 5. Записи (2 час.)**

Способы описания записи. Структура записи: иерархические записи, записи с вариантами, строки и массивы в качестве полей записи. Доступ к полям записи; оператор присоединения. Записи как операнды и параметры. Использование записей при обработке баз данных.

### **Тема 6. Множества (2 час.)**

Описание множеств. Формирование множеств с помощью конструкторов. Виды множеств: литеральные, множества со значениями перечислимого типа. Операции для работы с множествами.

### **Тема 7. Организация хранения данных во внешней памяти (4 час.)**

Понятие физического и логического файла. Файловый тип данных в ЯП Pascal. Внешние и внутренние имена файлов. Организация работы с файлами. Виды файлов: текстовые, типизированные, нетипизированные. Системные файлы INPUT, OUTPUT. Процедуры и функции работы с файлами. Прямой и последовательный доступ к данным.

### **Тема 8. Динамические структуры данных (6 час.)**

Динамическая память, адреса и указатели. Ссылочный тип данных: описание, стандартные процедуры и функции. Линейные списки: основные виды и способы реализации; линейный список как абстрактный тип данных. Сложные динамические структуры.

### **Тема 9. Графы (2 час.)**

Основные понятия теории графов. Способы задания графов. Типовые задачи теории графов. Алгоритмы, основанные на обходах графов.

## **Раздел III. Основные управляющие структуры программирования (14 час)**

### **Тема 1. Операторы Pascal (2 час.)**

Понятия: операция, операнд, функция, знак операции. Классификации операторов. Выражения, тип выражения. Оператор присваивания.

Смешанное присваивание. Представление математических формул на Pascal. Стандартные математические функции.

## **Тема 2. Представление основных управляющих структур программирования (6 час.)**

Структуры выбора: операторы if и case. Организация ветвлений в программах. Условный оператор. Оператор безусловного перехода. Оператор выбора case.

Циклические структуры - цикл-счетчик (for), циклы с условием (while, repeat). Методика создания циклов. Вложенные циклы. Циклы, управляемые событием. Циклы без итераций. Использование шаблонов при создании циклов: счетчики, сигнальные метки, Boolean-флаги.

## **Тема 3. Правила вывода для основных структур программирования (2 час.)**

Организация ввода/вывода. Процедуры Read, Write; ReadLn, WriteLn. Форматированный вывод целочисленных и вещественных значений. Организация ввода/вывода структурированных данных.

## **Тема 4. Процедуры и функции (4 час.)**

Виды подпрограмм на Pascal: процедуры и функции. Стандартные подпрограммы Pascal, подпрограммы определяемые пользователем. Способы оформления подпрограмм. Локальные и глобальные имена; область действия идентификаторов. Формальные и фактические параметры. Вызов подпрограмм. Организация обмена данными между главной программой и подпрограммами.

## **Раздел IV. Разработка прикладных программ (12 час.)**

### **Тема 1. Алгоритмы целочисленной арифметики (2 час.)**

Операции целочисленной арифметики (div, mod). Особенности алгоритмов целочисленной арифметики. Алгоритм поиска простых чисел (Решето Эратосфена).

### **Тема 2. Рекурсия (2 час.)**

Понятия рекурсии: рекурсивное определение, рекурсивный спуск, рекурсивный подъём, глубина рекурсии, рекурсивный алгоритм. Программирование рекурсивных алгоритмов.

### **Тема 3. Сортировка (2 час.)**

Понятие упорядоченных данных. Классификация методов сортировки. Оценка эффективности сортировки. Метод выбора, модификации. Метод простого обмена (пузырька), модификация - "шейкер-сортировка". Метод простых вставок; модификации - метод бинарных вставок, метод Шелла. Метод слияний; модификации. Метод Хоара.

### **Тема 4. Алгоритмы поиска (2 час.)**



Алгоритмы и программы поиска. Характеристики алгоритма. Алгоритмы линейного и бинарного поиска. Алгоритмы нахождения пути на карте: волновой алгоритм (Алгоритм Ли), маршрутный алгоритм.

#### **Тема 5. Численные методы (4 час.)**

Нелинейные уравнения: постановка задачи, методы решения. Решение дифференциальных уравнений. Решение систем линейных алгебраических уравнений методом Гаусса. Задача численного интегрирования: метод прямоугольников, метод трапеции, метод Симпсона; точность вычислений.

#### **Раздел V. Структурное программирование (10 час.)**

##### **Тема 1. Модули в Turbo Pascal (2 час.)**

Структура модуля. Доступ к объектам модуля. Компиляция и подключение модулей. Разработка модулей пользователя (Unit).

##### **Тема 2. Стандартные модули (библиотеки) Pascal (4 час.)**

Структура системы программирования Pascal. Модуль DOS: типы константы, переменные, подпрограммы. Модуль CRT - Основные процедуры и функции текстового режима.

##### **Тема 3. Библиотека работы с графикой (2 час.)**

Программное обеспечение графического режима работы монитора. Особенности работы в графическом режиме. Алгоритмы и основные операторы формирования графических изображений. Создание анимации.

##### **Тема 4. Методология и технология программирования (2 час.)**

Императивное программирование, Теорема структуры. Структурное программирование: основные характеристики. Жизненный цикл программы. Корректность программ; способы конструирования и верификации программ. Анализ программ.

## **II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА**

### **Практические занятия (54 час.)**

#### **Занятие 1. Способы описания алгоритмов. Исполнение алгоритмов (2 час.)**

1. Способы описания алгоритмов: словесный, структурограммы, псевдокод. Преимущества и недостатки способов.

2. Графический способ представления алгоритмов. Основные блоки. Общие правила проектирования визуальных алгоритмов.

3. Исполнение алгоритмов. Таблицы трассировки

#### **Занятие 2. Линейные алгоритмы (2 час.)**

1. Следование.
2. Правила записи арифметических выражений
3. Стандартные математические функции. Правила записи стандартных функций.

**Занятие 3. Решение физических задач (2 час.)**

1. Особенности записи формул.
2. Этапы решения физических задач на ПК.
3. Решение задач.

**Занятие 4. Логические выражения (2 час.)**

1. Логические операции NOT, AND, OR, XOR.
2. Формирование сложных условий в виде составных логических выражений.

**Занятия 5-6. Разветвляющиеся алгоритмы (4 час.)**

1. Алгоритмическая структура «ветвление»
2. Примеры полного и неполного ветвления
3. Формирование сложных условий в виде составных логических выражений.
4. Множественный выбор
5. Решение задач на использование условного оператора

**Занятия 7-8. Циклические алгоритмы (6 час.)**

1. Алгоритмическая структура «итерация»
2. Основные виды циклических алгоритмов: циклы с параметром, циклы с предусловием, циклы с постусловием.
3. Методика составления циклических алгоритмов
4. Особенности использования циклических алгоритмов.
5. Типовые задачи применения циклических алгоритмов: табулирование функций; рекуррентные формулы, суммирование /произведение числового ряда.

**Занятия 9-10. Алгоритмы целочисленной арифметики с использованием метода активного обучения работа в малых группах (4 час.)**

1. Операции целочисленной арифметики (div, mod).
2. Особенности алгоритмов целочисленной арифметики
3. Типовые задачи.

**Занятия 11-12. Алгоритмы обработки одномерных массивов с использованием метода активного обучения работа в малых группах (4 час.)**

1. Способы описания одномерных массивов

2. Способы формирования массивов: с помощью клавиатуры, генератора случайных чисел, формул.

3. Типовые задачи обработки одномерных массивов: поиск максимального (минимального) элемента, поиск элемента по условию, инвертирование массива.

**Занятия 13-14. Алгоритмы обработки двумерных массивов (матриц) с использованием метода активного обучения работа в малых группах (4 час.)**

1. Особенности описания матриц

2. Использование вложенных циклов в алгоритмах работы с матрицами.

3. Типовые задачи обработки матриц: задача минимакса, след матрицы, учет диагональных элементов.

**Занятия 15-16. Алгоритмы сортировки (4 час.)**

1. Классификация методов сортировки: Методы выбора, Методы обмена, методы вставок; методы слияний.

2. Особенности применения алгоритмов сортировки

**Занятия 17 - 18. Алгоритмы поиска (4 час.)**

3. Алгоритмы бинарного поиска.

4. Алгоритм Евклида

5. Алгоритмы нахождения пути на карте: волновой алгоритм (Алгоритм Ли), маршрутный алгоритм.

**Занятия 19-20. Динамические структуры (4 час.)**

1. Организация линейных списков в прямом направлении.

2. Организация линейных списков в обратном порядке.

3. Алгоритмы работы с очередями различных видов.

4. Алгоритмы работы со стеками.

5. Построение сложных структур (Списки с полутора связями, списки с двумя связями) в динамической памяти

6. Примеры применения сложных структур - задачи сложения и умножения многочленов, представления разреженных матриц.

**Занятия 21-22. Алгоритмы с использованием подпрограмм (4 час.)**

1. Организация процедур и функций.

2. Выделение вспомогательных алгоритмов в сложных задачах.

3. Реализация рекурсии

**Занятия 23-24. Работа с библиотекой CRT (4 час.)**

1. Особенности работы с библиотекой CRT

2. Использование процедур работы с окнами

3. Использование процедур перемещения курсора и управления цветом.

**Занятия 25-26. Алгоритмы создания анимации (4 час.)**

1. Содержание библиотеки Graph.
2. Реализация графического режима работы.
3. Создание графических объектов
4. Работа с текстом.
5. Создание графиков.
6. Основной анимационный алгоритм.
7. Использование страниц при реализации анимации.

**Лабораторные работы (72 час.)**

**Лабораторная работа 1. Описание функции по её графическому представлению (6 час.)**

**Лабораторная работа 2. Табуляция функций (4 час.)**

**Лабораторная работа 3. Вычисление суммы ряда с заданной точностью (4 час.)**

**Лабораторная работа 4. Работа с массивами (4 час.)**

**Лабораторная работа 5. Основные действия на матрицами (6 час.)**

**Лабораторная работа 6. Реализация метода сортировки и оценка его эффективности с использованием метода активного обучения коллективные решения творческих задач (6 час.)**

**Лабораторная работа 7. Геометрия на плоскости с использованием метода активного обучения коллективные решения творческих задач (6 час.)**

**Лабораторная работа 8. Работа с перечисляемыми типами (6 час.)**

**Лабораторная работа 9. Использование множеств при обработке символьных и числовых массивов (4 час.)**

**Лабораторная работа 10. Использование записей при обработке статистических данных (6 час.)**

**Лабораторная работа 11. Работа с различными типами файлов (6 час.)**

**Лабораторная работа 12. Программирование численных методов (8 час.)**

**Лабораторная работа 13. Работа с динамическими структурами (6 час.)**

### **III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ**

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Введение в программирование» представлено в специальном приложении 1 «Учебно-методическое обеспечение самостоятельной работы обучающихся» и включает в себя:

- план-график выполнения самостоятельной работы по дисциплине;
- задания для самостоятельной работы обучающихся и методические рекомендации по их выполнению;
- критерии оценки выполнения самостоятельной работы.

### **IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА**

№ п/п	Контролируемые разделы дисциплины	Коды и этапы формирования компетенций		Оценочные средства - наименование	
				текущий контроль	промежуточная аттестация
1.	Введение в дисциплину	ОПК-2	Знает основные понятия языка программирования высокого уровня;	Собеседование (УО-1)	Вопросы 2÷3,5 (I);
		ОПК-4	Знает основные этапы решения задач на ЭВМ;	Собеседование (УО-1)	Вопросы 1(I) , 15 (II);
2	Данные языка программирования Pascal	ОПК-2	Знает основные структуры данных Pascal и способы их	Собеседование (УО-1)	Вопросы 8, 14 (I); 1-10 (II);

№ п/п	Контролируемые разделы дисциплины	Коды и этапы формирования компетенций		Оценочные средства - наименование	
				текущий контроль	промежуточная аттестация
			реализации;		
			Умеет организовать обработку различных типов и структур данных	Лабораторные работы (ПР-6) 1-5, 11, 12 (ПР-6);	
		ОПК-4	Знает способы представления различных видов информации на ЭВМ.	Собеседование (УО-1)	Вопросы 7,14 -15 (I), 1-10 (II) Вопросы 2÷3,
			Умеет реализовать обработку различных видов информации на ЭВМ.	Практические занятия (ПР-6) 5-8, 21, 23	
3.	Основные управляющие структуры ЯП	ОПК-2	Знает основные конструкции Pascal.	Собеседование (УО-1)	Вопросы 5÷13 (I), Вопросы 1÷4,
			Умеет ставить задачу и разрабатывать алгоритм ее решения.	Практические занятия (ПР-6) 2, 4, 9-13	
		ОПК-4	Знает базовые алгоритмы	Собеседование (УО-1)	Вопросы 15, 19 (I),
			Умеет оценить сложность и эффективность алгоритмов	Лабораторные работы (ПР-6) 7-9	
4.	Разработка прикладных программ	ОПК-2	Умеет ставить задачу и разрабатывать алгоритм ее решения.	Практические занятия (ПР-6) 1, 3	Вопросы 1÷4,20 (I); 11-15 (II)
			Владеет навыками алгоритмизации и программирования при решении задач на ЭВМ.	Лабораторные работы (ПР-6) 6, 9,13	
		ОПК-4	Знает стандарты программной документации.	Собеседование (УО-1)	

№ п/п	Контролируемые разделы дисциплины	Коды и этапы формирования компетенций	Оценочные средства - наименование		
			текущий контроль	промежуточная аттестация	
			Владеет методами отладки и решения задач на ЭВМ в различных режимах.	Практические занятия (ПР-6) 30-33	
		ОПК-2	Владеет навыками работы в интегрированной среде разработки программ PascalABC.Net.	Лабораторные работы (ПР-6) 1-5	
5.	Структурное программирование	ОПК-4	Умеет использовать технологию и средства структурного программирования.	Практические занятия (ПР-6) 27-29	Вопросы 1÷4,20 (I); 11-15 (II)
			Владеет методами отладки и решения задач на ЭВМ в различных режимах.	Практические занятия (ПР-6) 34-36	

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 2.

## **V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

### **Основная литература**

*(электронные и печатные издания)*

1. Павловская Т. А. Паскаль. Программирование на языке высокого уровня. Учебник для вузов. 2-е изд. – СПб: Питер, 2010. – 464с.
2. Немнюгин С. Турбо Паскаль. Программирование на языке высокого уровня. Учебник для вузов. 2-е изд. – СПб: Питер, 2008. – 544с.

3. Павловская Т. А. Паскаль. Программирование на языке высокого уровня. Практикум. 2-е изд. – СПб: Питер, 2007. – 317с.

4. Немнюгин С. Турбо Паскаль: Практикум. 2-е изд. – СПб: Питер, 2007. – 272с.

**Дополнительная литература**  
(печатные и электронные издания)

1. Афанасьева Т.В. Основы визуальной алгоритмизации: Учеб. пособие для студентов спец. 5102, 5525, 5501/Под ред. Валеева С.Г. – Ульяновск, изд-во УлГТУ, 2002. – 70с.

2. Бентли Дж. Жемчужины программирования. 2-е изд. – СПб: Питер, 2002. – 272с.

3. Кнут Д. Искусство программирования, Т.1,2,3. – М.: Издательский дом “Вильямс”, 2010.

4. Коффман Э. Б. Turbo Pascal, 5-е издание. – М: Издательский дом “Вильямс”, 2003. – 896с.

5. Малыхина М.П. Программирование на языке высокого уровня. Turbo Pascal. Уч. Пособие. – СПб: БХВ-Петербург, 2006. – 544с.

6. Меженный О.А. Самоучитель TURBO PASCAL. – М: Диалектика, 2007. – 330с.

7. Методические указания и задания по курсу «Алгоритмические языки и программирование» для студентов специальности 220200. – Березкина Г.Л., Елсукова Е.А., Покотило А.Л., Церабаев А.В. – Владивосток, ДВГТУ, изд-во ДВГТУ, 2006. – 60с.

8. Павловская Т.А. Паскаль. Карманный справочник. – СПб: Питер, 2005. – 160с.

9. Попов В.Б Паскаль и Дельфи. Учебный курс. – СПб: Питер, 2005. – 576с.

10. Порублев И.Н., Ставровский А.Б. Алгоритмы и программы. Решение олимпиадных задач. — М.: Издательский дом “Вильямс”, 2007. – 480с.

11. Рапаков Г.Г., Ржеуцкая С.Ю. Turbo Pascal для студентов и школьников. – СПб.: БХВ-Петербург, 2012. – 352с.

12. Симонович С.В. Информатика. Базовый курс: Учебник для вузов. 3-е изд. Стандарт третьего поколения.– СПб: Питер, 2012. – 640с.

**Перечень ресурсов информационно-телекоммуникационной сети  
«Интернет»**



1. <http://pascal.guti.ru/> – Электронный учебник, примеры программ
2. <http://pascal.sources.ru/> - примеры программ
3. <http://www.pascal-c.com/> - примеры программ.

### **Электронно-библиотечная система IPRbooks**

1. Окулов С.М. Основы программирования [Электронный ресурс]: учебное пособие/ Окулов С.М.— Электрон. текстовые данные.— М.: БИНОМ. Лаборатория знаний, 2012.— 340с.— Режим доступа: <http://www.iprbookshop.ru/6449>.

2. Культин, Н. Б. Turbo Pascal в задачах и примерах. - СПб.: БХВ-Петербург, 2012. – 256с. .— Режим доступа: <http://www.iprbookshop.ru/>

3. Культин, Н. Б. Программирование в Turbo Pascal 7.0 и Delphi. Самоучитель. 3-е изд. .- СПб.: БХВ-Петербург.- 2012. – 390с. .— Режим доступа: <http://www.iprbookshop.ru>

### **Перечень информационных технологий и программного обеспечения**

На основе учебно-методического комплекса дисциплины «Введение в программирование» разработан электронный учебный курс в интегрированной платформе электронного обучения Blackboard ДВФУ.

Идентификатор курса в Blackboard – FU50706-09.03.01-Pr-01.

## **VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ**

### **Правила изучения учебной дисциплины**

1. Усвоить язык учебной дисциплины (ключевые слова).

*Составить словарь, каждое новое понятие многократно проговорить.*

2. Понять структуру, логику предмета, использовать системный подход.

*К каждой теме составить таблицы, матрицы, алгоритмы, опорные схемы и конспекты.*

3. Усваивать не только информацию, но и метод её изучения как основу способов профессиональной деятельности.

*Организовывать для себя публичное выступление, участвовать в коллективной работе на занятиях, усваивая правила и технологию общения.*

4. Оценивать собственный результат после каждого самостоятельного и аудиторного занятия.

*Оценивать усвоенную информацию и метод, которым овладели, а также степень интереса.*

5. Относиться к учебному труду эмоционально. Положительные эмоции повышают эффективность вашей учебной деятельности.

*В учебной деятельности обучающийся приобретает ценности, которые должны его радовать как всякое приобретение.*

*Подготовлено к.п.н Белоус Е. И.,  
доцент кафедры психологии Школы гуманитарных наук ДВФУ*

### **Методические указания к выполнению лабораторных работ**

Целью выполнения лабораторных работ является закрепление теоретических знаний и приобретение практических навыков программирования на языке высокого уровня.

Перед выполнением лабораторной работы студент должен ознакомиться с заданием и порядком выполнения работы на странице курса в LMS Blackboard. Также студент должен изучить соответствующий теоретический раздел.

По каждой лабораторной работе необходимо составить и защитить отчёт. Отчет оформляется по правилам, принятым в ДВФУ. Отчёт по работе должен содержать все элементы, перечисленные в руководстве к лабораторной работе. Результаты, полученные в ходе лабораторной работы, должны быть обобщены в выводах.

## **VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ**

Аудиторный фонд - аудитории со средствами вычислительной техники.

10 рабочих станций или виртуальных машин на рабочих станциях. Минимальные требования к персональным компьютерам: тактовая частота центрального процессора не ниже 2 ГГц, оперативная память объемом не менее 512 Мбайт, жесткие магнитные диски с интерфейсом SerialATA и емкостью не менее 300 Мбайт.

Персональный компьютер преподавателя с мультимедиа-проектором и экраном, программным обеспечением для демонстрации слайд-презентаций —1 комплект.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
**«Дальневосточный федеральный университет»**  
(ДВФУ)

---

---

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ  
РАБОТЫ ОБУЧАЮЩИХСЯ**

по дисциплине «Введение в программирование»

Направление подготовки 09.03.01 «Информатика и вычислительная  
техника»

профиль - «Автоматизированные системы обработки информации и  
управления»

**Форма подготовки - очная**

**Владивосток  
2017**

**План-график выполнения самостоятельной работы по дисциплине  
«Введение в программирование»**

**1 семестр**

<b>№ п/п</b>	<b>Дата/сроки выполнения</b>	<b>Вид самостоятельной работы</b>	<b>Примерные нормы времени на выполнение</b>	<b>Форма контроля</b>
1.	2-я неделя	Подготовка к занятию в активной форме	5	УО-1
2.	4-я неделя	Оформление отчета по лабораторной работе 1	4	ПР-6
3.	5-я неделя	Подготовка к занятию в активной форме	4	ПР-6
4.	6-я неделя	Оформление отчета по лабораторной работе 2	3	УО-1
5.	8-я неделя	Подготовка к занятию в активной форме	4	ПР-6
6.	9-я неделя	Оформление отчета по лабораторной работе 3	3	ПР-2
7.	10-я неделя	Подготовка к занятию в активной форме	4	ПР-6
8.	11-я неделя	Подготовка к контрольной работе 1	8	ПР-6
9.	12-я неделя	Оформление отчета по лабораторной работе 4	3	ПР-6
10.	13-я неделя	Подготовка к занятию в активной форме	4	УО-1
11.		Подготовка к экзамену	36	ПР-2
<b>Всего в семестре</b>			<b>90 часов</b>	

**2 семестр**

<b>№ п/п</b>	<b>Дата/сроки выполнения</b>	<b>Вид самостоятельной работы</b>	<b>Примерные нормы времени на выполнение</b>	<b>Форма контроля</b>
1.	3-я неделя	Оформление отчета по лабораторной работе	2	ПР-6

2.	10-я неделя	Подготовка к контрольной работе	5	Пр-2
3.	16-я неделя	Оформление отчета по лабораторной работе	2	ПР-6
4.		Подготовка к экзамену	27	ПР-2
<b>Всего в семестре</b>			<b>36 часов</b>	

### **1. Подготовка к занятиям в активной форме**

Рабочей учебной программой предусмотрено проведение занятий в активных формах:

- Практические занятия (12 час.) - Работа в малых группах.
- Лабораторные занятия (12 час.) - Коллективные решения творческих задач.

#### **Описание метода *Работа в малых группах***

*Работа в малых группах* – форма активного обучения, которая дает всем обучающимся возможность участвовать в работе, практиковать навыки сотрудничества, межличностного общения. Метод наиболее подходит для проведения практических занятий по данному курсу: используются не только собственные знания, но и знания одногруппников, меняются социальные роли в зависимости от ситуации, что способствует более глубокому усвоению материала и совершенствует навыки общения и взаимодействия в группе.

#### **Методика осуществления**

**Организационный этап.** Подбор практического задания, отвечающего следующим критериям: не имеет однозначного и односложного решения; является практическим и полезным для учащихся; вызывает интерес у учащихся; максимально служит целям обучения.

Группа студентов делится на несколько малых групп. Количество групп определяется числом творческих заданий, которые будут обсуждаться в процессе занятия. Малые группы формируются либо по желанию студентов, либо по родственной тематике для обсуждения. В группе определяются: капитан (занимает лидирующую позицию, организует обсуждение на уровне группы, формулирует общее мнение малой группы), оппонент (внимательно слушает предлагаемые позиции во время дискуссии и формулирует вопросы по предлагаемой информации), эксперт (формирует оценочное суждение по предлагаемому решению своей малой группы и сравнивает с предлагаемыми решениями других групп).

**Подготовительный этап.** Каждая малая группа обсуждает творческое задание в течение отведенного времени. Задача данного этапа – сформулировать групповую позицию по заданию.

**Основной этап – проведение обсуждения задания.** Заслушиваются суждения, предлагаемые каждой малой группой по заданию. После каждого суждения оппоненты задают вопросы, выслушиваются ответы авторов предлагаемых позиций. В завершении формулируется общее мнение, выражающее совместную позицию по заданию.

**Этап рефлексии – подведения итогов.** Эксперты предлагают оценочные суждения по высказанным путям решения предлагаемых заданий осуществляют сравнительный анализ предложенного пути решения с решениями других малых групп. Преподаватель дает оценочное суждение и работе малых групп, по решению заданий, и эффективности предложенных путей решения.

### **Ход занятия**

**Организация** Занятие разбивается на две ступени.

**I ступень:** Студенты разбиваются на группы по 3-4 человека. В каждой группе назначается Капитан. Получают задание. Постановка задачи и определение регламента (не более 20 минут)

**Основной этап**

1. Проверка выполнения задания.
2. Рефлексия: анализ ошибок; изучение распределения ролей в группе. Выявление самого активного участника.

**II ступень:** Группы остаются без самого активного члена команды, который становится экспертом. Выдаются новые задания. Повторение задачи и определение регламента (не более 20 минут)

**Основной этап.**

1. Проверка выполнения задания
2. Рефлексия: эксперты анализируют ошибки (в случае их наличия). изучение распределения ролей в группе. Выявление самого активного участника занятия.

**Итоги:** Определение тем, которые необходимо повторить или изучить. Выставление оценок.

### **Практические занятия с использованием данного метода**

1. Алгоритмы целочисленной арифметики (4 час.).
2. Алгоритмы обработки одномерных массивов (4 час.)
3. Алгоритмы обработки двумерных массивов (матриц) (4 час.)

## **Описание метода Коллективные решения творческих задач**

Коллективные решения творческих задач – форма активного обучения, предназначенная для проведения практических занятий, которая дает возможность обучающимся найти собственное «правильное» решение, основанное на персональном опыте и опыте своих одноклассников.

Метод подходит для лабораторных занятий по данному курсу, т.к. наиболее полное освоение материала предполагает решение творческих задач. Под творческими заданиями здесь понимаются нетиповые задания, которые требуют от обучающихся не просто воспроизвести изучаемый алгоритм, а проявить творческий подход.

### **Методика осуществления**

**Организационный этап.** Подбор творческого задания, отвечающего нескольким критериям: не имеет однозначного и односложного ответа или решения; является практическим и полезным для учащихся; связано с будущей профессиональной деятельностью; вызывает интерес у учащихся; максимально служит целям обучения.

Группа студентов делится на несколько малых групп. Количество групп определяется числом творческих заданий, которые будут обсуждаться в процессе занятия. Малые группы формируются либо по желанию студентов, либо по родственной тематике для обсуждения. В группе определяются: капитан (занимает лидирующую позицию, организует обсуждение на уровне группы, формулирует общее мнение малой группы), оппонент (внимательно слушает предлагаемые позиции во время дискуссии и формулирует вопросы по предлагаемой информации), эксперт (формирует оценочное суждение по предлагаемому решению своей малой группы и сравнивает с предлагаемыми решениями других групп).

**Подготовительный этап.** Каждая малая группа обсуждает задание в течение отведенного времени. Назначение данного этапа – выбрать из нескольких вариантов наиболее оптимальное решение задания.

**Основной этап – проведение обсуждения задания.**

Заслушиваются методы (алгоритмы), предлагаемые каждой малой группой по заданию. После каждого выступления оппоненты задают вопросы, выслушиваются ответы авторов предлагаемых позиций. В завершении формулируется общее мнение, выражающее совместную позицию по заданию.

**Этап рефлексии – подведения итогов.** Эксперты предлагают оценочные суждения по высказанным методам (алгоритмам) решения предлагаемых заданий; осуществляют сравнительный анализ предложенного пути решения с решениями других малых групп.

Преподаватель дает оценочное суждение и работе малых групп, по решению заданий, и эффективности предложенных путей решения.

### **Лабораторные занятия с использованием данного метода**

1. Реализация метода сортировки и оценка его эффективности (6 час.)
2. Геометрия на плоскости (6 час.)

## **2. Подготовка к контрольной работе**

РУПД предусмотрено проведение двух контрольных работ на лекционных занятиях. Цель проведения контрольных работ - определить уровень усвоения студентами и оценить качество их теоретических знаний по данным темам.

Контрольная работа 1. Тема: *Логические и арифметические выражения; условный оператор*

Контрольная работа проводится после изучения тем: “Выражения Pascal”, “Оператор присваивания”, “Условный оператор”.

**Для успешного выполнения контрольной работы необходимо знать:**

- ✓ Назначение и вид оператора присваивания, условного оператора в полной и неполной формах;
- ✓ Основные логические операции;
- ✓ Правила записи арифметических и логических выражений.

Контрольная работа 2. Тема: *Функции обработки строк. Операции над множествами.*

Контрольная работа проводится после изучения тем: “Множества”, “Обработка текстовой информации на Pascal”.

**Для успешного выполнения контрольной работы необходимо знать:**

- ✓ способы описания множеств, виды множеств;
- ✓ формирование множеств с помощью конструкторов;
- ✓ операции для работы с множествами.
- ✓ Способы описания строковых данных.
- ✓ Подпрограммы (функции и процедуры) обработки строк.
- ✓ Обработка длинных строк .

### **Критерии оценки контрольной работы:**

- ✓ **5 баллов** выставляется студенту, если все задания выполнены верно, без ошибок.



- ✓ **4 балла** выставляется студенту, если все задания выполнены верно, есть 2-3 неточности.
- ✓ **3 балла** выставляется студенту, если 1 задание выполнено неверно, допущены 2-3 неточности при решении других задач.
- ✓ **2 балла** выставляется студенту, если две и более задачи решены неверно.

### **1. Методические рекомендации к оформлению отчетов по лабораторным работам**

Отчеты по лабораторным работам представляются в электронной форме, оформленные в MSWord по правилам, принятым в ДВФУ. Отчет должен содержать:

1. Титульный лист \*.
2. Содержание \*.
3. Задание на лабораторную работу \* (указать цель лабораторной работы, привести вариант, тему)
4. Краткое изложение теоретических положений, необходимых для выполнения работы.
5. Основная часть: материалы выполнения заданий.
6. Анализ полученных результатов.
7. Выводы по работе \* (какие задачи решены, что освоено при выполнении работы).
8. Список использованной литературы\*.
9. Приложения \* (при необходимости, обычно в Приложении выносят листинг программы)

\* *Включаются в отчет с новой страницы*

### **Инструкция для студента**

Билет письменного экзамена по курсу “*Введение в программирование*” состоит из 20 вопросов по всем изученным темам семестра. На экзамен отводится один академический час (45 мин.).

#### **Прежде, чем писать ответы:**

Установите себе такие временные рамки, чтобы успеть ответить на каждый вопрос, а затем просмотреть и отредактировать все вопросы.

Внимательно читайте вопросы.

Предлагаются вопросы разной степени сложности, установите приоритеты и в соответствии с ними отводите время на каждый вопрос.

Ответы на некоторые вопросы приходят на ум уже в процессе чтения. Запишите ключевые слова, наметки, пока они еще свежи в памяти. Иначе, можно потерять эти мысли к тому моменту, когда надо будет писать ответ на вопрос.

Обратите внимание на формулировку вопросов – выражения "Укажите недостатки", "Объяснить", "Укажите различия", "Сравнить" определяют, что должен содержать ответ.

**При записи ответов:**

Отведенное время для записи ответов распределите и организуйте. Когда время, отведенное на вопрос, закончилось, оставьте свободное место и приступайте к следующему вопросу. Неполные ответы можно дополнить в течение времени, отведенного на проверку (да и за шесть неполных ответов обычно можно получить больше баллов, чем за три полных).

**Ответы должны быть ПОЛНЫМИ, ЯСНЫМИ И КОМПАКТНЫМИ.**



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
**«Дальневосточный федеральный университет»**  
(ДВФУ)

---

---

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ**  
**по дисциплине «Введение в программирование»**  
**Направление подготовки 09.03.01 «Информатика и вычислительная**  
**техника»**  
профиль - «Автоматизированные системы обработки информации и  
управления»  
**Форма подготовки - очная**

**Владивосток**  
**2017**

## Паспорт ФОС

Код и формулировка компетенции	Этапы формирования компетенции	
<b>ОПК-2</b> способностью осваивать методики использования программных средств для решения практических задач	Знает	основные понятия языка программирования высокого уровня; основные структуры данных Pascal и способы их реализации; основные конструкции Pascal.
	Умеет	ставить задачу и разрабатывать алгоритм ее решения.
	Владеет	навыками алгоритмизации и программирования при решении задач на ЭВМ.
<b>ОПК-4</b> способностью участвовать в настройке и наладке программно- аппаратных комплексов	Знает	основные этапы решения задач на ЭВМ; стандарты программной документации.
	Умеет	использовать технологию и средства структурного программирования.
	Владеет	методами отладки и решения задач на ЭВМ в различных режимах.

№ п/п	Контролируемые разделы дисциплины	Коды и этапы формирования компетенций		Оценочные средства - наименование	
				текущий контроль	промежуточная аттестация
1.	Введение в дисциплину	ОПК-2	Знает основные понятия языка программирования высокого уровня;	Собеседование (УО-1)	Вопросы 2÷3,5 (I);
		ОПК-4	Знает основные этапы решения задач на ЭВМ;	Собеседование (УО-1)	Вопросы 1(I) , 15 (II);
2	Данные языка программирования Pascal	ОПК-2	Знает основные структуры данных Pascal и способы их реализации;	Собеседование (УО-1)	Вопросы 8, 14 (I); 1-10 (II);
			Умеет организовать обработку различных типов и структур данных	Лабораторные работы (ПР-6) 1-5, 11, 12 (ПР-6);	

№ п/п	Контролируемые разделы дисциплины	Коды и этапы формирования компетенций		Оценочные средства - наименование	
				текущий контроль	промежуточная аттестация
		ОПК-4	Знает способы представления различных видов информации на ЭВМ.	Собеседование (УО-1)	Вопросы 7,14 -15 (I), 1-10 (II) Вопросы 2÷3,
			Умеет реализовать обработку различных видов информации на ЭВМ.	Практические занятия (ПР-6) 5-8, 21, 23	
3.	Основные управляющие структуры ЯП	ОПК-2	Знает основные конструкции Pascal.	Собеседование (УО-1)	Вопросы 5÷13 (I), Вопросы 1÷4,
			Умеет ставить задачу и разрабатывать алгоритм ее решения.	Практические занятия (ПР-6) 2, 4, 9-13	
		ОПК-4	Знает базовые алгоритмы	Собеседование (УО-1)	Вопросы 15, 19 (I),
			Умеет оценить сложность и эффективность алгоритмов	Лабораторные работы (ПР-6) 7-9	
4.	Разработка прикладных программ	ОПК-2	Умеет ставить задачу и разрабатывать алгоритм ее решения.	Практические занятия (ПР-6) 1, 3	Вопросы 1÷4,20 (I); 11-15 (II)
			Владеет навыками алгоритмизации и программирования при решении задач на ЭВМ.	Лабораторные работы (ПР-6) 6, 9,13	
		ОПК-4	Знает стандарты программной документации.	Собеседование (УО-1)	
			Владеет методами отладки и решения задач на ЭВМ в различных режимах.	Практические занятия (ПР-6) 30-33	
		ОПК-2	Владеет навыками	Лабораторные	

№ п/п	Контролируемые разделы дисциплины	Коды и этапы формирования компетенций		Оценочные средства - наименование	
				текущий контроль	промежуточная аттестация
			работы в интегрированной среде разработки программ PascalABC.Net.	работы (ПР-6) 1-5	
5.	Структурное программирование	ОПК-4	Умеет использовать технологию и средства структурного программирования.	Практические занятия (ПР-6) 27-29	Вопросы 1÷4,20 (I); 11-15 (II)
			Владеет методами отладки и решения задач на ЭВМ в различных режимах.	Практические занятия (ПР-6) 34-36	

**Шкала оценивания уровня сформированности компетенций  
по дисциплине «Введение в программирование»**

Код и формулировка компетенции	Этапы формирования компетенции		Критерии	Показатели	Баллы
ОПК-2 способностью осваивать методики использования программных средств для решения практических задач	знает (пороговый уровень)	основные понятия языка программирования высокого уровня (ЯВУ); основные структуры данных Pascal и способы их реализации; основные конструкции Pascal.	Знание базовых типов данных; основных алгоритмических структур; способы реализации на Pascal	Способность дать определения основных понятий ЯВУ; Способность пояснить назначение и раскрыть содержание основных структур данных Pascal и способы их реализации; основных конструкций Pascal.	45÷64

Код и формулировка компетенции	Этапы формирования компетенции		Критерии	Показатели	Баллы
	умеет (продвинутый)	ставить задачу и разрабатывать алгоритм ее решения.	выделять основные этапы создания программ; «читать» схемы алгоритмов; составлять алгоритм.	Способность выделять основные этапы создания программ. Способность «читать» схемы алгоритмов. Способность составлять алгоритм.	65÷84
	владеет (высокий)	навыками алгоритмизации и программирования при решении задач на ЭВМ.	разрабатывать программы с использованием подпрограмм по заданному алгоритму; составлять, редактировать и выполнять отладку программы в интегрированных средах программирования	Способность разрабатывать программы с использованием подпрограмм по заданному алгоритму. Способность составлять, редактировать и выполнять отладку программы в интегрированных средах программирования	85÷100
ОПК-4 способностью участвовать в настройке и наладке программно-аппаратных комплексов	знает (пороговый уровень)	основные этапы решения задач на ЭВМ; стандарты программной документации.	Знание основных этапов решения задач на ЭВМ; Знание основных положений оформления программной документации.	Способность перечислить этапы решения задач на ЭВМ. Способность раскрыть назначение и содержание основных положений оформления программной документации.	45÷64
	умеет (продвинутый)	использовать технологию и средства структурного программирования.	Выделять в алгоритмах отдельные виды работ и оформлять основные программные	Способность выделять в алгоритмах отдельные виды работ. Способность оформлять	65÷84

Код и формулировка компетенции	Этапы формирования компетенции		Критерии	Показатели	Баллы
			единицы в виде подпрограмм и модулей.	основные программные единицы в виде подпрограмм и модулей.	
	владеет (высокий)	методами отладки и решения задач на ЭВМ в различных режимах.	Владение навыками отладки и решения задач на ЭВМ в различных режимах.	Способность провести отладку программы. Способность подготовить тесты и выполнить тестирование программы. Способность запускать программы в консольном и графическом режимах.	85÷100

**Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины «Введение в программирование»**

**Оценочные средства для промежуточной аттестации**

**Вопросы к экзамену (1 семестр)**

1. Основные этапы решения задач на ЭВМ.
2. Алгоритм: понятие, основные свойства, способы описания.
3. Виды алгоритмов: линейный, разветвляющийся, циклический, подпрограмма.
4. Методика составления алгоритма с циклом
5. Элементы языка программирования Pascal: алфавит, имена, константы, операторы, операции, выражения.
6. Понятие программы. Структура программы на Pascal.
7. Понятие идентификатора (имени). Правила выбора имени.
8. Описание базовых (стандартных) типов данных на Pascal.
9. Операторы языка Pascal и их классификация.
10. Оператор присваивания. Пустой оператор. Составной оператор.
11. Операторы ввода/вывода на Pascal.



- 12.Реализация условий на Pascal: условный оператор, оператор варианта.
- 13.Реализация циклов на Pascal: цикл с параметром, цикл с проверкой условия.
- 14.Организация хранения данных в памяти ЭВМ. Переменные. Массивы.
- 15.Описание и обработка массивов на Pascal.
- 16.Подпрограммы языка Pascal: формальные и фактические параметры, локальные и глобальные переменные.
- 17.Описание функций, обращение к функциям.
- 18.Описание процедур, обращение к процедурам.
- 19.Организация рекурсии.
- 20.Структурное программирование: назначение, реализация.

### **Вопросы к экзамену (2 семестр)**

1. Типы данных Pascal.
2. Перечисляемый тип данных, особенности работы.
3. Тип данных – диапазон, особенности работы.
4. Строковый тип данных: операции, стандартные функции и процедуры обработки.
5. Тип данных – множества: определение, операции над множествами.
6. Тип данных – записи: объявление, обращение к записи, допустимые операции над целой записью и элементами, запись с вариантами.
7. Файловый тип данных: типы файлов, работа с файлами.
8. Ссылочный тип данных (указатели): определение, допустимые операции.
9. Динамические структуры – списки, стеки.
- 10.Совместимость данных.
- 11.Модули: назначение, создание, структура модуля.
- 12.Системные модули (библиотеки) Pascal: назначение и содержание библиотек CRT, DOS, GRAPH.
- 13.Основные процедуры и функции текстового режима.
- 14.Особенности работы в графическом режиме.
15. Этапы работы программы.

### **Примеры билетов письменного экзамена**

#### **Экзаменационный лист 1 (1 семестр)**

1. Перечислите способы описания алгоритмов
2. Укажите недостатки(-) и преимущества(+) описания алгоритмов с помощью графических схем.

3. Перечислите стандартные (базовые) типы данных, используемые в Pascal:
4. Что определяет тип данных, выбранный для конкретного объекта в Pascal-программе
5. Укажите назначение и общий вид оператора присваивания:
6. Какого типа будет результат, если в арифметическом выражении оператора присваивания есть данные и вещественного и целого типа?
7. Определите тип и значения следующих выражений:
  - a.  $2 * 5 + 9 \text{ div } 4 * 3 + \text{TRUNC}(3.67)$ ;
  - b.  $C + \text{Ord}(\text{Round}(A) + \text{Ord}(B)) - \text{Trunc}(A)$ ;  
(здесь  $A=7.91(\text{real})$ ;  $B=\text{true}(\text{Boolean})$ ;  $C=5(\text{Integer})$ ).
8. При каких значениях переменных A, B, C будет ложным логическое выражение:
 
$$\text{Not}(A \text{ And } \text{Not}(B \text{ Or } \text{Not } C))$$
9. В каких случаях в программе используется цикл с предусловием?
10. Запишите вид операторов цикла с предусловием на языке Pascal.
11. Возможно ли, что цикл с предусловием не отработает ни разу? Ответ поясните.
12. Сколько раз выполнится тело цикла в данном фрагменте программы:
 

```
B := False; r := 45;
While Not B Do Begin B := r = 15; r := r mod 4 + 15 End;
```
13. В каких случаях предпочтительнее использовать отдельную переменную для хранения данных, а в каких – организовывать для этого массив?
14. Как объявить в программе матрицу вещественных чисел из 7 столбцов и 15 строк:
15. Какое логическое выражение нужно вписать во фрагмент программы, определяющий индекс первого элемента массива P[1..n], не равного x?
 

```
m := 0;
for i := n downto 1 do
if _____ then m := i;
```
16. Дано натуральное число N. Что вычисляется в данном фрагменте программы?
 

```
M := 0;
While N <> 0 Do
Begin If N Mod 10 > M Then M := N mod 10; N := N div 10
End;
```
17. Как организовать вывод вещественных данных в нестандартной форме? Укажите вид этого оператора вывода. Приведите пример.
18. Перечислите типы реализации подпрограмм в Pascal.
19. Укажите общий вид описания (заголовка) процедуры.

20. Какие виды параметров могут быть указаны в заголовке процедуры при ее описании?

### Экзаменационный лист 2 (2 семестр)

1. Дано следующее описание переменных:  
Var A,B:extended; W:Word; J:integer; M, N:byte;  
Какие из этих переменных совместимы по присваиванию?
2. К процедурам/функциям работы со строками не относятся  
1) Insert 2) Dispose 3)Copy 4)Concat 5>Delete.
3. Определить результат выполнения функции Val,  
здесь – IntNum переменная Integer, RealNum переменная Real;  
a. Val(' -3507', IntNum, Error);  
b. Val('1.23E3', IntNum, Error);  
c. Val('1.23E\_3', RealNum, Error);
4. Вычислить следующие выражения:  
a. [9] <= [1 .. 9]; c. [7, 1,3..6] = [1 .. 7];  
b. ['A'..'D', 'K'..'M'] + ['D'..'K']; d. 15 IN [1.. 10];
5. Могут ли компоненты (поля) записи быть различных типов? Приведите пример, поясняющий ваш ответ
6. Как в записи задается Вариантная часть? Приведите пример, поясняющий ваш ответ.
7. Объясните разницу между внутренним и внешним именем файла.
8. Перечислите функции/процедуры, используемые для работы только текстовыми файлами:
9. Дано описание var f: file of integer; x, y: integer; Файл f содержит элементы 3,7. Определить значение Y после выполнения операторов:  
reset (f); y :=0;  
while not eof(f) do begin read(f,x); y := y+x; end;
10. К файлам какого типа в TP можно обратиться с помощью прямого доступа?
11. Типизированный указатель – это...  
Приведите примеры описания нетипизированных указателей в виде типов и переменных.
12. Опишите структуру библиотечного модуля
13. Назначение библиотеки DOS:
14. Перечислите 3-4 процедуры/функции библиотеки DOS
15. Кратко опишите механизм построения изображения в графическом режиме.

## Критерии выставления оценки на экзамене

Баллы рейтинговой оценки	Оценка экзамена (стандартная)	Требования к сформированным компетенциям
От 86% до 100%	«Отлично»	Выставляется студенту, если он глубоко и прочно усвоил программный материал курса, четко и последовательно излагает его, знает базовые алгоритмы; способы представления различных видов информации на ЭВМ.; владеет уверенными навыками работы в интегрированной среде разработки программ, использует технологию и средства структурного программирования..
От 76% до 85%	«Хорошо»	Выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, правильно применяет теоретические положения при решении практических вопросов и задач, владеет необходимыми навыками и приемами их выполнения.
От 61% до 75%	«Удовлетворительно»	Выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических заданий: реализовывает базовые алгоритмы с ошибками.
Менее 61%	«Неудовлетворительно»	Выставляется студенту, который не знает значительной части программного материала: основные понятия языка программирования высокого уровня; основные структуры данных Pascal и способы их реализации; основные конструкции Pascal., неуверенно, с большими затруднениями выполняет практические задания по составлению алгоритмов и программ.

Составитель: \_\_\_\_\_ Е.А. Елсукова 2017г.

## Оценочные средства для текущей аттестации

### Вопросы для собеседования (УО-1)

#### Раздел 1. Введение в дисциплину

1. Перечислить особенности решения задач с использованием ЭВМ?

2. На какие этапы разбивается решение задач на ЭВМ, какими операциями они могут быть дополнены?
3. Чем отличается этап постановки задачи от этапа формализации?
4. Что такое алгоритм, и его свойства?
5. Какие существуют способы описания алгоритмов? Охарактеризуйте (преимущества и недостатки) эти способы.
6. Какие правила должны выполняться при оформлении схем алгоритмов?
7. Что содержит команда на машинном языке?
8. Перечислить основные составляющие язык программирования высокого уровня:
9. Что образует синтаксис языка программирования?
10. Назначение семантики языка программирования.

## **Раздел 2. Данные языка программирования Pascal**

1. Что определяет тип данных?
2. Для данных целого, логического и символьного типов указать: диапазон значений, допустимые операции, представление в оперативной памяти.
3. Что соответствует простой величине?
4. Чем определяется представление данных вещественного типа?
5. Привести собственные примеры задач, в которых используются данные только целого (вещественного) типа.
6. Для каких данных следует использовать перечисляемый тип? Приведите примеры.
7. Приведите собственные примеры корректного описания диапазонов.
8. Указать характеристики структурной величины.
9. Для каких задач данные следует организовать в массив, а не представлять простыми переменными?
10. Какого типа не может быть индекс массива?
11. Как определяется положение элемента в массиве?
12. Обязательно ли указывать при объявлении строкового типа данных длину строки? Ответ поясните.
13. Структура таблицы ASCII.
14. Назначение процедур VAL, STR. Приведите примеры задач, для решения которых следует использовать данные процедуры?
15. Для данной строки символов привести несколько способов описания.
16. Поясните, почему длина строки не может превышать 255 символов?
17. В чем заключается существенное отличие между массивами и записями?

18. Какие служебные слова определяют список полей записи?
19. Каких типов не может быть идентификатор поля записи?
20. Укажите все способы обращения к полю записи.
21. Какое определение называется рекурсивным? Приведите собственные примеры рекурсивных определений.
22. Какой вспомогательный алгоритм (подпрограмма) называется рекурсивными? Приведите собственные примеры содержательных задач, где для решения может быть использован рекурсивный вспомогательный алгоритм.
23. Что такое граничное условие и каково его назначение в рекурсивной подпрограмме?
24. Что такое рекурсивный спуск, рекурсивный подъём, глубина рекурсии?
25. На каком этапе выполнения рекурсивной подпрограммы могут выполняться её операторы?
26. Перечислить основные этапы работы с файлами.
27. Какие из процедур должны обязательно предшествовать процедуре Read?
28. Какие файлы называются системными? Укажите их назначение, приведите собственные примеры описания.
29. Чем отличается структура текстового файла от типизированного файла?
30. Для каких типов файлов используются прямой доступ к данным?

### **Раздел 3. Основные управляющие структуры ЯП**

1. Перечислить основные этапы создания модуля пользователя.
2. Для чего предназначена при создании модуля секция инициализации?
3. В каком разделе модуля (библиотеки) содержатся тексты подпрограмм?
4. Как формируется изображение в графическом режиме работы монитора?
5. Перечислить основные этапы создания анимации.

### **Критерии оценки:**

- ✓ **5 баллов** выставляется студенту, если активно работал на занятиях.
- ✓ **4 балла** выставляется студенту, если активно работал на занятиях, но допустил неточности при обсуждении.
- ✓ **3 балла** выставляется студенту, если активно работал на одном занятии, допустил многочисленные неточности.

Составитель: \_\_\_\_\_ Е.А. Елсукова 2017г

## Комплект заданий для контрольной работы (Пр-2)

### Контрольная работа 1. : *Логические и арифметические выражения; условный оператор*

Контрольная работа проводится после изучения тем: “Выражения Pascal”, “Оператор присваивания”, “Условный оператор”.

Для успешного выполнения контрольной работы необходимо знать:

- ✓ Назначение и вид оператора присваивания, условного оператора в полной и неполной формах;
- ✓ Основные логические операции;
- ✓ Правила записи арифметических и логических выражений.

#### Пример задания на контрольную работу 1 (Вариант I)

1. Дана логическая функция  $\text{not } (x_2+x_3) \bullet (\text{not}x_3+\text{not}x_1)$ . Построить ее таблицу истинности.
2. Изобразить на плоскости  $(X, Y)$  область, в которой и только в которой, истинно указанное выражение  
 $(X \leq 0) \text{ and } (\text{Sqr}(X)+\text{Sqr}(Y) \leq 4) \text{ or } (X \geq 0) \text{ and } (Y \leq -X+2) \text{ and } (Y \geq X-2)$
3. Объяснить, в чем заключается синтаксическая ошибка (или ошибки) в приведенной программе. Написать правильный (по Вашему мнению) вариант программы.

```
Program MyError;  
Var X : Real;  
Begin  
  WriteLn(Введите значение X); ReadLn(X);  
  Y := SQR(X) - 1;  
  If Y < 0 Then Y := Y * -1  
  WriteLn(Y);  
End.
```

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

```
Program Kr_3;  
Var X1, Y1, X2, Y2, R1, R2 : Real;  
Begin  
  WriteLn('Введите X1, Y1');  
  ReadLn(X1, Y1);  
  WriteLn('Введите X2, Y2');  
  ReadLn(X2, Y2);  
  R1 := SQRT(X1 * X1 + Y1 * Y1);  
  R2 := SQRT(X2 * X2 + Y2 * Y2);  
  If R1 = R2 Then WriteLn('Лежат') Else WriteLn('Не лежат');
```

End.

5. По данному утверждению составить логическое выражение, значение которого равно True, если высказывание истинно, и False, — если ложно.

В двузначном натуральном числе **n** первая цифра меньше второй.

## **Контрольная работа 2. Тема: *Функции обработки строк. Операции над множествами***

Контрольная работа проводится после изучения тем: “Множества”, “Обработка текстовой информации на Pascal”.

**Для успешного выполнения контрольной работы необходимо знать:**

- ✓ способы описания множеств, виды множеств;
- ✓ формирование множеств с помощью конструкторов;
- ✓ операции для работы с множествами.
- ✓ Способы описания строковых данных.
- ✓ Подпрограммы (функции и процедуры) обработки строк.
- ✓ Обработка длинных строк .

### **Пример задания на контрольную работу 2 (Вариант I)**

1. Описать общий вид и назначение POS
2. Как обозначается операция конкатенация
3. Определите результаты обращений к процедурам и функциям. Все строковые переменные относятся к типу string[20], а исходными значениями переменных Temp1, Temp2 являются строки соответственно ‘Abra’ и ‘cadabra’.
  - a) Magic := Concat (Temp1, Temp2)
  - b) Length (Magic)
  - c) HisMagic:= Copy (Magic, 1,8)
  - d) Delete (HisMagic, 4,3)
  - e) Insert (Temp1, HisMagic, 3)
  - f) Pos (Temp1, Magic)
  - g) Val (‘1.234’, RealNum, Error)
  - h) Str (1.234:3:1, RealStr)
4. Какие из следующих конструкций не является множествами (в понятиях ЯП Pascal)? Ответ поясните.
  - a) [2..3, 5, 7]
  - b) [1.. 15, 4..18]
  - c) [true, false]
  - d) [9, 6, 3, 0]



- e) [2..10, 5..15]
  - f) [true, false]
  - g) [sqr(3), sqr(4), sqr(5)]
5. A – множество [1, 3, 5, 7], B - множество [2, 4, 6] и C - множество [1, 2, 3]. Вычислить значения следующих выражений:
- a)  $A + (B - C)$
  - b)  $(C - A) \leq B$
  - c)  $[] \leq A * B * C$
  - d)  $A + B \langle \rangle [1..7]$
  - e)  $C + (A - C)$
  - f)  $(C - A) - B$
  - g)  $A - C - [5, 7] = []$
  - h)  $2 \text{ IN } A + B$

**Критерии оценки:**

- ✓ **5 баллов** выставляется студенту, если все задания выполнены верно, без ошибок.
- ✓ **4 балла** выставляется студенту, если все задания выполнены верно, есть 2-3 неточности.
- ✓ **3 балла** выставляется студенту, если 1 задание выполнено неверно, допущены 2-3 неточности при решении других задач.
- ✓ **2 балла** выставляется студенту, если две и более задачи решены неверно.

Составитель: \_\_\_\_\_ Е.А. Елсукова

2017г.