

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)

III. Рабочая программа пересмотрена на заседании кафедры:

Протокол от « ____ » _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

IV. Рабочая программа пересмотрена на заседании кафедры:

Протокол от « ____ » _____ 20__ г. № _____

Заведующий кафедрой _____
(подпись) (И.О. Фамилия)

Цели и задачи освоения дисциплины:

Цель: 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 систему программирования PascalABC.NET;
- получить навыки оформления программной документации.

Для успешного изучения дисциплины «Основы программирования» 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 алгоритмы; тестировать готовые программы.

В результате изучения данной дисциплины у обучающихся формируются следующие профессиональные компетенции:

Тип задач	Код и наименование профессиональной компетенции (результат освоения)	Код и наименование индикатора достижения компетенции
производственно-технологический	ПК-3 Способен изготавливать компоненты информационных систем, включая программные комплексы, базы данных и интерфейсы "человек - электронно-вычислительная	ПК-3.1 Использует технологию программирования, системы баз данных, сетевые технологии, методы, формализмы и стандарты представления диалогов и экранных форм интерфейсов «человек – ЭВМ», инструментальные средства разработки и программно-технологические платформы информационных систем

Тип задач	Код и наименование профессиональной компетенции (результат освоения)	Код и наименование индикатора достижения компетенции
	машина", использовать современные инструментальные средства разработки, и программно-технологические платформы информационных систем	ПК-3.2 Осуществляет изготовление спецификаций программного обеспечения, определяет и устанавливает параметры используемых коммерческих программных пакетов, контролирует качество, создаваемого программного обеспечения, создает базы данных, устанавливает программное обеспечение и осуществляет загрузку баз данных, разработку технической документации, проводит начальное обучение и консультирование пользователей
		ПК-3.3 Применяет системные и прикладные программно-технологические платформы, стандарты визуального представления интерфейсов и электронных форматов данных, средства CASE, инструментальные средства разработки информационных систем и программирования

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
ПК-3.1 Использует технологию программирования, системы баз данных, сетевые технологии, методы, формализмы и стандарты представления диалогов и экранных форм интерфейсов «человек – ЭВМ», инструментальные средства разработки и программно-технологические платформы информационных систем	Знает назначение и содержание основных понятий языка программирования высокого уровня (концепцию типов данных, основные конструкции).
	Умеет осуществлять постановку задачи программирования, разрабатывать алгоритм ее решения используя технологию и средства структурного программирования.
	Владеет навыками алгоритмизации и программирования при решении задач на ЭВМ;
ПК-3.2 Осуществляет изготовление спецификаций программного обеспечения, определяет и устанавливает параметры используемых коммерческих программных пакетов, контролирует качество,	Знает основные этапы решения задач на ЭВМ и стандарты программной документации.
	Умеет реализовывать алгоритмы на языке программирования высокого уровня; Умеет оценить сложность и эффективность алгоритмов

Код и наименование индикатора достижения компетенции	Наименование показателя оценивания (результата обучения по дисциплине)
создаваемого программного обеспечения, создает базы данных, устанавливает программное обеспечение и осуществляет загрузку баз данных, разработку технической документации, проводит начальное обучение и консультирование пользователей	Владеет навыками тестирования программ и оформления программной документации.
ПК-3.3 Применяет системные и прикладные программно-технологические платформы, стандарты визуального представления интерфейсов и электронных форматов данных, средства CASE, инструментальные средства разработки информационных систем и программирования	Знает способы представления различных видов информации на ЭВМ; , методы формализации, алгоритмизации и реализации программного обеспечения с помощью языков программирования.
	Умеет использовать методы отладки и решения задач на ЭВМ в различных режимах (инструментальные средства разработки программ).
	Владеет навыками работы в интегрированной среде разработки программ PascalABC.Net.

2. Трудоёмкость дисциплины и видов учебных занятий по дисциплине

Общая трудоёмкость дисциплины составляет 16 зачётных единиц; 576 академических часов (1 зачетная единица соответствует 36 академическим часам).

Видами учебных занятий и работы обучающегося по дисциплине являются:

Обозначение	Виды учебных занятий и работы обучающегося
Лек	Лекции
Лаб	Лабораторные работы
СР	Самостоятельная работа обучающегося в период теоретического обучения
Контроль	Самостоятельная работа обучающегося и контактная работа обучающегося с преподавателем в период промежуточной аттестации

Структура дисциплины:

Форма обучения – очная.

№	Наименование раздела дисциплины	Семестр	Количество часов по видам учебных занятий и работы обучающегося						Формы промежуточной аттестации
			Лек	Лаб	Пр	ОК	СР	Контроль	
1.	Раздел 1. Введение в дисциплину	1	8	2	-	-	243	117	УО-1; ПР-3; ПР-6.
2.	Раздел 2. Данные языка	2	24						

№	Наименование раздела дисциплины	Семестр	Количество часов по видам учебных занятий и работы обучающегося						Формы промежуточной аттестации
			Лек	Лаб	Пр	ОК	СР	Контроль	
	программирования Pascal								
3.	Раздел 3. Основные управляющие структуры программирования	1	14						
4.	Раздел 4. Разработка прикладных программ	1	12						
5.	Раздел 5. Структурное программирование	2	14						
6.	Итого:		72	144		-	243	117	

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА (72 час.)

Раздел I. Введение в дисциплину (8 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 (2 час.)

Понятие алгоритма; его свойства. Способы представления алгоритмов. Преимущества и недостатки способов. Правила построения схем 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 (24 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.

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 Хоара.

Тема 4. Эвристические алгоритмы поиска (2 час.)

Понятие эвристического 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. Задача численного интегрирования: метод прямоугольников, метод трапеции, метод Симпсона; точность вычислений.

Раздел V. Структурное программирование (14 час.)

Тема 1. Модули в Pascal (2 час.)

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

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

Императивное программирование, Теорема структуры. Структурное программирование: основные характеристики. Жизненный цикл программы.

Тема 5. Этап отладки и тестирования (4 час.)

Корректность программ; способы конструирования и верификации программ. Анализ программ.

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА И САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Лабораторные работы (144 час.)

Лабораторные работы 1 семестра (72 час.)

Лабораторная работа 1. Способы описания алгоритмов. Исполнение алгоритмов (2 час.)

Лабораторная работа 2. Решение физических задач (2 час.)

Лабораторная работа 3. Разветвляющиеся алгоритмы (4 час.)

Лабораторная работа 4. Описание функции по её графическому представлению (6 час.)

Лабораторная работа 5. Алгоритмы целочисленной арифметики (4 час.)

Лабораторная работа 6. Циклические алгоритмы (4 час.)

Лабораторная работа 7. Табуляция функций(4 час.)

Лабораторная работа 8. Вычисление суммы ряда с заданной точностью (4 час.)

Лабораторная работа 9. Алгоритмы обработки одномерных массивов (4час.)

Лабораторная работа 10. Алгоритмы с использованием подпрограмм (4 час.)

Лабораторная работа 11. Программирование численных методов (6 час.)

Лабораторная работа 12. Алгоритмы обработки двумерных массивов (матриц) (6 час.)

Лабораторная работа 13. Основные действия на матрицами (6 час.)

Лабораторная работа 14. Организация рекурсии (4 час.)

Лабораторная работа 15. Реализация метода сортировки и оценка его эффективности (6 час.)

Лабораторная работа 16. Геометрия на плоскости (6 час.)

Лабораторные работы 2 семестра (72 час.)

Лабораторная работа 1. Алгоритмы поиска (4 час.)

Лабораторная работа 2. Работа с библиотекой CRT (6 час.)

Лабораторная работа 3. Решение задач на обработку строкового типа данных (6 час.)

Лабораторная работа 4. Работа с различными типами файлов (8 час.)

Лабораторная работа 5. Работа с перечисляемыми типами (4 час.)

Лабораторная работа 6. Использование множеств при обработке символьных и числовых массивов (4 час.)

Лабораторная работа 7. Использование записей при обработке статистических данных (6 час.)

Лабораторная работа 8. Создание пользовательских библиотек (4 час.)

Лабораторная работа 9. Работа с динамическими структурами (8 час.)

Лабораторная работа 10. Работа с графикой (6 час.)

Лабораторная работа 11. Алгоритмы создания анимации (4 час.)

Лабораторная работа 12. Обработка комбинированных типов данных (6 час.)

Лабораторная работа 13. Реализация жадных алгоритмов (6 час.)

III. III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

План-график выполнения самостоятельной работы»

1 семестр

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1.	В течение семестра	Подготовка к лабораторной работе (активная форма проведения)	18	УО-1
2.	В течение семестра	Оформление отчетов по лабораторным работам (6 отчетов)	25	УО-1
3.	6 неделя семестра	Подготовка к контрольным	18	Пр-2

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
	(КР 1); 14 неделя семестра (КР 2).	работам		
4.	1-6 недели семестра	Самостоятельное изучение отдельных тем дисциплины	20	УО-1
5.		Подготовка к экзамену	63	ПР-2
Всего в семестре			144 часа	

2 семестр

№ п/п	Дата/сроки выполнения	Вид самостоятельной работы	Примерные нормы времени на выполнение	Форма контроля
1.	В течение семестра	Курсовая работа	108	Пр-2
2.	В течение семестра	Подготовка к лабораторной работе (активная форма проведения)	24	УО-1
3.	В течение семестра	Оформление отчетов по лабораторным работам (5 отчетов)	30	УО-1
4.		Подготовка к экзамену	54	ПР-2
Всего в семестре			216 часов	

Методические рекомендации к самостоятельному изучению тем

Для самостоятельного изучения предлагаются следующие темы:

- Эволюция языков программирования;
- Освоение возможностей интегрированной среды PascalABC.NET.

Список литературы, рекомендации и вопросы для самопроверки приводятся в LMS BlackBoard в разделе “Материалы для организации самостоятельной работы студентов”.

Методические рекомендации для подготовки к контрольным работам

На лекционных занятиях проводятся контрольные работы по отдельным разделам дисциплины. Цель проведения контрольных работ - определить уровень усвоения студентами теоретических знаний по данным темам.

Темы контрольных работ I семестра

1. Реализация алгоритмических структур на Pascal.
2. Создание и обработка массивов.

Методические рекомендации к оформлению отчетов по лабораторным работам

Подробные рекомендации по составлению отчетов, требования к оформлению и наглядные примеры приводятся в LMS BlackBoard в разделе “Учебные материалы. Указания по оформлению работ”.

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

1. Титульный лист *.
2. Содержание *.
3. Задание.
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. Приложения * (при необходимости, обычно в Приложении выносят листинг программы)

* Включаются в отчет с новой страницы

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

№ п/п	Контролируемые модули/разделы /темы дисциплины	Код индикатора достижения компетенции	Результаты обучения	Оценочные средства – наименование	
				текущий контроль	промежуточная аттестация
1.	Введение в дисциплину	ПК-3.1	Знает назначение и содержание основных понятий языка программирования высокого уровня;	Собеседование (УО-1)	вопросы к экзамену 2÷3,5, 8, 14 (I); 1-10 (II);

№ п/п	Контролируемые модули/разделы /темы дисциплины	Код индикатора достижения компетенции	Результаты обучения	Оценочные средства – наименование	
				текущий контроль	промежуточная аттестация
2.	Данные языка программирования Pascal	ПК-3.3	Знает основные структуры данных Pascal и способы их реализации.	Собеседование (УО-1); Тестирование (ПР-2)	вопросы к экзамену 2, 3, 7,14 -15 (I), 1-10 (II)
			Умеет организовать обработку различных типов и структур данных	Лабораторные работы (ПР-6) 4,5 (I), 1-4, 6 (II)	
			Знает основные структуры данных Pascal и способы их реализации.	Лабораторные работы (ПР-6) 4,5 (I), 1-4, 6 (II)	
3.	Основные управляющие структуры программирования	ПК-3.2	Знает основные конструкции Pascal.	Собеседование (УО-1); Тестирование (ПР-2)	вопросы к экзамену 1÷13 (I)
			Умеет осуществлять постановку задачи программирования	Лабораторные работы (ПР-6) 2,3 (I)	
			Владеет методами отладки и решения задач на ЭВМ в различных режимах.	Лабораторные работы (ПР-6) 2,3,5 (I)	
4.	Разработка прикладных программ	ПК-3.2	умеет осуществлять постановку задачи программирования, разрабатывать алгоритм ее решения	Лабораторные работы (ПР-6) 1,2,3 (I) 1,2,3,5 (II),	вопросы к экзамену 15, 19 (I)
			владеет навыками алгоритмизации и программирования при решении задач на ЭВМ.	Лабораторные работы (ПР-6) 6,7 (I), 1,2,3,5 (II),	
5.	Структурное программирование	ПК-3.2	Умеет создавать компьютерные программы, используя технологию и средства структурного программирования.	Лабораторные работы (ПР-6) 6,7 (I)	вопросы к экзамену 1÷4,20 (I); 11-15 (II)
			владеет методами отладки и решения задач на ЭВМ в различных режимах (использует инструментальные средства разработки программ).	Лабораторные работы (ПР-6) 4-6 (II)	

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. Павловская Т. А. Паскаль. Программирование на языке высокого уровня. Практикум. 2-е изд. – СПб: Питер, 2007. – 317с.
3. Основы алгоритмизации и программирования (среда PascalABC.NET) : учеб. пособие / И.Г. Фризен. — Москва: ФОРУМ : ИНФРА-М, 2017. — 392 с. — Режим доступа: URL: <http://znanium.com/catalog/product/559358>
4. Окулов С.М. Основы программирования [Электронный ресурс]: учебное пособие/ Окулов С.М.— Электрон. текстовые данные.— М.: БИНОМ. Лаборатория знаний, 2012.— 340с. — Режим доступа: URL: <http://www.iprbookshop.ru/6449>.

Дополнительная литература

(печатные и электронные издания)

1. Потопахин В. Искусство алгоритмизации [Электронный ресурс] : . — Электрон. дан. — М. : ДМК Пресс, 2011. — 320 с. — Режим доступа: URL: http://e.lanbook.com/books/element.php?p11_id=1269
2. Кнут Д. Искусство программирования, Т.1,2,3. – М.: Издательский дом “Вильямс”, 2010.
3. Малыхина М.П. Программирование на языке высокого уровня. Turbo Pascal. Уч. пособие. – СПб: БХВ-Петербург, 2006. – 544с.
4. Порублев И.Н., Ставровский А.Б. Алгоритмы и программы. Решение олимпиадных задач. — М.: Издательский дом “Вильямс”, 2007. – 480с.
5. Рапаков Г.Г., Ржеуцкая С.Ю. Turbo Pascal для студентов и школьников. – СПб.: БХВ-Петербург, 2012. – 352с.
6. Бентли Дж. Жемчужины программирования. 2-е изд. – СПб: Питер, 2002. – 272с.
7. 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 «Основы программирования» 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

Результаты освоения дисциплины достигаются за счет использования в процессе обучения: лекций с применением мультимедийных технологий, активных методов обучения с использованием LMS Blackboard; лабораторных работ на базе компьютерной сети.

Цель лабораторных занятий - закрепить теоретические знания, полученные на лекционных занятиях и приобрести практические навыки разработки программ.

Задание на лабораторную работу обучающиеся получают заранее на лекции соответствующей тематики для качественной подготовки к ее выполнению. Перед выполнением лабораторной работы студент должен ознакомиться с заданием и порядком выполнения на странице курса в LMS Blackboard.

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 Белоус Е. И.,
доцент кафедры психологии Школы гуманитарных наук ДВФУ,
кандидат педагогических наук*

Общие требования к курсовой работе

По данной дисциплине предусмотрено выполнение курсовой работы во 2 семестре.

Выполнение курсовой работы является заключительным этапом освоения студентами дисциплины «Основы программирования».

Цель курсовой работы - закрепление приобретённых навыков алгоритмизации задач с использованием структурного подхода и самостоятельное освоение этапов создания программного продукта.

Порядок выдачи задания: Задание на курсовую работу выдается руководителем курсового проектирования в первые две недели второго семестра на консультации.

Если у студента есть пожелания по выбору темы, то необходимо об этом заранее сообщить руководителю курсового проектирования. Студент может предложить индивидуальную тему, ее сложность должна быть сопоставима с типовыми. Тематика, содержание курсовой работы должно быть

предварительно обсуждено с преподавателем и оформлено в виде задания. После официального получения задания тема не может быть изменена.

Тематика курсовых работ

1. Реализация методов вычислительной математики (Обработка табличных данных - Интерполяция; Аппроксимация; Решение нелинейных уравнений; Численное интегрирование; Одномерная оптимизация)
2. Обработка данных различных предметных областей (Библиотека, Цветочный магазин, Туристическое агентство, Домашняя медиатека, Доска объявлений и др.)
3. Игры.
4. Обучающие программы по разделам курса.

Состав курсовой работы

К работе оформляется отчет (описание / пояснительная записка). Отчет должен иметь объем не менее 15 страниц текста, выполнен в соответствии с требованиями ДВФУ. Отчет должен содержать следующие разделы:

1. Титульный лист
2. Задание
3. Оглавление
4. Ключевые слова (Словосочетания)
5. Теоретический материал
6. Структурное описание разработки
7. Функциональное описание
8. Заключение
9. Список используемой литературы
10. Приложения.

Содержание разделов отчета

Титульный лист (1) должен быть оформлен в соответствии с общими требованиями к курсовым работам, действующим в ДВФУ.

Задание (2) на специальном бланке выдается руководителем курсового проектирования.

Оглавление (3) должно строиться средствами текстового редактора.

В п.4 необходимо привести 5-6 ключевых слов (или ключевых словосочетаний), которые отражают основной смысл, являются характерными (ключевыми) для данной работы.

В теоретической части (5) должен быть рассмотрен один из аспектов программирования, связанный с реализацией поставленной задачи. Теоретический материал должен быть переработан применительно к содержанию работы.

В разделе “Структурное описание разработки” (6) нужно рассмотреть как будет использоваться в практической части работы (программе) изложенный теоретический материал; какие основные решения приняты при разработке. Привести содержательное описание основных алгоритмов, их особенностей; описание используемых данных, особенности проектирования.

Структурное описание – словесное, с привлечением необходимых графических иллюстраций (схем, рисунков), фрагментов оригинальных алгоритмов и структур данных.

В разделе “Функциональное описание” (7) следует рассмотреть работу программы. Привести описание: форматов данных (отдельно указав какие данные являются статическими, какие – динамическими); используемых подпрограмм, их интерфейсов; фрагментов реализации нетривиальных алгоритмов; работы программы на контрольных примерах (например, на большом текстовом файле). Для описания использовать смешанное текстово-формальное представление программы с включением фрагментов программы в связный текст изложения материала.

Заключение (8) содержит краткие выводы по результатам курсовой работы и применения теоретических знаний в практическом программировании.

В приложения (10) обычно включают: схемы, рисунки, графическое описание данных, формат выходного документа, схему последовательности обработки данных, листинг программы с комментариями по существу алгоритма и структур данных.

Защита курсовой работы

Курсовая работа должны быть представлена к защите в течение трех последних учебных недель второго семестра. Курсовая работа предварительно сдается для проверки руководителю. Работа сдается в бумажном (отчет) и электронном (файлы с программой) виде за 2-3 дня до защиты.

К защите следует подготовить доклад в форме презентации MS Power Point с 6-7 слайдами или средствами HTML, который должен отражать основные положения курсовой работы.

Защита выполняется с демонстрацией на ПК работы программы и необходимыми пояснениями по ее проектированию и программированию. Студент должен свободно ориентироваться в представленном материале.

Студент демонстрирует свою работу на тестовых примерах. Если у преподавателя имеются замечания принципиального характера, а работа

защищается в назначенный срок или раньше, то автор может отсрочить защиту, исправив замечания, для получения более высокой оценки.

Замечания, выявленные в процессе проверки должны быть устранены в течение 2-х недель после ее предъявления. Для всех, кто защищает курсовые на зачетной неделе, возможность доработки на более высокую оценку не предоставляется.

При выставлении оценки за курсовую работу учитывается реализация требований, полнота решения поставленной задачи, качество программной реализации, оформление отчета (пояснительной записки).

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Перечень материально-технического и программного обеспечения дисциплины приведен в таблице.

Материально-техническое и программное обеспечение дисциплины

Наименование специальных помещений и помещений для самостоятельной работы	Оснащенность специальных помещений и помещений для самостоятельной работы	Перечень лицензионного программного обеспечения. Реквизиты подтверждающего документа
690922, Приморский край, г. Владивосток, остров Русский, полуостров Саперный, поселок Аякс, 10, корпус D, ауд. Учебная аудитория для проведения занятий лекционного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	Помещение укомплектовано специализированной учебной мебелью (посадочных мест – 30) Оборудование: ЖК-панель 47", FullHD, LGM4716 CCBA – 1 шт. Доска аудиторная.	ПЕРЕЧЕНЬ ПО

В процессе изучения дисциплины используются мультимедийные и технические средства обучения. При проведении занятий используются аудитории со средствами вычислительной техники:

- рабочие станции или виртуальные машины на рабочих станциях с доступом к сети Интернет.
- Персональный компьютер преподавателя с мультимедиа-проектором и экраном, программным обеспечением для демонстрации презентаций.

Для самостоятельной работы с медиаматериалами каждому студенту требуется персональный компьютер или планшет, широкополосный доступ в

сеть Интернет, браузер последней версии, устройство для воспроизведения звука (динамики, колонки, наушники и др.).

В целях обеспечения специальных условий обучения инвалидов и лиц с ограниченными возможностями здоровья в ДВФУ все здания оборудованы пандусами, лифтами, подъемниками, специализированными местами, оснащенными туалетными комнатами, табличками информационно-навигационной поддержки.

VIII. ФОНДЫ ОЦЕНОЧНЫХ СРЕДСТВ

Для дисциплины «Основы программирования» используются следующие оценочные средства:

Устный опрос:

1. Собеседование (УО-1)

Письменные работы:

1. Лабораторная работа (ПР-6)

Методические рекомендации, определяющие процедуры оценивания результатов освоения дисциплины

Промежуточная аттестация студентов по дисциплине «Основы программирования» проводится в соответствии с локальными нормативными актами ДВФУ и является обязательной.

По данной дисциплине предусмотрены следующие виды промежуточной аттестации – экзамен, экзамен проводится в письменной форме с использованием оценочного средства: тест / письменный экзамен.

Промежуточная аттестация студентов

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. $2 * 5 + 9 \text{ div } 4 * 3 + \text{TRUNC}(3.67)$;
 - б. $C + \text{Ord}(\text{Round}(A) + \text{Ord}(B)) - \text{Trunc}(A)$;
(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:
 $B := \text{False}; r := 45$;
 $\text{While Not } B \text{ Do Begin } B := r = 15; r := r \text{ mod } 4 + 15 \text{ End}$;
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 ТР 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.

Инструкция для студента

Билет письменного экзамена состоит из 10 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.**

Критерии выставления оценки на экзамене по дисциплине «Основы программирования»

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

Баллы рейтинговой оценки	Оценка экзамена (стандартная)	Требования к сформированным компетенциям
		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.

Оценочные средства для текущей аттестации

На лекционных занятиях проводятся контрольные работы по основным разделам курса. Цель проведения контрольных работ - определить уровень усвоения студентами и оценить качество их теоретических знаний по данным темам.

Выполнение указанных видов работ является обязательным для всех студентов, по результатам текущего контроля выставляется аттестация.

Темы контрольных работ I семестра

1. Реализация алгоритмических структур на Pascal.
2. Создание и обработка массивов.

Для выполнения данной контрольной работы 1 студент должен ЗНАТЬ:

- ✓ Правила записи арифметических и логических выражений;
- ✓ Назначение и вид оператора присваивания, условного оператора в полной и неполной формах; операторов цикла;
- ✓ Правила применения операторов цикла.

Пример задания на контрольную работу 1

1. Изобразить на плоскости (X, Y) область, в которой и только в которой, истинно указанное выражение

$$(\text{Sqr}(X) + \text{Sqr}(Y) \leq 1) \text{ and } (Y \leq 0.5 * X)$$

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

```

Program What_Problem;
Var I, N : LongInt; S : Real;
Begin
  Write('Введите натуральное число: '); ReadLn(N);
  S := 0;
  For I := 1 To N Do
    If Odd(I) Then S := S + 1 / (Sqr(I) * I)
    Else S := S - 1 / (Sqr(I) * I);
  WriteLn('Ответ: ', S : 10 : 9)
End.

```

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

Каждая следующая цифра данного трехзначного натурального числа на 1 > предыдущей.

4. Составить программу для вычисления суммы:

$$S = 1 - 3 + 5 - 7 + \dots + (-1)^{n-1}(2n - 1).$$

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.