

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Степанов Павел Иванович

Должность: Руководитель НТИ НИЯУ МИФИ

Дата подписания: 23.02.2026 22:03:49

Уникальный программный ключ:

8c65c591e26b2d8e4609277401f101a071315

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский ядерный университет «МИФИ»

Новоуральский технологический институт –

филиал федерального государственного автономного образовательного учреждения
высшего образования «Национальный исследовательский ядерный университет
«МИФИ»

УТВЕРЖДЕНА

Ученым советом НТИ НИЯУ МИФИ

Протокол № 3 от 24.04.2023

Рабочая программа учебной дисциплины "Программирование"

Направление подготовки 09.03.01 – Информатика и вычислительная техника

Профиль подготовки Автоматизированные системы обработки информации и управления

Квалификация (степень) выпускника Академический бакалавр

Форма обучения очная

Семестр	1	2	3	Всего
Трудоемкость, ЗЕТ	3	5	3	11
Трудоемкость, ак. часов	108	180	108	396
Контактные занятия, в т. ч.:	72	72	36	180
- лекции	18	36	18	54
- практические занятия	18	-	-	18
- лабораторные занятия	36	36	-	72
- курсовая работа	-	-	18	18
Самостоятельная работа	9	72	72	153
Контроль	27	36	-	63
Форма итогового контроля	Экза- мен	Экза- мен	Зачет КР	

Программу составила
старший преподаватель кафедры АУ



Тихонова Е.В.

СОДЕРЖАНИЕ

1 ЦЕЛИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	4
2 МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВПО	4
3 ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ	4
4 ВОСПИТАТЕЛЬНЫЙ ПОТЕНЦИАЛ ДИСЦИПЛИНЫ	6
5 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	7
5.1 Структура курса «Программирование».....	7
5.2 Содержание лекционных занятий (1-й семестр) – 18 часов	8
5.3 Темы лабораторных занятий (1-й семестр) – 36 часов	9
5.4 Темы практических занятий (1-й семестр) – 18 часов	10
5.5 Содержание лекционных занятий (2-й семестр) – 36 часов	10
5.6 Темы лабораторных занятий (2-й семестр) – 36 часов	12
5.7 Содержание лекционных занятий (3-й семестр) – 18 часов	13
5.8 Этапы выполнения курсовой работы (3-й семестр) – 18 часов.....	14
5.9 Самостоятельная работа – 153 часа	15
6 ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ.....	16
7 СРЕДСТВА ДЛЯ КОНТРОЛЯ И ОЦЕНКИ.....	17
8 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	18
9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	20
Приложение 1. Перечень учебно-методического обеспечения самостоятельной работы студентов.....	22
Приложение 2. Методические указания для студентов по освоению дисциплины.....	23
Приложение 3. Балльно-рейтинговая система оценки.....	24
Приложение 4. Фонд оценочных средств	26

1 ЦЕЛИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Целью освоения учебной дисциплины «Программирование» является формирование базовых профессиональных компетенций по разработке программного обеспечения на языке программирования высокого уровня.

2 МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВПО

Данная учебная дисциплина входит в Общепрофессиональный модуль «Б1.О.03» ФГОС-3 по направлению подготовки ВПО «Информатика и вычислительная техника» профиля подготовки бакалавров «Автоматизированные системы обработки информации и управления».

Для освоения дисциплины «Программирование» студенты используют знания, умения и виды деятельности, формируемые одновременно с изучением дисциплин «Физика», «Высшая математика» Естественно-научного модуля и «Информатика» Общепрофессионального модуля.

Освоение дисциплины «Программирование» является необходимой для последующего изучения дисциплин: «Численные методы», «Операционные системы», «Теория информации и кодирования», «Базы данных», «Моделирование систем», «Системное программное обеспечение», «Объектно-ориентированное программирование», «Обработка экспериментальных данных на ЭВМ», а также для успешного прохождения учебной практики и итоговой государственной аттестации.

3 ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Профессиональные компетенции (ПК) в соответствии с задачами и объектами (областями знаний) профессиональной деятельности:

Код и наименование общепрофессиональной компетенции	Код и наименование индикатора достижения общепрофессиональной компетенции
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности;	З-ОПК-2 Знать: принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, используемых при решении задач профессиональной деятельности; У-ОПК-2 Уметь: выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности; В-ОПК-2 Владеть: навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности;
УКЦ-2 Способен искать нужные источники информации и данные, воспринимать, анализировать, запоминать и передавать информацию с использованием цифровых средств, а также с помощью алгоритмов при	З-УКЦ-2 Знать: методики сбора и обработки информации с использованием цифровых средств, а также актуальные российские и зарубежные источники информации в сфере профессиональной деятельности, принципы, методы и средства решения

работе с полученными из различных источников данных с целью эффективного использования полученной информации для решения задач	стандартных задач профессиональной деятельности с использованием цифровых средств и с учетом основных требований информационной безопасности; У-УКЦ-2 Уметь: применять методики поиска, сбора и обработки информации; с использованием цифровых средств, осуществлять критический анализ и синтез информации, полученной из разных источников, и решать стандартные задачи профессиональной деятельности с использованием цифровых средств и с учетом основных требований информационной безопасности; В-УКЦ-2 Владеть: методами поиска, сбора и обработки, критического анализа и синтеза информации с использованием цифровых средств для решения поставленных задач, навыками подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с использованием цифровых средств и с учетом требований информационной безопасности.
---	---

4 ВОСПИТАТЕЛЬНЫЙ ПОТЕНЦИАЛ ДИСЦИПЛИНЫ

Цели и задачи воспитания, воспитательный потенциал дисциплин:

Задачи воспитания (код)	Воспитательный потенциал дисциплин
Интеллектуальное воспитание В11 Формирование культуры умственного труда	Использование воспитательного потенциала дисциплин гуманитарного, естественнонаучного, общепрофессионального и профессионального модуля для формирования культуры умственного труда посредством вовлечения студентов в учебные исследовательские задания, курсовые работы и др.
Профессиональное и трудовое воспитание В14 Формирование глубокого понимания социальной роли профессии, позитивной и активной установки на ценности избранной специальности ответственного отношения к профессиональной деятельности, труду	1.Использование воспитательного потенциала дисциплин естественнонаучного и общепрофессионального модуля для: - формирования позитивного отношения к профессии инженера (конструктора, технолога), понимания ее социальной значимости и роли в обществе, стремления следовать нормам профессиональной этики посредством контекстного обучения, решения практических ситуационных задач. - формирования устойчивого интереса к профессиональной деятельности, способности критически, самостоятельно мыслить, понимать значимость профессии посредством осознанного выбора тематики проектов, выполнения проектов с последующей публичной презентацией результатов, в том числе обоснованием их социальной и практической значимости; - формирования навыков командной работы, в том числе реализации различных проектных ролей (лидер, исполнитель и пр.) посредством выполнения совместных проектов. 2. Использование воспитательного потенциала дисциплины для: - формирования навыков системного видения роли и значимости выбранной профессии в социально-экономических отношениях через контекстное обучение.
В15 Формирование психологической готовности к профессиональной деятельности по избранной профессии	Использование воспитательного потенциала дисциплин общепрофессионального модуля для: - формирования устойчивого интереса к профессиональной деятельности, потребности в достижении результата, понимания функциональных обязанностей и задач избранной профессиональной деятельности, чувства профессиональной ответственности через выполнение учебных, в том числе практических заданий, требующих строгого соблюдения правил техники безопасности и инструкций по работе с оборудованием в рамках лабораторного практикума.

5 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

5.1 Структура курса «Программирование»

Общая трудоемкость дисциплины составляет **11 зачетных единицы, 396 часов.**

№ п/п	Название темы/раздела учебной дисциплины	Виды учебной деятельности, включая самостоятельную работу студентов и трудоемкость (в часах)					Текущий контроль (форма* , неделя)	Максимальный балл за раздел	Индикаторы освоения компетенции
		Лекции	Лабораторные занятия	Практические занятия	Курсовая работа	Самостоятель- ная работа			
1 семестр									
1	Введение в алгоритмиза- цию и программирование	4	8	4	-	2	КТ1 (5)	16	3-ОПК-2 У-ОПК-2 В-ОПК-2 3-УКЦ-2 У-УКЦ-2 В-УКЦ-2
2	Структурный подход к программированию	8	14	8	-	4	ДЗ1 (12)	27	
3	Модульное программиро- вание	6	14	6	-	3	ПКР1 (14) КТ2 (17)	27	
Итого:		18	36	18	-	9		70	
Экзамен								30	
2 семестр									
4	Программирование аб- страктных типов данных	14	14	-	-	26	ПКР2 (4)	29	3-ОПК-2 У-ОПК-2 В-ОПК-2 3-УКЦ-2 У-УКЦ-2 В-УКЦ-2
5	Введение в объектно- ориентированное про- граммирование	22	22	-	-	46	ПКР3 (17) ДЗ2 (14)	46	
Итого:		36	36	-	-	72		75	
Экзамен								25	
3 семестр									
6	Система программирова- ния Visual Studio	18	-	-	18	72	Курсовая работа	100	3-ОПК-2 У-ОПК-2 В-ОПК-2 3-УКЦ-2 У-УКЦ-2 В-УКЦ-2
Итого:		18	-	-	18	72		100	
Зачет								100	

*Сокращение наименований форм текущего контроля и аттестации разделов:

ПКР – практическая контрольная работа, ДЗ – домашнее контрольное задание.

КТ – контрольный тест

5.2 Содержание лекционных занятий (1-й семестр) – 18 часов

Неделя	Раздел курса, № занятия	Темы лекционных занятий	Трудоемкость, час.
1	Раздел 1 Л1	Введение. Цели и методика изучения дисциплины. Обзор современных технологий разработки программного обеспечения. Обзор языков программирования высокого уровня. Этапы решения задач на ЭВМ.	2
3	Раздел 1 Л2	Алгоритм и программа. Основные управляющие алгоритмические структуры. Способы записи и требования к алгоритмам, базовые структуры. Язык программирования. Пример конструирования алгоритма методом нисходящего программирования.	2
5	Раздел 2 Л3	Основные понятия языка TPascal. Алфавит и синтаксис языка. Составные части программы. Стандартные типы данных. Описание констант и переменных стандартного типа. Стандартные функции. Арифметические и логические выражения.	2
7	Раздел 2 Л4	Линейные алгоритмы. Программирование простых задач. Оператор присваивания. Операторы ввода и вывода данных. Разветвленные алгоритмы. Управляющие конструкции языка. Условный оператор. Оператор выбора.	2
9	Раздел 2 Л5	Циклические алгоритмы. Организация циклических процессов. Оператор цикла While, Repeat, For.	2
11	Раздел 2 Л6	Массивы – как типы данных. Одномерные и двумерные массивы.	2
13	Раздел 3 Л7	Подпрограммы – процедуры.	2
15	Раздел 3 Л8	Подпрограммы – функции.	2
17	Раздел 3 Л9	Структура модулей. Основные процедуры и функции модулей GRAPH, CRT. Создание и использование модулей.	2

5.3 Темы лабораторных занятий (1-й семестр) – 36 часов

Неделя	Раздел курса, № занятия	Темы лабораторных занятий Мероприятие по текущему аудиторному контролю знаний	Трудоемкость, час.
1	Раздел 1 ЛР1	Составление алгоритмов.	2
2	Раздел 1 ЛР2	Знакомство с оболочками Turbo Pascal, Pascal ABC.	2
3	Раздел 1 ЛР3	Программы линейной структуры с использованием операторов присваивания, ввода-вывода.	2
4	Раздел 1 ЛР4	Реализация разветвленных алгоритмов.	2
5	Раздел 2 ЛР5	Использование оператора цикла While.	2
6	Раздел 2 ЛР6	Использование оператора цикла Repeat.	2
7	Раздел 2 ЛР7	Использование операторов цикла For.	2
8	Раздел 2 ЛР8	Вычисление сумм функциональных рядов.	2
9	Раздел 2 ЛР9	Программы с использованием одномерных массивов.	2
10	Раздел 2 ЛР10	Вложенные циклы.	2
11	Раздел 2 ЛР11	Программы с использованием двумерных массивов.	2
12	Раздел 3 ЛР12	Подпрограммы-процедуры.	2
13	Раздел 3 ЛР13	Подпрограммы-функции.	2
14	Раздел 3 ПКР1	Практическая контрольная работа ПКР-1 по теме «Подпрограммы».	2
15	Раздел 3 ЛР14	Обработка последовательностей символов.	2
16	Раздел 3 ЛР15	Использование ввода-вывода из файла.	2
17	Раздел 3 ЛР16	Построение графиков функций.	2
18	Раздел 3 ЛР17	Создание движущихся изображений.	2

5.4 Темы практических занятий (1-й семестр) – 18 часов

Неделя	Раздел курса, № занятия	Темы практических занятий Мероприятие по текущему аудиторному контролю знаний	Трудоемкость, час.
2	Раздел 1 ПР1	Составление алгоритмов	2
4	Раздел 1 ПР2	Составление алгоритмов	2
6	Раздел 2 ПР3	Программы линейной структуры	2
8	Раздел 2 ПР4	Операторы цикла	2
10	Раздел 2 ПР5	Программы с использованием массивов	2
12	Раздел 2 ПР6	Защита домашнего задания ДЗ1	2
14	Раздел 3 ПР7	Использование подпрограмм	2
16	Раздел 3 ПР8	Строковые переменные	2
18	Раздел 3 ПР9	Работа с графикой	2

5.5 Содержание лекционных занятий (2-й семестр) – 36 часов

Неделя	Раздел курса, № занятия	Темы лекционных занятий	Трудоемкость, час.
1	Раздел 4 Л1	Типы данных, определяемые программистом. Записи, работа с ними. Инструкция With.	2
2	Раздел 4 Л2	Порядковые типы данных: перечисляемый, тип-диапазон.	2
3	Раздел 4 Л3	Множества как тип данных.	2
4	Раздел 4 Л4	Динамические структуры данных. Переменные-указатели, динамические переменные, Динамические структуры. Динамическое распределение памяти. связанные списки.	2
5	Раздел 4 Л5	Виды списков. Примеры использования списков. Организация динамических структур данных – стек, очередь.	2

Неделя	Раздел курса, № занятия	Темы лекционных занятий	Трудоемкость, час.
6	Раздел 4 Л6	Создание консольного приложения в C#. Объявление переменных и констант простых типов в языке C#; выполнение простейших действий с переменными и константами. Операторы, позволяющие организовывать непоследовательное выполнение программного кода (условный оператор if .. else; операторы цикла for, while, do ... while; операторы выбора switch).	2
7	Раздел 4 Л7	Программирование с использованием массивов в C#. Типы и принципы работы с массивами. Работа с простыми массивами; работа с многомерными массивами.	2
8	Раздел 5 Л8	Классы. Объектное моделирование в C#. Структура и принципы объявления классов, технология создания экземпляров классов (объектов). Работа с полями данных и методами классов.	2
9	Раздел 5 Л9	Конструктор класса в C#. Перегрузка конструкторов класса. Принципы работы конструктора. Объявление конструктора класса; создание перегруженных конструкторов.	2
10	Раздел 5 Л10	Проектирование иерархии классов. Механизм организации наследования классов. Объявление производных классов; создание иерархии классов; использование механизма полиморфизма.	2
11	Раздел 5 Л11	Полиморфизм на основе интерфейсов. Принципы работы с типами интерфейсов в C#. Объявление интерфейсов в C#; создание классов, реализующих интерфейсы.	2
12	Раздел 5 Л12	Разработка приложения Windows Forms в среде MS Visual Studio и программирование алгоритмов с использованием простых элементов управления. Основные члены класса Form; Принципы использования стандартных элементов управления в приложениях Windows. использование кнопок, флажков, переключателей; списков, выпадающих списков. Обработка событий от простых элементов управления.	2
13	Раздел 5 Л13	Использование инструмента визуализации двумерной графики, предоставляемые пространством имен GDI+. Основные типы, предоставляемые библиотекой GDI+; использование различных режимов вывода графики; принципы построения графиков, диаграмм; создание интерактивных приложений; обработка событий в графических приложениях; использование класса Timer; механизм анимации в приложениях Windows.	2
14	Раздел 5 Л14	Использование в программах контекстные меню, главное меню приложения и строки состояния. Координирование главного меню приложения с контекстными меню и другими элементами управления; создание строки состояния с различными функциональными возможностями; Программирование обработчиков событий от элементов управления «панель инструментов» и «строка состояния».	2
15	Раздел 5 Л15	Создание формы верхнего уровня и модальные диалоговые окна. Создание окон различного типа (модальные и нет) в ответ на события	2

Неделя	Раздел курса, № занятия	Темы лекционных занятий	Трудоемкость, час.
		мышь, реализация обмена данными между модальными диалоговыми окнами и главной формой; принципы работы с диалоговым окном «Выбор цвета»; принципы работы с диалоговым окном «Выбор шрифта»; принципы работы с диалоговыми окнами, предназначенными для работы с каталогами; принципы работы с диалоговыми окнами, предназначенными для работы с файлами.	
16	Раздел 5 Л16	Многомодульные приложения. Проектирование консольных приложений на основе нескольких сборок. Управление несколькими разрабатываемыми проектами в одном решении; организация взаимодействия нескольких модулей приложения.	2
17	Раздел 5 Л17	Использование механизмов файлового ввода-вывода. Применение классов для работы с файлами; применение классов для работы с каталогами; использование потоков ввода-вывода.	2
18	Раздел 5 Л18	Обзорная лекция	2

5.6 Темы лабораторных занятий (2-й семестр) – 36 часов

Неделя	Раздел курса, № занятия	Темы лабораторных занятий Мероприятие по текущему аудиторному контролю знаний	Трудоемкость, час.
1	Раздел 4 ЛР1	Составление программ для обработки записей.	2
2	Раздел 4 ЛР2	Использование в программах перечисляемого типа, интервального типов данных.	2
3	Раздел 4 ЛР3	Программы работы с множествами.	2
4	Раздел 4 ПКР2	Практическая контрольная работа ПКР-2 по темам «Записи, Порядковые типы данных, Множества».	2
5	Раздел 4 ЛР4	Работа со списком типа стек, очередь.	2
6	Раздел 4 ЛР5	Консольное приложение C#. Управление потоком выполнения в программе	2
7	Раздел 4 ЛР6	Программирование с использованием массивов	2
8	Раздел 5 ЛР7	Классы. Объектное моделирование	2
9	Раздел 5 ЛР8	Конструктор класса. Перегрузка конструкторов класса.	2

Неделя	Раздел курса, № занятия	Темы лабораторных занятий Мероприятие по текущему аудиторному контролю знаний	Трудоемкость, час.
10	Раздел 5 ЛР9	Проектирование иерархии классов	2
11	Раздел 5 ЛР10	Полиморфизм на основе интерфейсов	2
12	Раздел 5 ЛР11	Разработка приложения Windows Forms. Механизм событий. Простые элементы управления	2
13	Раздел 5 ЛР12	Технология GDI+ в приложениях Windows Forms	2
14	Раздел 5 ЛР13	Меню и строка состояния в приложениях Windows Forms	2
15	Раздел 5 ЛР14	Диалоговые окна	2
16	Раздел 5 ЛР15	Многомодульные приложения	2
17	Раздел 5 ПКР3	Практическая контрольная работа ПКР-3 по темам «Разработка приложения Windows Forms».	2
18	Раздел 5 ЛР16	Основы файлового ввода-вывода	2

5.7 Содержание лекционных занятий (3-й семестр) – 18 часов

Неделя	Раздел курса, № занятия	Темы лекционных занятий	Трудоемкость, час.
1	Раздел 6 Л1	Создание Windows-приложений в среде VS.NET. Работа с последовательностями данных, размерность которых может меняться в процессе работы программы (с динамическими массивами), в C# , класс ArrayList, принадлежащий пространству имен System.Collections. Связанные списки. Алгоритмы на графах. Алгоритмы обработки символьных строк. Строковый тип данных string. Правила работы с элементом управления ListBox.	2
2	Раздел 6 Л2	Создание простого текстового редактора Демонстрация работы таких компонентов управления как MenuStrip, TextBox, OpenFileDialog, SaveFileDialog. Демонстрация принципа работы StreamReader и StreamWriter, вариант использования MessageBox. Событие формы Closing.	2
3	Раздел 6 Л3	Программное создание компонентов. Создание приложения Windows Forms, которое программно создает на форме компоненты. Создание пользовательских компонентов.	2

Неделя	Раздел курса, № занятия	Темы лекционных занятий	Трудоемкость, час.
4	Раздел 6 Л4	Разработка многопоточных приложений. Методы и средства, пространства имен и классы, применяемые при работе с потоками в среде .NET Framework. Практические навыки разработки многопоточных приложений на языке C#. Виды многозадачности: с ориентацией на процессы и с ориентацией на потоки.	2
5	Раздел 6 Л5	Разработка сетевых приложений. Основы работы компонентов, связанных с обработкой данных, полученных после использования HTTP-запросов и создать приложение, иллюстрирующее работу этих компонентов и возможности получения информации.	2
6	Раздел 6 Л6	Работа с ASP.NET в среде Visual Studio. Основные принципы разработки ASP.NET приложений. Механизмы WEB-сервисов и их использование в NET. Основные принципы работы с документами XML	2
7	Раздел 6 Л7	Основы построения приложений Windows Presentation Foundation. Проектирование интерфейса приложений на основе технологии WPF. Возможности работы с классом Window; принципы использования контейнеров компоновки; обработка событий простых элементов управления.	2
8	Раздел 6 Л8	Ресурсы, стили, триггеры. Использование инструментов XAML, позволяющие гибко настраивать и управлять внешним видом приложения.	2
9	Раздел 6 Л9	Создание MDI-приложений. Сериализация объектов. Стандартные диалоги. Особенности разработки MDI-приложений в Visual Studio .Net; способы сохранения данных в файл и загрузки из файла; механизм сериализации и десериализации объектов.	2

5.8 Этапы выполнения курсовой работы (3-й семестр) – 18 часов

Неделя	Этапы выполнения	% выполнения
1	Получение задания на КР. Обсуждение задания с преподавателем.	5
2	Получение задания на КР. Обсуждение задания с преподавателем.	5
3	Подбор и изучение литературы и Стандартов предприятия на оформление текстовой документации.	10
4	Анализ задания. Построение математической модели задачи.	20
5	Анализ задания. Построение математической модели задачи.	25

6	Выбор численного метода для решения задачи.	30
7	Построение алгоритма решения задачи для выбранного численного метода.	35
8	Построение алгоритма решения задачи для выбранного численного метода.	40
9	Кодирование разработанного алгоритма на языке.	45
10	Кодирование разработанного алгоритма на языке.	50
11	Отладка программы на компьютере.	55
12	Отладка программы на компьютере.	60
13	Тестирование программы.	65
14	Оформление пояснительной записки к курсовой работе.	70
15	Оформление пояснительной записки к курсовой работе.	80
16	Сдача пояснительной записки на проверку.	95
17-18	Защита курсовой работы.	100

5.9 Самостоятельная работа – 153 часа

Самостоятельная работа студента по учебной дисциплине регламентируется «Положением об организации самостоятельной работы студентов в НТИ НИЯУ МИФИ».

№ п.п	Тема/раздел учебной дисциплины	Вид самостоятельной работы и ее содержание	Трудоемкость, час
1	Введение в алгоритмизацию и программирование	Проработка лекционного материала Подготовка к лабораторным работам	1 1
2	Структурный подход к программированию	Проработка лекционного материала Подготовка к лабораторным работам Выполнение домашнего задания 1 (ДЗ1)	1 1 2
3	Модульное программирование	Проработка лекционного материала Подготовка к лабораторным работам Подготовка к контрольной работе 1 (КР1)	1 1 1
4	Программирование абстрактных типов данных.	Проработка лекционного материала Подготовка к лабораторным работам Подготовка к контрольной работе 2 (КР2)	7 14 5
5	Введение в объектно-ориентированное программирование	Проработка лекционного материала Подготовка к лабораторным работам Подготовка к контрольной работе 3 (КР3) Выполнение домашнего задания 2 (ДЗ2)	11 22 4 9
6	Курсовая работа	Выполнение курсовой работы	72

Перечень учебно-методического обеспечения самостоятельной работы студентов приведен в Приложении 1.

Методические указания для студентов по освоению дисциплины приведены в Приложении 2.

6 ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При реализации программы дисциплины «Программирование» используются различные образовательные технологии – во время аудиторных занятий (180 часов) занятия проводятся в форме лекций и лабораторных (практических) занятий.

В процессе изучения дисциплины на лекциях, которые проводятся в специализированной аудитории, используется мультимедийный проектор и заранее подготовленный демонстрационный материал.

В начале каждого семестра все желающие студенты обеспечиваются электронными версиями методических пособий, имеющихся на кафедре, по изучаемому курсу для работы дома.

На сервере кафедры организован каталог со всеми методическими пособиями, разработанными на кафедре, для возможности постоянного студенческого доступа к ним с любого компьютера во время всех видов занятий.

Самостоятельная работа студентов (153 часа) подразумевает проработку лекционного материала с использованием рекомендуемой литературы (методических пособий по курсу) для подготовки к лабораторным и контрольным работам, контрольным тестам и зачету, а также выполнение контрольных домашних заданий и самостоятельное изучение ряда тем. Виды самостоятельной работы и их трудоемкость подробнее описаны в п. 5.9.

Для повышения уровня знаний студентов по курсу «Программирование» в течение семестра организуются консультации преподавателей (согласно графику консультаций кафедры АУ). Во время консультационных занятий:

- проводится объяснение непонятных для студентов разделов теоретического курса;
- разъясняются алгоритмы решения задач индивидуальных домашних заданий;
- принимаются задолженности по тестовым и контрольным работам и т.д.

Перечень учебно-методического обеспечения самостоятельной работы студентов приведен в Приложении 1.

Методические указания для студентов по освоению дисциплины приведены в Приложении 2.

Реализация компетентного подхода предусматривает широкое использование в учебном процессе активных и интерактивных форм проведения занятий, предполагающих активную обратную связь между преподавателем и студентами.

В процессе изучения дисциплины «Программирование» используются интерактивные формы обучения при проведении лабораторных (практических) занятий:

- выступление студентов с докладом по теме для самостоятельного изучения;
- защита домашнего контрольного задания;
- дискуссии;
- презентации.

7 СРЕДСТВА ДЛЯ КОНТРОЛЯ И ОЦЕНКИ

В целях повышения эффективности процесса обучения студентов и стимулирования их самостоятельной работы в течение семестра используется система контроля текущей успеваемости и достижения ПР УД, включающая:

- посещение лекций;
- выполнение лабораторных работ;
- выполнение домашних контрольных работ;
- выполнение практических контрольных работ (проверка практических навыков студента);
- выполнение контрольных тестов (программированный экспресс-опрос по теоретическому материалу);
- самостоятельное изучение ряда тем.

Для оценки достижений студента используется балльно-рейтинговая система (Приложение 3).

Для целей промежуточной аттестации используется фонд оценочных средств (ФОС) по дисциплине (Приложение 4).

Результаты каждого тестового задания оцениваются в баллах, на основании которых выставляется оценка.

Задание, по которому проводится тест, считается зачтенным, если по нему набрано не менее половины от максимального количества баллов.

К экзамену в конце семестра студент допускается, если он сдал все лабораторные работы, выполнил все тестовые задания на положительные оценки, а также сдал все домашние контрольные задания.

На экзамене студенту предлагается выполнить 2 конкретных практических задания на компьютере по различным темам курса.

Итоговая экзаменационная оценка по курсу выводится с учетом балла, полученного на экзамене, и баллов, полученных по указанным выше компонентам аттестации текущей работы студента в семестре. Шкала перевода баллов в традиционную систему оценок представлена в следующей таблице:

Оценка по 5 бальной шкале	Зачет	Сумма баллов по дисциплине	Оценка (ECTS)	Градация
5 (отлично)	Зачтено	90-100	A	Отлично
4 (хорошо)		85-89	B	Очень хорошо
		75-84	C	Хорошо
		70-74	D	Удовлетворительно
3 (удовлетворительно)		65-69	E	Посредственно
	60-64			
2 (неудовлетворительно)	Не зачтено	Ниже 60	F	Неудовлетворительно

8 УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

8.1 Перечень литературы для освоения дисциплины

1. Гусева А. И. Учимся программировать : PASCAL 7.0. Задачи и методы их решения : учеб. пособие / А. И. Гусева. - 2-е изд., перераб. и доп. - М.: Диалог-МИФИ, 2012. - 216 с.
2. Долинер Л. И. Основы программирования в среде PascalABC.NET : учебное пособие / Л. И. Долинер ; [науч. ред. Г. А. Матвеева]. – Екатеринбург: Изд-во Урал. ун-та, 2014. – 128 с.
3. Комлев Н.Ю. Самоучитель игры на Паскале ABC. - М. : СОЛОН-ПРЕСС, 2015. – 256с.: ил.
4. Павловская Т. А. Паскаль. Программирование на языке высокого уровня: учеб. для вузов / Т. А. Павловская. - 2-е изд. - СПб. : Питер, 2010. - 464 с. : ил.
5. Рапаков Г. Г., Ржеуцкая С. Ю. Программирование на языке Pascal. – СПб.: БХВ-Петербург. ISBN: 5-94157-401-0, 2014. - 480 с.
6. Орлов С. А. Теория и практика языков программирования: [учеб. для бакалавров и магистров] / С. А. Орлов. - СПб. : Питер, 2013. - 688 с
7. Аляев Ю.А., Козлов О.А. Алгоритмизация и языки программирования Pascal, C++, Visual Basic, - Финансы и статистика, 2008.
8. Голицына О. Л. Основы алгоритмизации и программирования: учеб. пособие - М.: Форум : Инфра-М, 2014. - 432 с.
9. Махно В.В., Михалкович С.С., Пучкин М.В.. Основы программирования графики PascalABC. Базовые возможности. Ростов-на-Дону: Федеральное государственное образовательное учреждение высшего профессионального образования, 2007. - 49 с.
10. Павловская Т.А. C#. Программирование на языке высокого уровня. Учебник для вузов- СПб.: Питер, 2012.-432с. 2. Культин Н.Б. Microsoft Visual C# в задачах и примерах.-СПб.: БХВ-Петербург, 2009.- 320 с.
11. Нейгел Кристиан. C# 4.0 и платформа .NET 4 для профессионалов : пер. с англ. / Нейгел Кристиан, Ивьян Билл, Глинн Джей, Уотсон Карли. — М. : ООО «И.Д. Вильямс», 2011. — 1440 с.
12. Биллиг В.А. Основы программирование на C# 3.0: ядро языка[Электронный ресурс] - Национальный Открытый Университет «ИНТУИТ» 2015 г. 411с. Режим доступа: <http://www.knigafund.ru/books/178108>
13. Биллиг В.А. Объектное программирование в классах на C# 3.0 [Электронный ресурс] - Национальный Открытый Университет «ИНТУИТ» 2016 г. 397с. Режим доступа: <http://www.knigafund.ru/books/177201>
14. Язык C# и .NET Framework [Электронный ресурс] – Режим доступа: https://professorweb.ru/my/csharp/charp_theory/level1/infonet.php
15. Шилдт Г. Полный справочник по C# / Г. Шилдт. – 4-е изд. – М.: Вильямс, 2009. – 800 с.
16. Шилдт Г. Самоучитель C#/ Г. Шилдт. – 3-е изд. – СПб.: БХВ-Петербург, 2002. – 688 с.
17. Дейтел Х. C# / Х. Дейтел, П. Дейтел, Дж. Листфилд, Т.Нieto, Ш. Йегер и др. – СПб.: БХВ-Петербург, 2006. – 1056 с.
18. Агупов П.В. C#. Разработка компонентов в MS Visual Studio 2005/2008 / П.В. Агупов - СПб.: БХВ-Петербург, 2008. – 480 с.
19. Бишоп Дж. C# в кратком изложении / Дж. Бишоп, Н. Хорспул. - М.: Бином, 2005. – 472 с.

8.2 Учебно-методическое обеспечение дисциплины

1. Николаев Н.А. Введение в программирование на языке Pascal ABC. Методическое пособие. Новоуральск, Новоуральск, НГТИ, 2019. - 73 с.
2. Николаев Н.А. Сборник заданий по программированию. Часть 1. Методические указания. Новоуральск, НГТИ, 2009, - 52 с.
3. Николаев Н.А., Шивырев А.А. Сборник заданий по программированию. Часть 2. Методическое пособие. Новоуральск, НГТИ, 2009, - 48 с.

4. Николаев Н.А. Оформление исходного текста программ. Методические указания. Новоуральск, НГТИ, 2006, - 44 с.
5. Николаев Н.А. Работа с графикой в системе Turbo Pascal. Методическое пособие. Новоуральск, НГТИ, 2006, - 48 с.
6. Николаев Н.А. Основы работы с функциями Windows-интерфейса прикладной программы. Методические указания. Новоуральск, НГТИ, 2007, - 20 с.
7. Николаев Н.А. Система типов данных языка TurboPascal. Методическое пособие. Новоуральск, НГТИ, 2008, - 43 с.
8. Николаев Н.А. Программирование в Delphi Методическое пособие. Новоуральск, НГТИ, 2014, - 60 с.
9. Николаев Н.А. Конспект лекций по курсу «Программирование на языке высокого уровня». Часть 1. Методическое пособие. Новоуральск, НГТИ, 2009, - 74 с.
10. Николаев Н.А. Конспект лекций по курсу «Программирование на языке высокого уровня». Часть 2. Методическое пособие. Новоуральск, НГТИ, 2009, - 62 с.
11. Николаев Н.А. Продвинутое возможности Turbo Pascal: Методическое пособие по курсу «Программирование на языках высокого уровня» для студентов специальности 230102 очной формы обучения. Новоуральск, НГТИ, 2009, 62 с.
12. Николаев Н.А. Введение в объектно-ориентированное программирование. Методическое пособие по курсу «Программирование на языке высокого уровня» для студентов специальности 230102 очной формы обучения. - Новоуральск, НГТИ, 2009, - 36 с.

8.3 Перечень ресурсов информационно-коммуникационной сети «Интернет», необходимых для освоения дисциплины

Наименование ресурса	Электронный адрес ресурса
2) ЭБС «Лань»	https://e.lanbook.com
4) Образовательная платформа Юрайт	https://urait.ru/bcode/468952
5) Образовательный портал НИЯУ МИФИ	https://online.mephi.ru/
6) Научная библиотека НИЯУ МИФИ	http://library.mephi.ru/
7) Лекторий Teach-in	https://teach-in.ru/
8) Научная библиотека избранных естественно-научных изданий	https://scask.ru/
Интернет-портал образовательных ресурсов по ИТ	http://www.intuit.ru
ЭБС «Университетская библиотека online»	www.biblioclub.ru

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Для осуществления образовательного процесса по дисциплине необходимо:

1 Лекционные занятия:

- аудитория, оборудованная техническими средствами для демонстрации лекций-визуализаций (проектор, экран, компьютер/ноутбук);
- комплект электронных презентаций/слайдов;

2 Лабораторные занятия:

- компьютерный класс;
- среда программирования PascalABC;
- среда программирования Visual Studio;

НТИ НИЯУ МИФИ располагает данными средствами в полном объеме.

Учебная дисциплина обеспечена учебно-методической документацией и материалами. Ее содержание представлено в локальной сети института и находится в режиме свободного доступа для студентов. Доступ студентов для тренинга по прохождению тестовых заданий и для самостоятельной подготовки осуществляется через компьютеры дисплейного класса (в стандартной комплектации).

В библиотечном фонде представлены необходимые учебные пособия согласно нормативам ФГОС.

Все рекомендуемые методические пособия и материалы по курсу «Программирование», разработанные преподавателями кафедры, имеются в электронном виде, на бумажных носителях, представлены в УМКД. Пособия хранятся на кафедре Автоматизация управления, представлены в электронном читальном зале НТИ НИЯУ МИФИ. Электронные копии пособий также могут индивидуально предоставляться студентам по их запросу на кафедре Автоматизация управления.

Студенты своевременно обеспечиваются индивидуальными вариантами домашних заданий. Варианты заданий имеются в электронном виде и представлены в УМКД (кафедра Автоматизация управления).

Лабораторные работы по курсу осуществляются в компьютерных классах. Задания для выполнения на лабораторных работах представлены в методических пособиях кафедры.

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ

к рабочей программе по курсу
«Программирование»
для ООП ВПО 09.03.01

на 20____/20____ уч.год

В рабочую программу вносятся следующие изменения:

Рабочая программа пересмотрена и одобрена на заседании кафедры «__»____20__ г.
Заведующий кафедрой АУ

на 20____/20____ уч.год

В рабочую программу вносятся следующие изменения:

Рабочая программа пересмотрена и одобрена на заседании кафедры «__»____20__ г.
Заведующий кафедрой АУ

на 20____/20____ уч.год

В рабочую программу вносятся следующие изменения:

Рабочая программа пересмотрена и одобрена на заседании кафедры «__»____20__ г.
Заведующий кафедрой АУ

Программа действительна

на 20____/20____ уч.год _____ (заведующий кафедрой АУ)

на 20____/20____ уч.год _____ (заведующий кафедрой АУ)

на 20____/20____ уч.год _____ (заведующий кафедрой АУ)

на 20____/20____ уч.год _____ (заведующий кафедрой АУ)

на 20____/20____ уч.год _____ (заведующий кафедрой АУ)

ПРИЛОЖЕНИЕ 1. ПЕРЕЧЕНЬ УЧЕБНО-МЕТОДИЧЕСКОГО ОБЕСПЕЧЕНИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ.

№	Литература	Год	Курс	Номер группы	Семестр	Кол-во студентов	Кол-во книг	Коэффициент книго-обеспеченности
Основная литература								
1	Финогенов К.Г. Основы объектно-ориентированного программирования : [лабораторный практикум]. Электронный ресурс. Москва : МИФИ. 2008 – точка доступа – ЭБС НИЯУ МИФИ – mehpri.ru	2008	1	ИТ-12	1	7	10	1,0
2	Давыдова Н. А. Программирование : учебное пособие / Н. А. Давыдова, Е. В. Боровская. - 4-е изд. - Москва : Лаборатория знаний, 2020. - 241	2020	1	ИТ-12	1	7	12	1,0
3	Тузовский А.Ф. Объектно-ориентированное программирование. М.: Юрайт, 2018.-206 с.	2018	1	ИТ-12	1	7	10	1,0
4	Павловская Т. А. С#. Программирование на языке высокого уровня: [учебник для вузов] / Т. А. Павловская. -Санкт-Петербург : Питер, 2014. - 432 с : ил	2014	1	ИТ-12	1	7	12	1,0
Дополнительная литература								
1	Финогенов К.Г. Программирование в системе Windows с помощью объектно-ориентированный библиотек [лабораторный практикум]. Электронный ресурс. Москва : МИФИ. 2008 – точка доступа – ЭБС НИЯУ МИФИ – mehpri.ru	2008	1	ИТ-12	1	7	5	0,71

ПРИЛОЖЕНИЕ 2. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ СТУДЕНТОВ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ.

- стандарт организации СТО НТИ-2-2014. Требования к оформлению текстовой документации;
- методические рекомендации по организации самостоятельной работы обучающихся НТИ НИЯУ МИФИ.

ПРИЛОЖЕНИЕ 3. БАЛЛЬНО-РЕЙТИНГОВАЯ СИСТЕМА ОЦЕНКИ.

Таблица 3.1. Распределение баллов текущего рейтинга по видам деятельности студента направления подготовки 09.03.01 при изучении курса "Программирование" (I семестр)

№ п-п	Вид деятельности	Кол-во	Стоимость (в баллах)		Количество баллов	
			max	min	max	min
1.	Посещение лекций	9	1	-	9	5
2.	Выполнение лабораторных работ по расписанию	17	2	-	34	-
3.	Выполнение лабораторных работ вне расписания (без уважительных причин)	17	-	1	-	17
4.	Выполнение практических работ по расписанию	9	1		9	
5.	Выполнение практических работ вне расписания (без уважительных причин)	9		1		9
6.	Практические контрольные работы	1	5	3	5	3
7.	Выполнение домашних заданий.	1	5	2	5	2
8.	Контрольное тестирование (автоматизированное) по темам (модулям) курса:					
	- КТ1 – Основы алгоритмизации	1	4	2	4	2
	- КТ2 – общие вопросы программирования	1	4	2	4	2
Итого					70	40
9.	Экзамен	1	30		30	20
Итого					100	60

Таблица 3.2. Распределение баллов текущего рейтинга по видам деятельности студента направления подготовки 09.03.01 при изучении курса "Программирование" (II семестр)

№ п-п	Вид деятельности	Кол-во	Стоимость (в баллах)		Количество баллов	
			max	min	max	min
1.	Посещение лекций	18	1	-	18	9
2.	Выполнение лабораторных работ по расписанию	16	2	-	32	-
3.	Выполнение лабораторных работ вне расписания (без уважительных причин)	16	-	1	-	16
4.	Практические контрольные работы	2	10	6	20	12
5.	Выполнение домашних заданий.	1	5	3	5	3
Итого					75	40
6.	Экзамен	1	25		25	20
Итого					100	60

Таблица 3.3. Распределение баллов текущего рейтинга по разделам при изучении курса "Программирование" студентами направления подготовки 09.03.01

№ п/п	Раздел учебной дисциплины	Виды учебной деятельности, включая самостоятельную ра- боту студентов и трудоем- кость (в часах)				Аттестация раздела (форма*, неделя)	Максималь- ный балл за раздел
		Лекции	Лабораторные занятия	Практические занятия	Самостоятель- ная работа		
1 семестр							
1	Введение в алгоритмиза- цию и программирование	4	8	4	2	КТ1 (5)	16
2	Структурный подход к программированию	8	14	8	4	ДЗ1 (12)	27
3	Модульное программиро- вание	6	14	6	3	ПКР1 (14) КТ2 (17)	27
Экзамен							30
Итог							100
2 семестр							
4	Программирование аб- страктных типов данных	14	14	-	26	ПКР2 (4)	29
5	Введение в объектно- ориентированное про- граммирование	22	22	-	46	ПКР3 (17) ДЗ2 (14)	46
Экзамен							25
Итог							100
6	Система программирова- ния Visual Studio	18	-	-	72	Курсовая работа	100
Зачет							100

*Сокращение наименований форм текущего контроля и аттестации разделов:

ДЗ – домашнее контрольное задание, КТ – контрольный тест, ПКР – практическая контрольная работа.

ПРИЛОЖЕНИЕ 4. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий рубежного и промежуточного контроля по дисциплине.

Связь между формируемыми компетенциями и формами контроля их освоения представлена в таблице:

№ п.п.	Контролируемые модули, разделы (темы) дисциплины	Результаты освоения ООП		Виды аттестации		Наименование оценочного средства
		Код контролируемой компетенции	Индикаторы освоения компетенции	Текущий контроль	Промежуточная аттестация	
1	Введение в алгоритмизацию и программирование	ОПК-2 УКЦ-2	3-ОПК-2 У-ОПК-2 В-ОПК-2 3-УКЦ-2 У-УКЦ-2 В-УКЦ-2	КТ1	По итогам текущего контроля	Компьютерное тестирование 1.
2	Структурный подход к программированию			ДЗ1	По итогам текущего контроля	Домашнее задание 1.
3	Модульное программирование			ПКР1 КТ2	По итогам текущего контроля	Практическая контрольная работа 1 Компьютерное тестирование 2
4	Программирование абстрактных типов данных			ПКР2	По итогам текущего контроля	Практическая контрольная работа 2,
5	Введение в объектно-ориентированное программирование			ПКР3 ДЗ2	По итогам текущего контроля	Практическая контрольная работа 3, Домашнее задание 2
6	Система программирования Visual Studio			Курс. работа	По итогам текущего контроля	Курсовая работа

Варианты практических контрольных работ

ПКР - 1 Типовые задачи практической контрольной работы 1 (I семестр, 14 неделя)

Вариант 1

1. Описать процедуру $\text{RectPS}(x_1, y_1, x_2, y_2, P, S)$, вычисляющую периметр P и площадь S прямоугольника со сторонами, параллельными осям координат, по координатам (x_1, y_1) , (x_2, y_2) его противоположных вершин (x_1, y_1, x_2, y_2 — входные, P и S — выходные параметры вещественного типа). С помощью этой процедуры найти периметры и площади трех прямоугольников с данными противоположными вершинами.

2. Описать функцию $\text{Fact2}(N)$ вещественного типа, вычисляющую двойной факториал:

$N!! = 1 \cdot 3 \cdot 5 \cdot \dots \cdot N$, если N — нечетное;

$N!! = 2 \cdot 4 \cdot 6 \cdot \dots \cdot N$, если N — четное

($N > 0$ — параметр целого типа; вещественное возвращаемое значение используется для того, чтобы избежать целочисленного переполнения при больших значениях N). С помощью этой функции найти двойные факториалы пяти данных целых чисел.

Вариант 2

1. Описать процедуру $\text{DigitCountSum}(K, C, S)$, находящую количество C цифр целого положительного числа K , а также их сумму S (K — входной, C и S — выходные параметры целого типа). С помощью этой процедуры найти количество и сумму цифр для каждого из пяти данных целых чисел.

2. Описать функцию $\text{DigitN}(K, N)$ целого типа, возвращающую N -ю цифру целого положительного числа K (цифры в числе нумеруются справа налево). Если количество цифр в числе K меньше N , то функция возвращает -1 . Для каждого из пяти данных целых положительных чисел $K_1, K_2, \dots, K_5$ вызвать функцию DigitN с параметром N , изменяющимся от 1 до 5.

Вариант 3

1. Описать процедуру $\text{InvertDigits}(K)$, меняющую порядок следования цифр целого положительного числа K на обратный (K — параметр целого типа, являющийся одновременно входным и выходным). С помощью этой процедуры поменять порядок следования цифр на обратный для каждого из пяти данных целых чисел.

2. Описать функцию $\text{RingS}(R_1, R_2)$ вещественного типа, находящую площадь кольца, заключенного между двумя окружностями с общим центром и радиусами R_1 и R_2 (R_1 и R_2 — вещественные, $R_1 > R_2$). С ее помощью найти площади трех колец, для которых даны внешние и внутренние радиусы. Воспользоваться формулой площади круга радиуса R : $S = \pi \cdot R^2$.

Вариант 4

1. Описать процедуру $\text{AddRightDigit}(D, K)$, добавляющую к целому положительному числу K справа цифру D (D — входной параметр целого типа, лежащий в диапазоне 0–9, K — параметр целого типа, являющийся одновременно входным и выходным). С помощью этой процедуры последовательно добавить к данному числу K справа данные цифры D_1 и D_2 , выводя результат каждого добавления.

2. Описать функцию $\text{SumRange}(A, B)$ целого типа, находящую сумму всех целых чисел от A до B включительно (A и B — целые). Если $A > B$, то функция возвращает 0. С помощью этой функции найти суммы чисел от A до B и от B до C , если даны числа A, B, C .

Вариант 5

1. Описать процедуру $\text{AddLeftDigit}(D, K)$, добавляющую к целому положительному числу K слева цифру D (D — входной параметр целого типа, лежащий в диапазоне 1–9, K — параметр целого типа, являющийся одновременно входным и выходным). С помощью этой процедуры последовательно добавить к данному числу K слева данные цифры D_1 и D_2 , выводя результат каждого добавления.

2. Описать функцию $\text{IsSquare}(K)$ логического типа, возвращающую True , если целый параметр K (> 0) является квадратом некоторого целого числа, и False в противном случае. С ее помощью найти количество квадратов в наборе из 10 целых положительных чисел.

Вариант 6

1. Описать процедуру $\text{Swap}(X, Y)$, меняющую содержимое переменных X и Y (X и Y — вещественные параметры, являющиеся одновременно входными и выходными). С ее помощью для данных переменных A, B, C, D последовательно поменять содержимое следующих пар: A и B , C и D , B и C и вывести новые значения A, B, C, D .

2. Описать функцию $\text{DigitCount}(K)$ целого типа, находящую количество цифр целого положительного числа K . Используя эту функцию, найти количество цифр для каждого из пяти данных целых положительных чисел.

Вариант 7

1. Описать процедуру $\text{Minmax}(X, Y)$, записывающую в переменную X минимальное из значений X и Y , а в переменную Y — максимальное из этих значений (X и Y — вещественные параметры, являющиеся одновременно входными и выходными). Используя четыре вызова этой процедуры, найти минимальное и максимальное из данных чисел A, B, C, D .

2. Описать функцию $\text{IsPower5}(K)$ логического типа, возвращающую True , если целый параметр K (> 0) является степенью числа 5, и False в противном случае. С ее помощью найти количество степеней числа 5 в наборе из 10 целых положительных чисел.

Вариант 8

1. Описать процедуру $\text{SortInc3}(A, B, C)$, меняющую содержимое переменных A, B, C таким образом, чтобы их значения оказались упорядоченными по возрастанию (A, B, C — вещественные параметры, являющиеся одновременно входными и выходными). С помощью этой процедуры упорядочить по возрастанию два данных набора из трех чисел: (A_1, B_1, C_1) и (A_2, B_2, C_2) .

2. Описать функцию $\text{Calc}(A, B, Op)$ вещественного типа, выполняющую над ненулевыми вещественными числами A и B одну из арифметических операций и возвращающую ее результат. Вид операции определяется целым параметром Op : 1 — вычитание, 2 — умножение, 3 — деление, остальные значения — сложение. С помощью Calc выполнить для данных A и B операции, определяемые данными целыми N_1, N_2, N_3 .

Вариант 9

1. Описать процедуру $\text{SortDec3}(A, B, C)$, меняющую содержимое переменных A, B, C таким образом, чтобы их значения оказались упорядоченными по убыванию (A, B, C — вещественные параметры, являющиеся одновременно входными и выходными). С помощью этой процедуры упорядочить по убыванию два данных набора из трех чисел: (A_1, B_1, C_1) и (A_2, B_2, C_2) .

2. Описать функцию $\text{IsPowerN}(K, N)$ логического типа, возвращающую True , если целый параметр K (> 0) является степенью числа N (> 1), и False в противном случае. Дано число N (> 1) и набор из 10 целых положительных чисел. С помощью функции IsPowerN найти количество степеней числа N в данном наборе.

Вариант 10

1. Описать процедуру $\text{ShiftRight3}(A, B, C)$, выполняющую *правый циклический сдвиг*: значение A переходит в B , значение B — в C , значение C — в A (A, B, C — вещественные параметры, являющиеся одновременно входными и выходными). С помощью этой процедуры выполнить правый циклический сдвиг для двух данных наборов из трех чисел: (A_1, B_1, C_1) и (A_2, B_2, C_2) .

2. Описать функцию $\text{Quarter}(x, y)$ целого типа, определяющую номер координатной четверти, в которой находится точка с ненулевыми вещественными координатами (x, y) . С помощью этой функции найти номера координатных четвертей для трех точек с данными ненулевыми координатами.

ПКР - 2 Типовые задачи практической контрольной работы 2 (III семестр, 4 неделя)

Вариант 1

1. Используя следующий фрагмент программы, опишите перечисленные ниже функции и напишите программу, их использующую:

type

Name = (Anja, Valja, Jenja, Petja, Sasha, Tanja, Shura, Jura);

```

Data = record
    pol: (man, female);
    length: 140..200
end;
gruppa = array [Name] of Data;

```

- а) функция SrednLength(Gr) определяет средний рост женщин из группы GR;
- б) функция MaxMan(Gr) определяет имя самого высокого мужчины из группы Gr;
- с) логическая функция OneLength(Gr) проверяет, есть ли в группе Gr хотя бы два человека одинакового роста.

Данные для программы ввести из файла.

2. Пусть дан фрагмент программы:

```

const n = 10;
type
    number = 1..n;
    matrix = array [ number, number ] of real;
    num = set of number;

```

Опишите функцию Sum(A, S1, S2), вычисляющую сумму тех элементов матрицы A, номера строк и номера столбцов которых принадлежат соответственно непустым множествам S1 и S2 типа num.

Вариант 2

1. Используя следующий фрагмент программы, опишите перечисленные ниже процедуры, функции и напишите программу, их использующую:

```

type
    Data = record
        Day : 1..31;
        Month : 1..12;
        Year : 1901..1997
    end;
    Anketa = record
        Fam : string[12];
        pol: (man, female);
        Birthday : Data
    end;
    Gruppa = array[1..25] of Anketa;

```

- а) процедура Last(Gr, Famil) присваивает строке Famil фамилию самого старшего мужчины из группы Gr (считать, что такой есть и он единственный);
 - б) процедура Vivod (Gr, Bukva) выводит все фамилии людей из группы Gr, начинающиеся с литеры Bukva, и даты рождения этих людей.
 - с) функция Feb29(Gr) определяет есть ли в группе женщины, родившиеся 29 февраля.
- Данные для программы ввести из файла

2. Пусть дан фрагмент программы:

```

type
    tovar = ( hleb, maslo, mjaso, riba. sol, sir, kolbasa, sahar, kofe );

```

assortment = set of tovar;

shop {магазин} = array [1..20] of assortment;

Опишите процедуру Availability(sh, A, B, C), которая по информации из массива sh типа shop присваивает параметрам A, B, C типа assortment значения, перечисленные ниже.

- a) A — продукты, которые есть во всех магазинах;
- b) B — продукты, которые есть хотя бы в одном магазине;
- c) C — продукты, которых нет ни в одном магазине.

ПКР - 3 Типовые задачи практической контрольной работы 3 (II семестр, 17 неделя)

Вариант 1

Сформировать матрицу случайных чисел в диапазоне [0; M] размера N×N.

1. Заменить элементы на главной диагонали на 1.
2. Вычислить сумму элементов, расположенных выше главной диагонали.

Разместить компоненты на форме (см. рисунок). Обеспечить выполнение действий: ОБНОВИТЬ, ЗАМЕНИТЬ, ВЫЧИСЛИТЬ с помощью кнопок на форме и с помощью меню.

The interface for Variant 1 includes input fields for N (value 3) and M (value 6). Below these is a 4x4 matrix of random numbers. To the right are three buttons: 'ОБНОВИТЬ', 'ЗАМЕНИТЬ', and 'ВЫЧИСЛИТЬ'. Below the matrix is a label 'СУММА' and an empty rectangular box for the result.

3.1	4.7	1.8	
2.6	4.9	3.1	
1.7	0.1	2.0	

Вариант 2

Подсчитать количество элементов массива, которые больше всех своих предшественников. Элементы массива получить с помощью генератора случайных чисел в диапазоне [0; M]. Предусмотреть возможность изменения любых элементов с клавиатуры. Осуществить контроль правильности величины M при вводе. Обеспечить интерфейс, показанный ниже. Дать возможность выхода из программы с помощью меню.

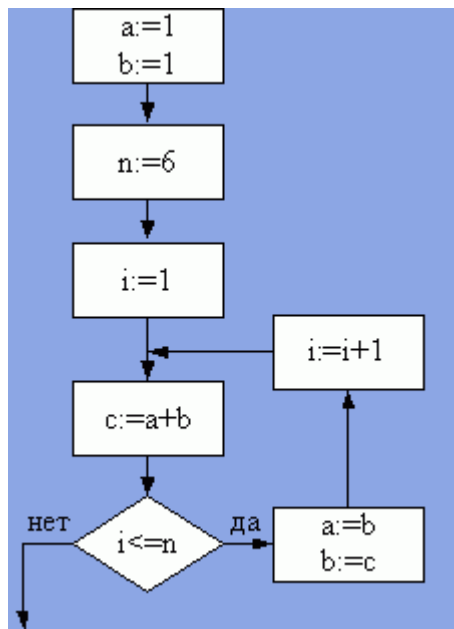
The interface for Variant 2 includes input fields for N (value 4) and M (value 16). Below these is a horizontal array labeled 'Массив' containing 10 numbers: 1, 8, 2, 13, followed by six empty boxes. Below the array is a label 'ОТВЕТ' and a text box containing '2 элемента'. At the bottom are two buttons: 'ОБНОВИТЬ' and 'ВЫЧИСЛИТЬ'.

Массив: 1 | 8 | 2 | 13 | | | | | |

ОТВЕТ: 2 элемента

Вариант 3

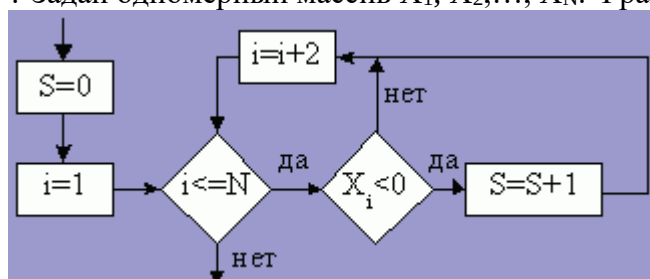
Написать программу, проверяющую, является ли строка S допустимым идентификатором (используются только латинские заглавные буквы и цифры, первый символ – буква, максимальная длина N символов). При утвердительном ответе возвращается 0, если строка начинается с цифры, то возвращается -1, если длина больше допустимой -2, если S содержит недопустимые символы, то возвращается номер первого недопустимого символа. Исходные данные: N, S. С помощью меню выполнять очистку исходной строки и выполнение проверки.



Тогда значение переменной С равно ...

- 13
- 21
- =34
- 32

? Задан одномерный массив $X_1, X_2, \dots, X_N$. Фрагмент алгоритма



определяет

- количество положительных элементов с четными номерами
- сумму отрицательных элементов с нечетными номерами
- индекс первого отрицательного элемента с нечетным номером
- количество отрицательных элементов с четными номерами
- =количество отрицательных элементов с нечетными номерами

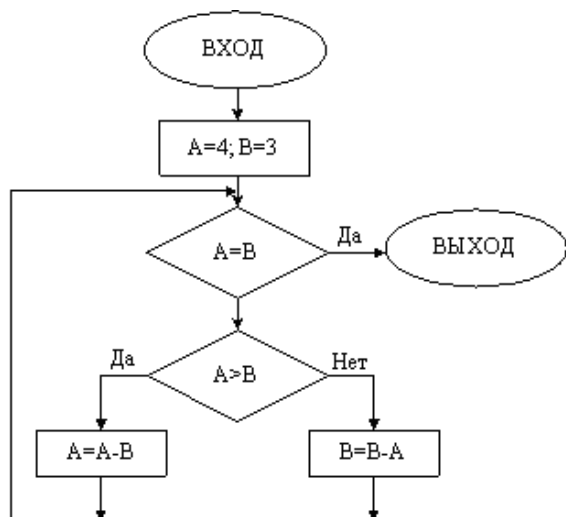
? В результате выполнения фрагмента алгоритма

1. $s=1; n=1$
2. НЦ для $i=2$ до 5 (начало цикла)
3. $n=n+1; s=s+i$
4. КЦ (конец цикла)

Переменные n, s примут значения

- $n=5, s=5$
- $n=5, s=14$
- $n=4, s=15$
- $n=1, s=5$
- = $n=5, s=15$

? Дана блок-схема алгоритма



На выходе **A** и **B** примут следующие значения:

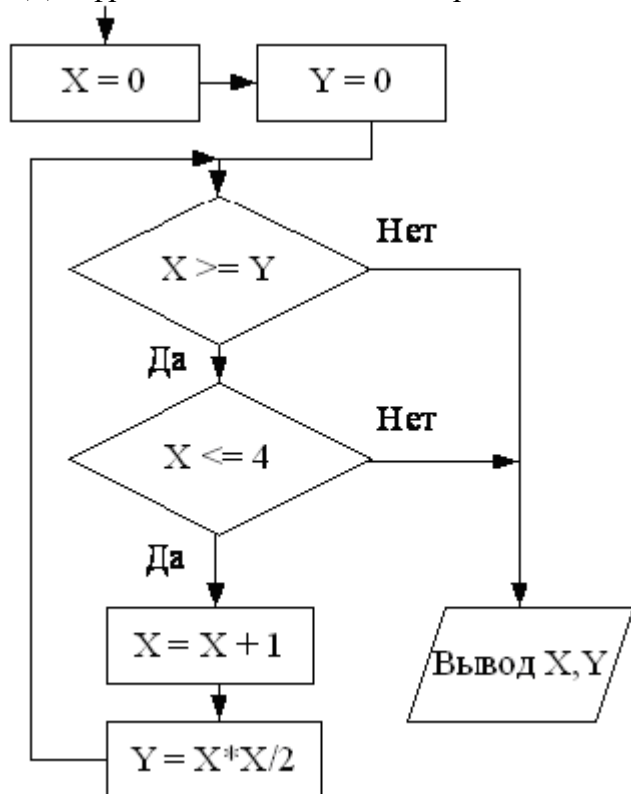
-A=1, B=1

-A=3, B=3

-A=4, B=3

-A=0, B=0

? Дан фрагмент блок-схемы алгоритма:



На выходе **X** и **Y** будут принимать следующие значения ...

-X=4, Y=1

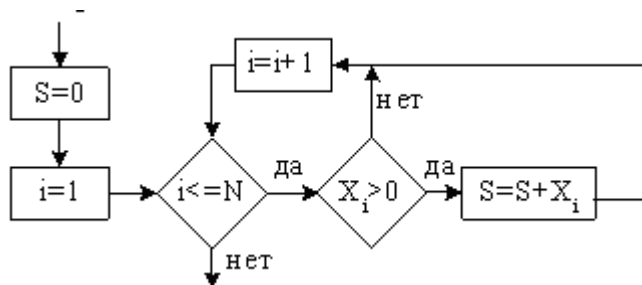
-X=2, Y=3.5

=X=3, Y=4.5

-X=3, Y=0.5

-X=2, Y=2

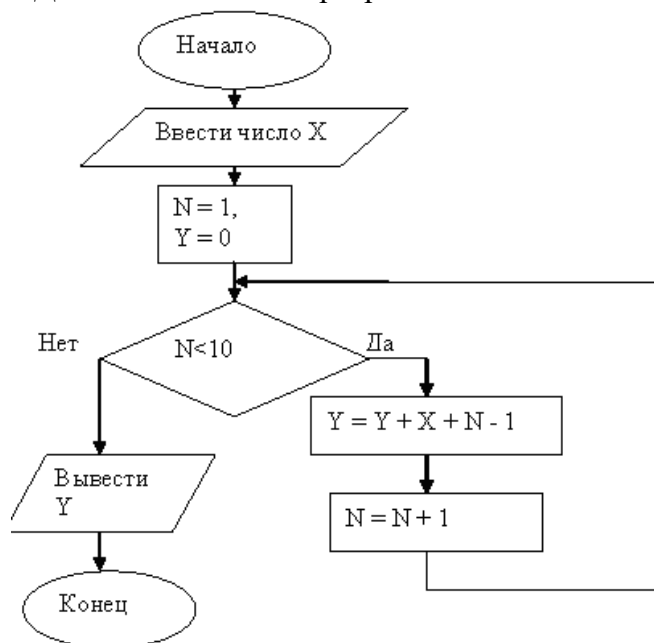
? Задан одномерный массив $X_1, X_2, \dots, X_N$. Фрагмент алгоритма



определяет

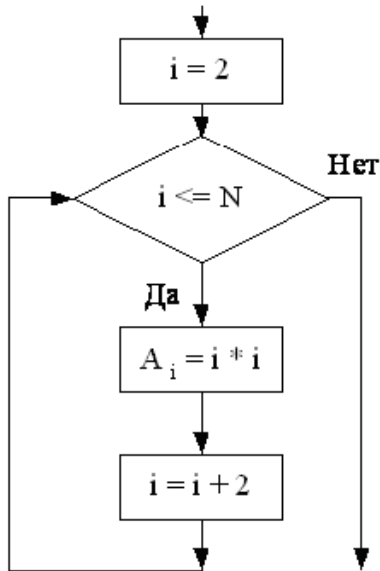
- сумму отрицательных элементов
- индекс последнего положительного элемента
- =сумму положительных элементов
- количество положительных элементов
- максимальный элемент массива

? Данная блок-схема программы



- возводит число X в 9-ю степень и выводит результат
- возводит число X в 10-ю степень и выводит результат
- производит сложение 10 подряд идущих натуральных чисел, начиная с X , и выводит результат
- =производит сложение 9 подряд идущих натуральных чисел, начиная с X , и выводит результат

? В результате работы фрагмента алгоритма:



элементы массива A_2, A_4, A_6, A_8 при $N = 8$ получают, соответственно, значения ...

- 2, 4, 16, 32
- 4, 12, 24, 36
- =4, 16, 36, 64
- 1, 9, 25, 49
- 4, 16, 32, 48

#2 ОБЩИЕ ВОПРОСЫ ПРОГРАММИРОВАНИЯ

? Ни в одном языке программирования нет выражений

- =физических
- текстовых
- арифметических
- логических

? Наиболее точным определением понятия «программа» является ...

- законченное минимальное смысловое выражение на языке программирования
- =алгоритм, записанный на языке программирования
- понятное и точное предписание человеку совершить последовательность действий, направленных на достижение поставленной цели
- набор команд операционной системы компьютера
- протокол взаимодействия компонентов компьютерной сети

? Наиболее точным определением понятия «массив» является ...

- законченное минимальное смысловое выражение на языке программирования
- =последовательность фиксированного числа однотипных переменных, имеющих общее имя
- набор переменных, начинающихся с одной буквы
- самый простой оператор языка программирования
- ограниченная апострофами последовательность любых символов

? Наиболее точным определением понятия «переменная» является ...

- =именованная область памяти, в которой хранится некоторое значение
- служебное слово на языке программирования
- любое законченное минимальное смысловое выражение на языке программирования
- порядковый номер элемента в массиве
- описание действий, которые должна выполнить программа

? Процесс поиска ошибок в программе принято называть ...

- испытанием

- =отладкой
- профилактикой
- ремонтom
- диагностикой

? Операторы присваивания в языках программирования

- =задают значение переменных
- соотносят переменным некоторые множества допустимых значений
- организуют выполнение повторяемых действий
- меняют значения констант
- вычисляют значения математических выражений

? Переменная – это

- описание действий, которые должна выполнять программа
- законченное минимальное смысловое выражение на языке программирования
- =область памяти, в которой хранится некоторое значение
- порядковый номер элемента в массиве
- служебное слово на языке программирования

? Обнаруженное при тестировании нарушение формы записи программы приводит к сообщению об ошибке

- локальной
- семантической
- орфографической
- грамматической
- стилистической
- =синтаксической

? Если устранены все синтаксические ошибки и ошибки времени выполнения, то может ли программа вычислять не то, что требуется?

- =может, из-за ошибок в логике
- может, из-за неправильно расставленных END
- может, из-за неправильно расставленных ELSE
- может, из-за неправильно введенных данных
- не может

? Что может выводить на экран компилятор?

- =сообщение о синтаксической ошибке
- сообщение о логической ошибке
- сообщение об ошибке времени выполнения
- результат пошаговой трассировки

? Что может изменяться при выполнении оператора присваивания?

- алгоритм
- имя переменной
- тип переменной
- =значение переменной
- =значение типизированной константы

? Этап разработки программ, состоящий в формировании исходного текста программы на одном из языков программирования в соответствии с заданным алгоритмом, получил название ...

- =этап кодирования
- этап реализации
- этап моделирования

- этап системного анализа
- исходный этап

? Интегрированная система программирования включает компонент для набора исходного текста программы (исходного кода), который называется ..

- отладчик
- конструктор
- редактор связей
- =текстовый редактор
- библиотека

? Набор операторов, выполняющих заданное действие и не зависящих от других частей исходного кода, называют ...

- телом программы
- вложенной программой
- разделом программы
- параметрами программы
- =подпрограммой

? В объектно-ориентированном программировании способность объекта сохранять свойства и методы класса-родителя называют ...

- полиморфизмом
- встраиванием
- инкапсуляцией
- резидентностью
- =наследованием

? Утверждение – «Программа разбивается на множество подпрограмм, комбинирование которых и формирует итоговый алгоритм решения задачи» - отражает идею _____ программирования

- модульного
- алгоритмического
- объектно-ориентированного
- =структурного
- событийно-ориентированного

? Методика анализа, проектирования и написания приложений с помощью структуры классов, каждый из которых является целостным фрагментом кода и обладает свойствами и методами, называется _____ программированием

- структурным
- алгоритмическим
- формальным
- =объектно-ориентированным
- модульным

? В объектно-ориентированном программировании каждый объект по отношению к своему классу является

- кодом
- свойством
- =экземпляром
- фрагментом
- функцией

? Компонент интегрированной системы программирования, предназначенный для перевода исходного текста программы в машинный код, называется ...

- =транслятором
- построителем кода
- редактором связей
- переводчиком

? Параметры, указываемые в момент вызова подпрограммы из основной программы, называются

- ...
- глобальными
- =фактическими
- абсолютными
- постоянными

? Характерным признаком линейной программы является ...

- использование в ней исключительно операторов присваивания
- =строгое последовательное выполнение операторов в порядке их записи
- наличие в каждой программной строке только одного оператора
- присутствие в ней операторов безусловного перехода

? Динамическими структурами данных являются ...

- =дерево
- =очередь
- запись
- массив

? Программа – это ...

- протокол взаимодействия компонентов компьютерной сети
- набор команд операционной системы компьютера
- законченное минимальное смысловое выражение на языке программирования
- =алгоритм, записанный на языке программирования

? Подпрограмму, вычисляющую некоторое значение и передающую его обратно в главную программу, называют ...

- аргументом
- объектом
- скриптом
- =функцией

? Использование подпрограмм в программе позволяет организовать выполнение ...

- разветвления вычислений в соответствии с алгоритмом
- =повторяющихся действий над различными исходными данными
- циклических вычислений в соответствии с алгоритмом
- ввода данных в программу

? Многократное исполнение одного и того же участка программы называется ...

- итерацией
- рекурсией
- обращением к подпрограмме
- =циклическим процессом

? Параметры, указываемые в заголовке подпрограммы, называются ...

- фактическими
- глобальными
- абсолютными
- =формальными

? Что такое формальные параметры подпрограмм:

- неопределенные входные данные
- входные данные определенного типа
- =список параметров, перечисленных в заголовке подпрограммы
- конкретные значения входных данных

? Языки программирования высокого уровня характеризует:

- =близость к естественному языку
- близость к машинному языку
- наличие понятия типа данных
- зависимость от архитектуры конкретного компьютера

? Какие типы конструкций не должны применяться для реализации логики алгоритма и программы, с точки зрения структурного программирования:

- =безусловные переходы
- повторение (циклы)
- ветвление
- последовательное выполнение

? Не является принципом объектно-ориентированного программирования ...

- наследование
- инкапсуляция
- =использование функций
- полиморфизм

? Структурное программирование по-другому называют:

- программирование без WHILE
- программирование без BEGIN
- программирование без PRINT
- =программирование без GOTO

? Какие из перечисленных языков наиболее подходят для системного программирования:

- =Ассемблер
- =C, C++
- PHP
- Java

? Что означает положительный результат при отладке программных систем:

- =ошибки исправлены
- есть замечания
- ошибки найдены
- ошибки не найдены

? Как называется деятельность, направленная на обнаружение и исправление ошибок в программной системе:

- демонстрация
- =отладка
- рефакторинг
- тестирование

Задания для домашних работ

Д31 Варианты домашнего задания (I семестр, 12 неделя)

Matrix1. Дана матрица и целое число $K \in M$). Найти сумму и произведение элементов K -й строки данной матрицы.

Matrix2. Дана матрица и целое число $K \in N$). Найти сумму и произведение элементов K -го столбца данной матрицы.

Matrix3. Дана матрица Для каждой строки матрицы найти сумму ее элементов.

Matrix4. Дана матрица Для каждого столбца матрицы найти произведение его элементов.

Matrix5. Дана матрица Для каждой строки матрицы с нечетным номером (1, 3, ...) найти среднее арифметическое ее элементов. Условный оператор не использовать.

Matrix6. Дана матрица Для каждого столбца матрицы с четным номером (2, 4, ...) найти сумму его элементов. Условный оператор не использовать.

Matrix7. Дана матрица В каждой строке матрицы найти минимальный элемент.

Matrix8. Дана матрица В каждом столбце матрицы найти максимальный элемент.

Matrix9. Дана матрица Найти номер ее строки с наибольшей суммой элементов и вывести данный номер, а также значение наибольшей суммы.

Matrix10. Дана матрица Найти номер ее столбца с наименьшим произведением элементов и вывести данный номер, а также значение наименьшего произведения.

Matrix11. Дана матрица Найти максимальный среди минимальных элементов ее строк.

Matrix12. Дана матрица Найти минимальный среди максимальных элементов ее столбцов.

Matrix13. Дана матрица В каждой ее строке найти количество элементов, меньших среднего арифметического всех элементов этой строки.

Matrix14. Дана матрица В каждом ее столбце найти количество элементов, больших среднего арифметического всех элементов этого столбца.

Matrix15. Дана матрица Найти номера строки и столбца для элемента матрицы, наиболее близкого к среднему значению всех ее элементов.

Matrix16. Дана целочисленная матрица Найти номер первой из ее строк, содержащих равное количество положительных и отрицательных элементов (нулевые элементы матрицы не учитываются). Если таких строк нет, то вывести 0.

Matrix17. Дана целочисленная матрица Найти номер последнего из ее столбцов, содержащих равное количество положительных и отрицательных элементов (нулевые элементы матрицы не учитываются). Если таких столбцов нет, то вывести 0.

Matrix18. Дана целочисленная матрица Найти номер последней из ее строк, содержащих только четные числа. Если таких строк нет, то вывести 0.

Matrix19. Дана целочисленная матрица Найти номер первого из ее столбцов, содержащих только нечетные числа. Если таких столбцов нет, то вывести 0.

Matrix20. Дана целочисленная матрица элементы которой могут принимать значения от 0 до 100. Различные строки матрицы назовем похожими, если совпадают множества чисел, встречающихся в этих строках. Найти количество строк, похожих на первую строку данной матрицы.

Д32 Варианты домашнего задания (II семестр, 14 неделя)

ООП-1 Составить программу создания средствами ООП с использованием библиотеки Graph следующих управляемых с клавиатуры графических объектов:

1. набор из 10 пикселей разного цвета;
2. два ромба с диагоналями, параллельными краям экрана;
3. два эллипса (`Ellipse (Xc, Yc, 0, 360, Rx, Ry)`).

Организовать управление движением объектов курсорными клавишами по вертикали и по горизонтали. Менять размеры ромба и эллипса по вертикали и по горизонтали. Ввести возможность менять цвет объектов (изменением кода цвета).

ООП-2 Составить программу создания средствами ООП с использованием библиотеки Graph следующих управляемых с клавиатуры графических объектов:

1. набор из 20 пикселей одного цвета;
2. два прямоугольника со сторонами, параллельными краям экрана;
3. две дуги окружности (`Arc (Xc, Yc, 0, Uгол, R)`).

Организовать управление движением объектов курсорными клавишами по вертикали и по горизонтали. Менять размеры прямоугольника по вертикали и по горизонтали, а для дуги менять радиус и конечный угол (стартовый угол принять равным нулю). Ввести возможность менять цвет объектов (изменением кода цвета).

ООП-3 Составить программу создания средствами ООП с использованием библиотеки Graph следующих управляемых с клавиатуры графических объектов:

1. набор из 30 пикселей разного цвета;
2. два прямоугольных треугольника с катетами, параллельными краям экрана;
3. два отрезка толстой линии (стиль `SetLineStyle (0, 0, 3)`).

Организовать управление движением объектов курсорными клавишами по вертикали и по горизонтали. Менять размеры треугольников по вертикали и по горизонтали. Для отрезков - вращение по или против часовой стрелки. Ввести возможность менять цвет объектов (изменением кода цвета).

ООП-4 Составить программу создания средствами ООП с использованием библиотеки Graph следующих управляемых с клавиатуры графических объектов:

1. набор из 40 пикселей одного цвета;
2. два прямоугольника со сторонами, параллельными краям экрана;
3. два отрезка прямой линии.

Организовать управление движением объектов курсорными клавишами по вертикали и по горизонтали. Менять размеры прямоугольника по вертикали и по горизонтали, а для отрезка задать вращение по или против часовой стрелки. Ввести возможность менять цвет объектов (изменением кода цвета).

ООП-5 Составить программу создания средствами ООП с использованием библиотеки Graph следующих управляемых с клавиатуры графических объектов:

1. набор из 50 пикселей одного цвета;
2. два прямоугольных треугольника с гипотенузами, параллельными краям экрана;
3. две окружности.

Организовать управление движением объектов курсорными клавишами по вертикали и по горизонтали. Менять размеры треугольника по вертикали и по горизонтали, а для окружности задать изменение радиуса. Ввести возможность менять цвет объектов (изменением кода цвета).

Примерные темы курсовых работ

1 Работа с динамическими переменными

Составить программу, позволяющую организовать в виде стека, набор записей о студентах. Каждая запись должна содержать фамилию студента, пол и дату рождения.

Предусмотреть работу со стеком, которая включает в себя:

- ввод данных;
- редактирование данных;
- удаление данных из стека;
- вывод данных на экран;
- выполнение запроса 1 – нахождение самого старшего мужчины;
- выполнение запроса 2 – вывод фамилий студентов, начинающихся с заданной буквы.

Взаимодействие программы с пользователем должно быть реализовано с помощью меню, содержащему пункты: Добавить, Удалить, Просмотреть, Запрос 1, Запрос 2, Выход.

После выполнения любой функции необходимо обеспечить возврат в главное меню и выбор любой другой функции.

2 Программа обработки текстовых файлов

В заданном текстовом файле (имя вводится с помощью стандартного окна открытия файлов) выполнить:

- 1 Перекодировку из ASCII в кодировку ANSI;
- 2 Удалить все конечные пробелы строк;
- 3 Удалить заданное число пробелов в начале строк;
- 4 Удалить служебные символы (те, которых нет на клавиатуре);
- 5 Удалить те разделители строк, за которыми **не** следует другой разделитель строки либо пробел.

Перед запуском программы на выполнение в диалоговом окне должна быть реализована возможность указания, какие операции из перечисленных нужно выполнять.

Результат работы программы - записать в новый файл.

3 Двухнаправленный динамический список

Реализовать базу данных на основе двухнаправленного динамического списка, содержащую фамилию, имя, отчество человека, дату его рождения и знак зодиака.

Предусмотреть работу со списком, которая включает в себя:

- ввод данных;
- удаление данных из списка;
- вывод данных на экран;
- сортировку списка по фамилиям.

Взаимодействие программы с пользователем должно быть реализовано с помощью меню, содержащему пункты: Добавить, Удалить, Просмотреть, Сортировать, Выход.

После выполнения любой функции необходимо обеспечить возврат в главное меню и выбор любой другой функции.

Знак зодиака должен выбираться автоматически, в зависимости от даты.

Каждая запись должна содержать указатели на следующий и предыдущий элемент списка.

4 Калькулятор для работы с датами

Написать программу-калькулятор, которая позволяет по введенной дате D:

- вычислять количество дней в том месяце, которому принадлежит дата D;
- проверять правильность даты D;

- подсчитывать, сколько дней прошло с 1 января 1-го года нашей эры до даты D;
- определять день недели, на который приходится дата D (учтите, что 1 января 1-го года нашей эры было понедельником);
- вычислять ближайшую дату, соответствующую пятнице 13-го числа.

5 База данных «КАДРЫ»

Создать базу данных, содержащую следующие сведения о работниках:

- фамилия;
- табельный номер;
- пол (0 – 1);
- заработная плата;
- дата рождения;
- код образования (1 – 7);
- возраст (полных лет).

Обеспечить ввод данных с клавиатуры всех характеристик, кроме последней, которая должна вычисляться, запись данных в файл и открытие имеющегося файла.

Взаимодействие программы с пользователем должно быть реализовано с помощью меню, содержащему пункты: Добавить, Удалить, Просмотр, Сортировка по фамилии, Сортировка по возрасту, Выход. Для лиц с одинаковым возрастом вычислить среднюю заработную плату.

9 Калькулятор для работы с комплексными числами

Написать программу - калькулятор, который позволяет представлять комплексные числа в алгебраической, тригонометрической и показательных формах и производить над ними четыре арифметических действия.

Реализовать в Delphi.

6 Работа с римскими числами

Написать программу, позволяющую переводить числа из арабской в римскую систему и наоборот, а также производить над ними действия сложения, вычитания и умножения. (Максимальное число, с которым должна работать программа – 1000).

Реализовать в Delphi.

7 Криптографическая программа

Текст закодирован с помощью сетки, представленной ниже, где цифрой 0 обозначено отверстие. Для того чтобы декодировать сообщение (криптограмму), нужно наложить сетку на текст так, чтобы в отверстия можно было видеть символы закодированного текста. Первый раз сетка накладывается так, чтобы сторона, отмеченная знаком «+», была верхней, затем сетка поворачивается по часовой стрелке на 90 градусов, читается следующий набор символов и так далее до полного оборота сетки на 360 градусов.

```

+
1 0 1 0 1 0
1 1 1 1 0 1
1 1 0 1 1 1
1 0 1 1 0 1
1 1 1 1 0 0
1 1 1 0 1 1

```

Создать программу для ввода закодированного текста по строкам и расшифровки его с помощью данной сетки.

8 Игра «Дорожка»

Два противника двигают фишки, находящиеся на противоположных сторонах дорожки, разбитой на n полей (например, одна из вертикалей шахматной доски). За один ход участник может продвинуть свою фишку не более чем на m полей вперед или назад, при этом нельзя перепрыгивать через фишку партнера и покидать пределы дорожки. Проигрывает тот, у кого не окажется ходов, т.е. чья фишка будет загнана в одно из двух крайних полей дорожки.

Стратегия: если расстояние между фишками перед ходом партнера равно $(m+1)$ полю, то, какой бы ход он не сделал, мы своим следующим ходом поставим свою фишку вплотную к его, и он будет вынужден отступить. Теперь нужно просто копировать его ходы.

Итак, надо добиться, чтобы перед последним ходом партнера наши фишки разделяло $(m+1)$ полей. Анализируя “назад”, можно прийти к выводу, что перед предпоследним ходом партнера расстояние между фишками должно составлять $2(m+1)$ полей. Продолжая рассуждать, придем к простому положению: перед первым ходом партнера фишки должна разделять дистанция, измеряемая $k(m+1)$ полей, где k – любое целое число. Тогда, независимо от того, как будет ходить партнер, мы всегда будем иметь возможность сократить к его очередному ходу расстояние между фишками до целого числа отрезков длиной $(m+1)$.

Реализация – графический режим.

9 Работа со стеком

Сведения о студенте НГТИ состоят из его имени и фамилии, группы, в которой он учится, пяти оценок, полученных студентом за последнюю сессию. Составить программу, позволяющую организовать в виде стека, набор записей о студентах.

Предусмотреть работу со стеком, которая включает в себя:

- ввод данных;
- редактирование данных;
- удаление данных из стека;
- вывод данных на экран;
- выполнение запроса 1 – вывод списка отличниц (если последняя буква имени «а» или «я» – то это студентка, в противном случае – студент);
- выполнение запроса 2 – вывод списка групп, в которых насчитывается более 15 студентов, получивших в сессию только 4 и 5;
- выполнение запроса 3 - вывод числа двоечников на каждом курсе.

Взаимодействие программы с пользователем должно быть реализовано с помощью меню, содержащему пункты: Добавить, Удалить, Просмотреть, Запрос1, Запрос2, Запрос3, Выход.

После выполнения любой функции необходимо обеспечить возврат в главное меню и выбор любой другой функции. Данные в программу должны вводиться из файла.

10 Программа преобразования числа в текст

Программа должна преобразовывать денежные суммы в чековой книжке из формата чисел в словесный формат.

Функции программы:

- 1) вывод на экран окна для запроса исходной числовой суммы;
- 2) организация ввода данных с отображением их на экране и контролем ввода нечисловых символов, а также с возможностью удаления уже введенных числовых значений;
- 3) перевод введенной суммы в словесный формат;
- 4) вывод введенной суммы на экран в словесном формате.

Входными данными для программы является число - денежная сумма, вводимая пользователем с клавиатуры.

Выходные данные - словесное выражение, составленное в соответствии с введенной пользователем денежной суммы полученное с учетом правил русского языка.

Перечень вопросов для подготовки к экзамену

Вопросы к экзамену 1 семестр.

1. Понятие алгоритмических языков программирования. Интерпретаторы, компиляторы. Общая схема формирования .EXE файла из исходного кода. Этапы трансляции и компиляции.
2. Интегрированная среда программирования на примере Turbo Pascal. Написание текста программ, многооконный редактор.
3. Структура программы на языке Pascal (все разделы). Назначение каждого раздела. Очередность описания разделов программы.
4. Алфавит языка. Имена. Правила формирования имен.
5. Переменные: определение, назначение и типы. Предопределенные (стандартные) типы. Раздел описания переменных. Область действия переменных.
6. Использование комментариев в программе на языке Pascal. Символы обозначения комментариев. Вложенные комментарии.
7. Предопределенные (стандартные) типы языка Pascal: основные и производные от них. Выбор типа переменных при создании программ.
8. Арифметические операции над целыми и вещественными данными. Приоритеты операций. Изменение приоритета. Математические и тригонометрические функции. Целочисленные операции.
9. Логические операции "и", "или", "не", "или-не". Таблицы истинности для этих операций. Использование данных логических операций в условных выражениях.
10. Простой и составной операторы. Операторные скобки: назначение, правила и порядок их использования.
11. Оператор присваивания. Назначение, синтаксис и порядок использования. Совместимость типов данных в левой и правой частях оператора присваивания.
12. Управляющие операторы языка Pascal. Определение, назначение. Примеры управляющих операторов.
13. Условный оператор. Назначение, синтаксис и порядок использования.
14. Циклический процесс. Определение, назначение. Оператор цикла с заранее известным количеством повторений (синтаксис и порядок работы).
15. Циклический процесс. Определение, назначение. Оператор цикла с предусловием (синтаксис и порядок работы).
16. Циклический процесс. Определение, назначение. Оператор цикла с постусловием (синтаксис и порядок работы).
17. Подпрограммы. Определение и назначение. Общая структура подпрограмм. Два типа подпрограмм языка Pascal: процедуры и функции. Отличия процедур и функций.
18. Оператор процедуры. Порядок и правила описания и использования процедур. Использование процедур из стандартных библиотек языка Pascal.
19. Порядок описания и использования функций. Использование функций из стандартных библиотек языка Pascal.
20. Общая структура подпрограмм-функций. Правила завершения тела функций. Результат, возвращаемый функцией. Описание типа результата.
21. Описание переменных в подпрограммах. Область действия переменных. Вложенные подпрограммы.

22. Механизм передачи параметров в подпрограммы. Формальные и фактические параметры. Параметры-значения и параметры-переменные.
23. Понятие файла с точки зрения его использования в программе. Типы файлов в языке Pascal. Общая схема работы с файлом.
24. Текстовые файлы: процедуры и функции обработки данных; действия над указателем файла.
25. Операции файлового ввода-вывода. Обработка ошибок файлового ввода-вывода.

Вопросы к экзамену 2 семестр.

1. Общая структура типов данных Паскаля.
2. Перечисляемые типы, их применение, правила использования, примеры применения.
3. Тип-диапазон, правила описания, порядок и примеры использования.
4. Записи. Объявление типа записей. Операции с переменными типа запись. Обращения к полям записей. Оператор присоединения WITH. Массивы записей. Программирование типовых алгоритмов обработки записей.
5. Множества в Паскале. Основные понятия множеств. Эквивалентность. Описание. Определенные в Паскале операции над множествами. Примеры использования.
6. Расположение данных языка Pascal в оперативной памяти. Статическая и динамическая память. Указатели и динамическая память. Цели использования динамической памяти.
7. Выделение и освобождение динамической памяти.
8. Типизированные и нетипизированные указатели. Объявление типизированных указателей. Выделение и освобождение динамической памяти. Простейшие действия с указателями (занесение, копирование информации, копирование адреса).
9. Несвязанные динамические данные. Использование динамических переменных для работы с простыми переменными, массивами, записями.
10. Связанные динамические данные. Общие понятия. Организация взаимосвязей в связанных динамических данных. Организация списочных структур. Типы динамических списков.
11. Стек. Процедуры обработки стека. Создание стека. Добавление элемента стека. Удаление элемента стека.
12. Очереди. Процедуры обработки очереди. Создание очереди. Добавление элемента очереди. Удаление элемента очереди.
13. Основные понятия объектно-ориентированного программирования (ООП). Отличие ООП подхода к программированию от структурного.
14. Принципы ООП: описание объектов программы и программы как объекта. Иерархия объектов.
15. Инкапсуляция. Примеры.
16. Наследование. Примеры.
17. Полиморфизм. Примеры.
18. Реализация принципов объектно-ориентированного программирования.

Образцы экзаменационных билетов

Семестр 1

НОВОУРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ

Кафедра Автоматизации управления

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

по курсу «Программирование», направление 09.03.01, 1 семестр.

1. Описать процедуру $\text{RemoveX}(A, N, X)$, удаляющую из целочисленного массива A размера N элементы, равные целому числу X . Массив A и число N являются входными и выходными параметрами. С помощью этой процедуры удалить число 33 из массива $\{Z\}$, содержащего M элементов и вывести размер и содержимое полученного массива.

2. Дана строка символов. После каждого символа «.» (точка) вставить один символ пробела.

Преподаватель:

Тихонова Е.В.

Зав. кафедрой:

Степанов П.И.

НОВОУРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ

Кафедра Автоматизации управления

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 2

по курсу «Программирование», направление 09.03.01, 1 семестр.

1. Построить график функции $Y=X^2$, для $-2 \leq x \leq 2$. График должен занимать весь экран.

2. Описать процедуру $\text{Even}(K, m, L)$, возвращающую $L=\text{True}$, если в записи трехзначного целого числа K имеется цифра m ($0 \leq m \leq 9$) и False в противном случае. С ее помощью найти количество чисел, в которых есть цифра 5, содержащихся в массиве из N целых чисел.

Преподаватель:

Тихонова Е.В.

Зав. кафедрой:

Степанов П.И.

НОВОУРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ

Кафедра Автоматизации управления

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 3

по курсу «Программирование», направление 09.03.01, 1 семестр.

1. Описать функцию $\text{IsSquare}(K)$ логического типа, возвращающую True , если целый параметр K (> 0) является квадратом некоторого целого числа, и False в противном случае.

Операцию извлечения квадратного корня в функции не использовать. С ее помощью найти количество квадратов в массиве из 10 целых положительных чисел.

2. Дана строка символов (набор слов, разделенных одним или несколькими пробелами, начальных и конечных пробелов нет). Определить наибольшую длину слова в строке.

Преподаватель:

Тихонова Е.В.

Зав. кафедрой:

Степанов П.И.

НОВОУРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ

Кафедра Автоматизации управления

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 4

по курсу «Программирование», направление 09.03.01, 1 семестр.

1. Удалить из массива A длины N максимальный элемент, если все элементы разные. Вывести полученный массив. Тело программы должно иметь вид:

Begin

<Ввод исходных данных>

k:=Maximum(A,N); {поиск номера макс. элемента}

Delete_max(k,A,N); {удаление элемента с номером k}

<Вывод результатов>

End.

2. Построить в правой верхней четверти экрана график функции $y=8/(4+x^2)$, где $10 \leq x \leq 10$.

Преподаватель:

Тихонова Е.В.

Зав. кафедрой:

Степанов П.И.

Семестр 2

НОВОУРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ

Кафедра Автоматизации управления

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

по курсу «Программирование», направление 09.03.01, 2 семестр.

- 1 Динамическая память. Адреса и указатели. Объявление указателей. Типизированные и нетипизированные указатели.
- 2 Дана целочисленная матрица размером N×M. Найти номер строки и столбца, в которых находится наименьший положительный элемент. Разместить компоненты на форме:

N	4	M	3
ПУСК			
-3	4	-1	Ответ
9	-5	3	2
10	8	-4	

Преподаватель:

Тихонова Е.В.

Зав. кафедрой:

Степанов П.И.

НОВОУРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ

Кафедра Автоматизации управления

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 2

по курсу «Программирование», направление 09.03.01, 2 семестр.

- 1 Записи. Объявление типа записей. Операции с переменными типа запись. Обращения к полям записей. Оператор присоединения WITH. Массивы записей.
- 2 Сформировать матрицу N×N целых чисел. Получить вектор из N чисел, содержащий минимальные элементы соответствующих строк матрицы. Разместить компоненты на форме:

Размерность		3	^ v
Пуск			
1	2	-3	-3
0	4	7	0
2	-5	3	-5

Преподаватель:

Тихонова Е.В.

Зав. кафедрой:

Степанов П.И.

НОВОУРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ

Кафедра Автоматизации управления

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 3

по курсу «Программирование», направление 09.03.01, 2 семестр.

- 1 Основные понятия ООП. Объекты, классы, методы. Использование типа OBJECT.
- 2 Задано предложение, состоящее из слов, разделенных одним или несколькими пробелами. Определить, какая буква встречается чаще всего.

Обеспечить интерфейс вида:

Предложение	
МАМА МЫЛА МИЛУ	ОЧИСТИТЬ
СЧЕТ	
Имя файла:	E:\A.TXT ЗАПИСАТЬ

Предусмотреть запись предложения в файл, имя которого задается в программе.

Преподаватель:

Тихонова Е.В.

Зав. кафедрой:

Степанов П.И.