

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
(МИНОБРНАУКИ РОССИИ)

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНБАССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ДонГТУ»)

Факультет	информационных технологий и автоматизации производственных процессов
Кафедра	интеллектуальных систем и информационной безопасности



УТВЕРЖДАЮ
И.о. проректора
по учебной работе
Д.В. Мулов

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Основы программирования
(наименование дисциплины)

10.05.03 Информационная безопасность автоматизированных систем
(код, наименование специальности)

Безопасность открытых информационных систем
(специализация)

Квалификация специалист по защите информации
(бакалавр/специалист/магистр)

Форма обучения очная
(очная, очно-заочная, заочная)

1 Цели и задачи изучения дисциплины

Цели дисциплины. Целью изучения дисциплины «Основы программирования» является выработка у студентов передовых научно-технических воззрений, ориентации их на мировой уровень производительности труда, подготовке специалистов, которые должны обеспечить разработку качественных компьютерных программ, снижение материальных затрат, сокращение сроков проектирования.

Задачи изучения дисциплины:

- приобретение студентами знаний, умений и навыков в области алгоритмизации, программирования, основным этапам решения задач на ЭВМ;
- ознакомление с основами программирования на языке С;
- приобретение навыков использовать полученные знания в прикладных исследованиях, проектировании и эксплуатации информационных систем и технологий;
- умение создавать программные приложения для проведения инженерных расчетов, обработки числовой, текстовой и графической информации.

Дисциплина направлена на формирование общепрофессиональной (ОПК-7) компетенции выпускника.

2 Место дисциплины в структуре образовательной программы

Логико-структурный анализ дисциплины – курс входит в БЛОК 1 «Дисциплины (модули)», часть, формируемую участниками образовательных отношений подготовки студентов по направлению по специальности 10.05.03 Информационная безопасность автоматизированных систем (10.05.03-05 Безопасность открытых информационных систем).

Дисциплина реализуется кафедрой специализированных компьютерных систем. Основывается на базе дисциплин: «Информатика», «Основы алгоритмизации».

Является основой для изучения следующих дисциплин: «Объектно-ориентированное программирование», «Вычислительная математика», «Базы данных», «Безопасность баз данных», «Учебно-лабораторный практикум».

Для изучения дисциплины необходимы компетенции, сформированные у студента для решения профессиональных задач деятельности, связанных с разработкой программного обеспечения.

Курс является фундаментом для ориентации студентов в сфере разработки программного обеспечения информационных систем.

Общая трудоемкость освоения дисциплины составляет 10 зачетных единицы, 360 ак.ч. Программой дисциплины предусмотрены лекционные (72 ак.ч.), лабораторные (54 ак.ч.), практические (36 ак.ч.) занятия и самостоятельная работа студента (198 ак.ч.).

Дисциплина изучается на 1,2 курсе в 1-3 семестрах. Форма промежуточной аттестации – экзамен и дифференцированный зачёт.

3 Перечень результатов обучения по дисциплине, соотнесённых с планируемыми результатами освоения образовательной программы

Процесс изучения дисциплины «Основы программирования» направлен на формирование компетенции, представленной в таблице 1.

Таблица 1 – Компетенции, обязательные к освоению

Содержание компетенции	Код компетенции	Код и наименование индикатора достижения компетенции
Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ	ОПК-7	ОПК-7.1. Создает программы на языках общего назначения ОПК-7.2. Применяет методы и инструментальные средства программирования для решения профессиональных задач ОПК-7.3. Осуществляет обоснованный выбор инструментария программирования и способов организации программ

4 Объём и виды занятий по дисциплине

Общая трудоёмкость учебной дисциплины составляет 10 зачётных единицы, 360 ак.ч.

Самостоятельная работа студента (СРС) включает проработку материалов лекций, подготовку к практическим занятиям, текущему контролю, выполнение индивидуального задания, самостоятельное изучение материала и подготовку к экзамену и дифференцированному зачету.

При организации внеаудиторной самостоятельной работы по данной дисциплине используются формы и распределение бюджета времени на СРС для очной формы обучения в соответствии с таблицей 2.

Таблица 2 – Распределение бюджета времени на СРС

Вид учебной работы	Всего ак.ч.	Ак.ч. по семестрам		
		1	2	3
Аудиторная работа, в том числе:	162	90	54	18
Лекции (Л)	72	36	36	-
Практические занятия (ПЗ)	18	18	-	-
Лабораторные работы (ЛР)	54	36	18	-
Курсовая работа/курсовой проект	18	-	-	18
Самостоятельная работа студентов (СРС), в том числе:	198	54	90	54
Подготовка к лекциям	18	9	9	-
Подготовка к лабораторным работам	36	18	18	-
Подготовка к практическим занятиям / семинарам	27	9	-	18
Выполнение курсовой работы / проекта	20	-	-	20
Расчетно-графическая работа (РГР)	-	-	-	-
Реферат (индивидуальное задание)	12	-	12	-
Домашнее задание	12	-	12	-
Подготовка к контрольным работам	10	10	-	-
Подготовка к коллоквиуму	-	-	-	-
Аналитический информационный поиск	18	-	18	-
Работа в библиотеке	18	-	6	12
Подготовка к экзамену (диф.зачету)	27	8	15	4
Промежуточная аттестация – экзамен (Э), диф.зачет (ДЗ)	Э, ДЗ	Э	Э	ДЗ
Общая трудоемкость дисциплины				
ак.ч.	360	144	144	72
з.е.	10	4	4	2

5 Содержание дисциплины

С целью освоения компетенции, приведенной в п.3 дисциплина разбита на 15 тем:

- тема 1 (Начальные сведения о языках программирования);
- тема 2 (Характеристика языка Си);
- тема 3 (Арифметические операции и выражения);
- тема 4 (Операции сравнения и логические операции);
- тема 5 (Циклы);
- тема 6 (Массивы);
- тема 7 (Функции);
- тема 8 (Строки);
- тема 9 (Способы инициализации указателей);
- тема 10 (Указатели на функции);
- тема 11 (Поразрядные логические операции и операции сдвига);
- тема 12 (Типы данных, определяемые пользователем);
- тема 13 (Файлы текстовые и бинарные);
- тема 14 (Стадии препроцессорной обработки);
- тема 15 (Основные понятия объектно-ориентированного программирования).

Виды занятий по дисциплине и распределение аудиторных часов для очной формы приведены в таблице 3.

Таблица 3 – Виды занятий по дисциплине и распределение аудиторных часов (очная форма обучения)

№ п/п	Наименование темы (раздела) дисциплины	Содержание лекционных занятий	Трудоемкость в ак.ч.	Темы практических занятий	Трудоемкость в ак.ч.	Тема лабораторных занятий	Трудоемкость в ак.ч.
1	2	3	4	5	6	7	8
1	Начальные сведения о языках программирования	Характеристики и свойства языков программирования. Понятие среды разработки. Алгоритмы, их свойства, виды и способы записи. Программные средства представления алгоритмов. Основные этапы решения задачи на ЭВМ. Принципы структурного программирования.	2	Постановка задач, алгоритмы	2	Алгоритмы. Виды и способы представления. Знакомство со средой разработки	2
2	Характеристика языка Си	Основные программные объекты. Структура программы. Стандартные типы данных. Операторы описания. Стандартный ввод и вывод. Ввод вывод из текстовых файлов.	4	Работа с переменными и константами, ввод-вывод	2	Переменные и константы. Ввод-вывод	4
3	Арифметические операции и выражения	Вычисления, математические функции и библиотека math.h	4	Программирование алгоритмов линейной структуры	2	Программирование алгоритмов линейной структуры	6
4	Операции сравнения и логические операции	Условные операторы как инструмент языка для управления процессом обработки информации. Две формы условного оператора. Оператор безусловного перехода.	4	Разветвления и циклы	2	Программирование алгоритмов разветвляющейся и циклической структуры	6

Продолжение таблицы 3

1	2	3	4	5	6	7	8
5	Циклы	Три формы оператора цикла. Примеры применения операторов цикла в программах.	4	—	—	—	—
6	Массивы	Одномерные массивы. Базовые алгоритмы их обработки. Многомерные массивы. Указатель. Адресная арифметика. Связь указателей с массивами. Доступ к данным через указатели.	6	Обработка массивов	4	Обработка массивов	6
7	Функции	Определение, описание и вызов функции. Способы передачи параметров в функцию. Передача массивов как параметров. Классы памяти. Рекурсивные функции.	4	Функции	2	Функции	6
8	Строки	Строковые константы, стандартные функции для работы со строками. Массивы строк, массив указателей.	2	Указатели и строки	2	Указатели и строки	6
9	Способы инициализации указателей	Понятие динамической памяти и использование динамических переменных. Размещение в динамической памяти одномерных, двумерных массивов и массивов свободных строк	2	Работа с памятью программ	2	-	-

Продолжение таблицы 3

1	2	3	4	5	6	7	8
10	Указатели на функции	Указатели на функции. Передача функций функциям в качестве параметров	4	Указатели на функции	2	-	-
11	Поразрядные логические операции и операции сдвига	Побитовые операции и области их применения	4	—	—	Побитовые операции	4
12	Типы данных, определяемые пользователем	Структуры, объединения, битовые поля. Переименование типов. Перечисления. Использование битовых полей	8	—	—	Использование структур	4
13	Файлы текстовые и бинарные.	Файлы последовательного и прямого доступа. Режим работы в файлах прямого доступа. Функции для работы с файлами и директориями. Файловый ввод-вывод.	12	—	—	Обработка файлов	6
14	Стадии препроцессорной обработки.	Стадии препроцессорной обработки. Директивы. Макроподстановки. Условная компиляция	6	—	—	Преобразование типов	4

Завершение таблицы 3

1	2	3	4	5	6	7	8
15	Основные понятия объектно-ориентированного программирования (ООП).	Отличия ООП от процедурного программирования. Принцип инкапсуляции. Понятия наследования и полиморфизма. Конструкторы и деструкторы. Дружественные классы и функции. Статические элементы класса. Константные методы и объекты. Указатель this. Указатели и ссылки на объекты, массивы объектов.	6	—	—	-	-
Всего аудиторных часов			72	18		54	

Таблица 4 – Виды занятий по дисциплине и распределение аудиторных часов по курсовой работе (очная форма обучения)

№ п/п	Тема занятия	Трудоемкость в ак.ч.
1	Описание задачи, разработка содержательной блок-схемы алгоритма программы	2
2	Определение и описание переменных, констант и подпрограмм (функций)	2
3	Разработка интерфейса программы	2
4	Разработка программного кода основного алгоритма	4
5	Разработка программного кода обработчиков событий	4
6	Выполнение тестового примера	2
7	Оформление пояснительной записки	2
Всего аудиторных часов		18

6 Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации студентов по дисциплине

6.1 Критерии оценивания

В соответствии с Положением о кредитно-модульной системе организации образовательного процесса ФГБОУ ВО «ДонГТУ» (https://www.dstu.education/images/structure/license_certificate/polog_kred_modul.pdf) при оценивании сформированности компетенций по дисциплине используется 100-балльная шкала.

Перечень компетенций по дисциплине и способы оценивания знаний приведены в таблице 5.

Таблица 5 – Перечень компетенций по дисциплине и способы оценивания знаний

Код и наименование компетенции	Способ оценивания	Оценочное средство
ОПК-7	Экзамен	Комплект контролирующих материалов для экзамена
ОПК-7	Дифференцированный зачет	Комплект контролирующих материалов для дифзачета

Всего по текущей работе в семестре студент может набрать 100 баллов, в том числе:

- контрольные работы на практических занятиях (4 контрольных работы) – всего 30 баллов;
- лабораторные работы – всего 60 баллов;
- за выполнение реферата – всего 10 баллов.

Экзамен проставляется автоматически, если студент набрал в течении семестра не менее 60 баллов и отчитался за каждую контрольную точку. Минимальное количество баллов по каждому из видов текущей работы составляет 60% от максимального.

Экзамен по дисциплине «Основы программирования» проводится по результатам работы в семестре. В случае, если полученная в семестре сумма баллов не устраивает студента, во время сессии студент имеет право повысить итоговую оценку либо в форме устного собеседования по приведенным ниже вопросам (п.п. 6.5), либо в результате тестирования.

Шкала оценивания знаний при проведении промежуточной аттестации приведена в таблице 6.

Таблица 6 – Шкала оценивания знаний

Сумма баллов за все виды учебной деятельности	Оценка по национальной шкале зачёт/экзамен
0-59	Не зачтено/неудовлетворительно
60-73	Зачтено/удовлетворительно
74-89	Зачтено/хорошо
90-100	Зачтено/отлично

6.2 Домашние задания

Задание 1. Создать структуру согласно своего варианта задания к лабораторной работе № 2 (2 семестр). Написать программу, обеспечивающую автоматическую генерацию 10 переменных типа структур. Диапазон граничных значений вводить с клавиатуры.

Задание 2. Написать программу, которая заносит в файл и считывает из него структуры из задания 1.

Задание 3. Обеспечить вывод из файла, сформированного в результате выполнения задания 2, в виде таблицы на экран с окаймлением границ ячеек. Для окаймления ячеек использовать символы псевдографики в кодировке UTF-8.

6.3 Темы для рефератов (презентаций) – индивидуальное задание

- 1) Типы данных Си, определяемые пользователем.
- 2) Работа с текстовыми файлами в Си.
- 3) Работа с файлами, содержащими типы данных пользователя в Си.
- 4) Вывод данных в Си в виде таблиц.
- 5) Как функция Си работает с локальными и глобальными переменными.
- 6) Передача параметров в функции Си различными способами.
- 7) Динамические массивы в Си.
- 8) Рекурсивные функции в Си.
- 9) Многопоточный ввод-вывод в Си.
- 10) Возможности компиляторов Си различных производителей.
- 11) Компиляторы Си, поставляемые по программе OpenSource.
- 12) Массивы строк в Си: поиск и замена символов и фрагментов строк.
- 13) Рекурсивная обработка строк в Си.
- 14) Семейство Си-подобных языков программирования.
- 15) Особенности использования языка Си в Windows/Linux.
- 16) Вывод таблиц на экран в программах Си с использованием псевдографики.
- 17) Использование моделей памяти в программах Си.
- 18) Свободные массивы в языке Си.
- 19) Управление версиями программы на Си для Windows/Linux с помощью директив условной компиляции.
- 20) Сортировка данных в файлах Си.

6.4 Оценочные средства для самостоятельной работы и текущего контроля успеваемости

Контрольная работа № 1 (1 семестр). Выполняется в виде решения тестовых задач:

1) Для переменных заданы максимальные, минимальные и начальные значения. Выберите для них подходящие типы данных и запишите объявления для программы.

2) Какие значения будут иметь переменные после выполнения приведенного ниже фрагмента программы? Покажите ход решения.

Контрольная работа № 2 (1 семестр). Выполняется в виде решения тестовых задач вида:

1) Дана программа с ошибками. Найти и исправить ошибки.

2) Дана программа с использованием операторов цикла и условия. Переписать ее с использованием других операторов, использующих аналогичные функции.

Контрольная работа № 3 (1 семестр). Выполняется в виде решения тестовых задач:

Дан массив размерностью $n \times m$. Выполнить выборку элементов k -й строки (j -го столбца), рассчитать среднее значение элементов массива, выделить в отдельный вектор элементы массива согласно признакам.

Контрольная работа итоговая (1 семестр). Выполняется в виде решения тестовых задач, собранных из контрольных работ №№ 1 – 3.

Во втором семестре контрольные работы не проводятся.

6.5 Вопросы для подготовки к экзамену

I семестр

- 1) Какова классификация языков программирования?
- 2) Какие известны этапы и особенности решения задач с помощью ЭВМ?
- 3) Какие известны алгоритмы, их свойства, способы описания алгоритмов?
- 4) Что такое блок-схемы алгоритмов?
- 5) Какова структура программы?
- 6) Что такое структурное программирование? Его основные принципы? Как формулируется теорема о структурировании?
- 7) Какова классификация языков программирования?
- 8) Какие свойства языков программирования, характеризуют качество программ?
- 9) Что такое алфавит, синтаксис и семантика языка программирования?
- 10) Какие есть способы описания синтаксиса?
- 11) Что такое идентификатор в языках программирования? Как определить идентификатор с помощью синтаксической диаграммы и языка БНФ?
- 12) Каковы основные характеристики языка Си?
- 13) Что такое структура программы в языке Си?
- 14) Что такое тип данных?
- 15) Каковы основные типы данных в языке Си? Что такое квалификаторы и спецификаторы?
- 16) Как задаются константы в языке Си?
- 17) Что такое переменные в языке Си, их описание и инициализация?
- 18) Что такое стандартный ввод данных в языке Си?
- 19) Что такое стандартный вывод данных в языке Си?
- 20) Какие есть арифметические операции в языке Си?
- 21) Какие есть математические функции библиотеки `math.h`?
- 22) Какие есть операции присваивания и корректирующие операции присваивания в языке Си?
- 23) Что такое явные и неявные преобразования типов в языке Си?
- 24) Какие есть операции отношения в языке Си?
- 25) Какие есть логические операции в языке Си? Каковы правила вычислений логических выражений?
- 26) Какие есть условные выражения в языке Си?
- 27) Как организовать бинарное и множественное ветвления в языке Си?
- 28) Как организованы циклы в языке Си?
- 29) Какие есть операторы разрыва, продолжения, перехода и возврата в языке Си?
- 30) Что такое функции в языке Си, как их объявить и определить? Что такое аргументы функций формальные и фактические?
- 31) Какие есть классы памяти в языке Си?
- 32) Как производится инициализация переменных, принадлежащих различным классам памяти в языке Си?

- 33) Какие есть модели памяти?
- 34) Что такое одномерные массивы в языке Си, как производится их описание и инициализация?
- 35) Как передать одномерные массивы функциям в качестве параметров в языке Си?
- 36) Что такое многомерные массивы в языке Си, их описание и инициализация?
- 37) Как передать многомерные массивы функциям в качестве параметров в языке Си?
- 38) Каков принцип рекурсии? Какие Вы знаете примеры формирования рекурсивных функций?
- 39) Что такое строки в языке Си?
- 40) Какие есть стандартные функции работы со строками в языке Си?
- 41) Что такое указатели в языке Си, их описание, инициализация? Какие есть операции над указателями?
- 42) Какова связь массивов с указателями в языке Си?
- 43) Какие есть функции для работы с динамической памятью?
- 44) Как создать одномерный динамический массив?
- 45) Что такое свободные массивы в языке Си?
- 46) Что такое указатели на функции в языке Си (описание типа указатель на функцию, передача указателя на функцию в качестве параметров)?

II семестр

- 1) Какие есть побитовые операции в языке Си?
- 2) Что такое структуры в языке Си? Их описание и использование, передача структур в качестве параметра переменных.
- 3) Как создать в динамической памяти структуры для работы с матрицами произвольного порядка?
- 4) Что такое поля битов в структурах в языке Си?
- 5) Что такое перечисляемый тип в языке Си?
- 6) Что такое Оператор определения типов в языке Си?
- 7) Что такое объединения в языке Си?
- 8) Что такое файлы в языке Си? Что такое функции ввода-вывода потоком?
- 9) Как организован прямой доступ к потокам в языке Си?
- 10) Какие функции содержит библиотека `conio.h`?
- 11) Как организована работа с файлами на нижнем уровне?
- 12) Что такое и зачем нужен препроцессор языка Си?
- 13) Что такое условная компиляция в языке Си?
- 14) Что такое модульное программирование?
- 15) Создание программ на языке Си из нескольких файлов?
- 16) Каков принцип инкапсуляции? Что такое наследование и полиморфизм?
- 17) Что такое конструкторы и деструкторы? Что такое дружественные классы и функции?
- 18) Что такое статические элементы класса? Что такое константные методы и объекты?

- 19) Что такое указатели и ссылки на объекты, массивы объектов?
- 20) Какие есть классы памяти в языке Си?
- 21) Какие есть модели памяти в Си?
- 22) Что такое одномерные массивы в языке Си, как производится их описание и инициализация?
- 23) Как передать одномерные массивы функциям в качестве параметров в языке Си?
- 24) Что такое многомерные массивы в языке Си, их описание и инициализация?
- 25) Как передать многомерные массивы функциям в качестве параметров в языке Си?
- 26) Что такое указатели в языке Си, их описание, инициализация? Операции над указателями?
- 27) Как создать одномерный динамический массив?
- 28) Что такое свободные массивы в языке Си?
- 29) Что такое указатели на функции в языке Си (описание типа указатель на функцию, передача указателя на функцию в качестве параметров)?
- 30) Что такое структуры в языке Си? Их описание и использование, передача структур в качестве параметра переменных.
- 31) Как создать в динамической памяти структуры для работы с матрицами произвольного порядка?
- 32) Что такое роля битов в структурах в языке Си?
- 33) Какова организация прямого доступа к потокам в языке Си?
- 34) Как работать с файлами на нижнем уровне?
- 35) Что такое препроцессор языка Си?
- 36) Как происходит условная компиляция в языке Си?

6.6 Примерная тематика курсовых работ

Разработать программу с использованием языка программирования Си.

Задача решается согласно индивидуального варианта, приведенного в таблице ниже. Индивидуальное задание студент выбирает самостоятельно (таблица 7).

Таблица 7 – Индивидуальные варианты заданий

№	Задача	Примечание
1	2	3
1	Статистическая обработка данных *	1) Ввод данных из файла 2) Вычисление для массива данных: – Среднего значения – Дисперсии – Математического ожидания – Среднеквадратичного отклонения – Среднегеометричного значения 3) Запись результатов в отдельный файл *.html

Продолжение таблицы 7

1	2	3
2	Аппроксимация, интерполяция и экстраполяция функций, заданных таблицей данных *	1) Ввод данных из файла *.txt 2) Выбор наилучшего метода аппроксимации по критерию наименьшей ошибки 3) Интерполяция и экстраполяция за пределами диапазона исходных данных 4) Вывод результатов на форму и в файл *.html.
3	Решение систем линейных уравнений произвольной размерности методом Гаусса *	1) Ввод данных из файла *.txt 2) Анализ исходных данных 3) Выбор метода решения 4) Вывод результатов на экран и в файл 5) Предусмотреть решение недоопределенных и переопределенных уравнений 6) Вывод результатов в файл *.html.
4	Поиск экстремумов функций первого и второго порядка, заданных таблицей данных *	1) Ввод данных из файла 2) Выбор метода поиска *.txt 3) Поиск глобальных и локальных экстремумов 4) Вывод результатов на экран и в файл *.html.
5	Интегрирование функций, заданных таблицей данных *	1) Ввод данных из файла *.txt 2) Выбор метода интегрирования (3) 3) Вывод результатов на форму и в файл *.html.
6	Интегрирование функций, заданных формулой **	1) Ввод данных из файла *.txt 2) Выбор метода интегрирования (2) 3) Вывод результатов на форму и в файл *.html.
7	Дифференцирование функций, заданных формулой **	1) Ввод данных из файла *.txt 2) Выбор метода дифференцирования (2) 3) Вывод результатов на форму и в файл *.html.
8	Работа с квадратными матрицами произвольного размера	1) Ввод данных из файла *.txt 2) Вычисление детерминанта 3) Вычисление суммы элементов матрицы 4) Поиск минимального и максимального элементов с определением индексов строки и столбца. 5) Поиск нулевых элементов с определением индексов строки и столбца. 6) Вывод исходных данных и результатов в файл *.html.
9	Работа с массивами произвольного размера	1) Ввод данных из файла *.txt 2) Вычисление суммы, разности, произведения и частного двух массивов размером $n \times k$ и $k \times \ell$. 3) Поиск минимального и максимального элементов массивов с определением индексов строки и столбца. 4) Вывод исходных данных и результатов в файл *.html.
10	Вычисление производных функций, заданных таблицей данных *	1) Ввод данных из файла *.txt 2) Выбор метода дифференцирования (2) 3) Вывод результатов на экран и в файл

Продолжение таблицы 7

1	2	3
11	Сортировка данных в файлах	1) Файл формата *.txt может содержать: - один или несколько столбцов из данными - числа или слова 2) Сортировки осуществляется за одним или несколькими столбцами 3) Для сортировки может задаваться дополнительный признак 4) Результат вывести в файл формата *.html.
12	Вычисление функций, заданных аналитически **	1) Функция задается аналитически, например, так: $y = 20 * \cos(\sqrt{x} * 2)$ 2) Функция может содержать только допустимые символы, знаки операций 3) Предусмотреть возможность записи таблицы функции в файл *.html. 4) Предусмотреть использование функций $\sin, \cos, \operatorname{tg}, \sqrt{}, \sqrt[n]{}, e^x$
13	Поиск экстремумов функций, заданных аналитически **	1) Функция задается аналитически, например, так: $y = 20 * \cos(\sqrt{x} * 2)$ 2) Функция может содержать только допустимые символы, знаки операций 3) Предусмотреть возможность записи таблицы функции в файл *.html. 4) Предусмотреть использование функций $\sin, \cos, \operatorname{tg}, \sqrt{}, \sqrt[n]{}, e^x$ 5) Поиск глобального и локальных экстремумов функции
14	Регрессионный анализ функции, заданной таблицей данных *	1) Ввод данных из файла *.txt 2) Подбор наилучшей регрессионной зависимости 3) Вычисление коэффициентов регрессии 4) Запись исходных данных и результатов в отдельный файл *.html
15	Решение систем линейных уравнений произвольной размерности итерационным методом *	1) Ввод данных из файла *.txt 2) Анализ исходных данных 3) Выбор метода решения 4) Вывод результатов на экран и в файл 5) Предусмотреть решение недоопределенных и переопределенных уравнений 6) Вывод результатов в файл *.html.
16	Регрессионный анализ многомерной функции, заданной таблицей данных **	1) Ввод данных из файла *.txt 2) Количество независимых переменных – от 2 до 5 3) Количество зависимых переменных – от 1 до 3 4) Подбор наилучшей регрессионной зависимости 5) Вычисление коэффициентов регрессии 6) Построение таблицы исходной функции и таблицы функции регрессии. 7) Вычисление ошибки регрессии 7) Запись исходных данных и результатов в файл *.html

Продолжение таблицы 7

1	2	3
17	Описание многомерной функции, заданной таблицей данных, с помощью системы одновременных уравнений **	1) Ввод данных из файла *.txt 2) Количество независимых сменных – от 2 до 5 3) Количество зависимых сменных – от 1 до 3 3) Подбор наилучшей регрессионной зависимости 4) Вычисление коэффициентов регрессии 5) Построение таблицы исходной функции и таблицы функции регрессии. 6) Вычисление ошибки регрессии 7) Запись исходных данных и результатов в файл *.html
18	Представление функции, заданной таблицей данных, с помощью рядов Фурье **	1) Ввод данных из файла *.txt 2) Количество гармоник определяется заданной ошибкой вычисления: от 0,01% до 1%. 3) Запись исходных данных, результатов в виде таблицы в отдельный файл *.html
19	Аппроксимация, интерполяция и экстраполяция функций, заданных таблицей данных за методом Лагранжа *	1) Ввод данных из файла *.txt 2) Интерполяция и экстраполяция за пределами диапазона исходных данных 3) Вычисление ошибок интерполяции и экстраполяции 4) Вывод результатов на форму и в файл *.html.
20	Решение систем нелинейных уравнений произвольной размерности методом Ньютона *	1) Ввод данных из файла *.txt 2) Анализ исходных данных 3) Выбор метода решения 4) Вывод результатов на форму и в файл 5) Предусмотреть использование функций $\sin$, $\cos$, tg , $\sqrt{\quad}$, $\sqrt[x]{\quad}$, e^x 6) Вывод результатов в файл *.html.
21	Решение систем нелинейных уравнений произвольной размерности методом итераций *	1) Ввод данных из файла *.txt 2) Анализ исходных данных 3) Выбор метода решения 4) Вывод результатов на форму и в файл 5) Предусмотреть использование функций $\sin$, $\cos$, tg , $\sqrt{\quad}$, $\sqrt[x]{\quad}$, e^x 6) Вывод результатов в файл *.html.
22	Решение систем нелинейных уравнений произвольной размерности методом покоординатного спуска *	1) Ввод данных из файла *.txt 2) Анализ исходных данных и выбор метода решения 3) Вывод результатов на форму и в файл 4) Предусмотреть использование функций $\sin$, $\cos$, tg , $\sqrt{\quad}$, $\sqrt[x]{\quad}$, e^x 5) Вывод результатов в файл *.html.
23	Решение систем нелинейных уравнений произвольной размерности методом градиентного спуска *	1) Ввод данных из файла *.txt 2) Анализ исходных данных 3) Выбор метода решения 4) Вывод результатов на форму и в файл 5) Предусмотреть использование функций $\sin$, $\cos$, tg , $\sqrt{\quad}$, $\sqrt[x]{\quad}$, e^x 6) Вывод результатов в файл *.html.

Завершение таблицы 7

1	2	3
24	Решение систем линейных уравнений произвольной размерности методом обращения *	1) Ввод данных из файла *.txt 2) Анализ исходных данных 3) Выбор метода решения 4) Вывод результатов на экран и в файл 5) Предусмотреть решение недоопределенных и переопределенных уравнений 6) Вывод результатов в файл *.html.

Примечания:

* – на повышенную оценку (от 74 до 90 баллов);

** – на повышенную оценку (от 90 до 100 баллов).

В таблице 8 представлена структура пояснительной записки.

Таблица 8 – Структура пояснительной записки

Раздел	Количество страниц
1	2
Титульный лист	1
Задание и календарный план	1
РЕФЕРАТ	1
ВВЕДЕНИЕ	2-3
1 АНАЛИЗ ЗАДАНИЯ	
1.1 Теоретические сведения, необходимые для выполнения работы	5-6
1.2 Концепция решения задачи	1-2
1.3 Описание переменных и констант	2-3
2 РАЗРАБОТКА ТРЕБОВАНИЙ К ПРОГРАММЕ И АЛГОРИТМОВ	
2.1 Разработка общих требований к программе и ее структуры	3-5
2.2 Разработка интерфейса пользователя	3-5
2.3 Разработка алгоритмов расчетов	5-7
3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ И УПРАВЛЯЮЩЕЙ ПРОГРАММЫ	
3.1 Создание подпрограмм для реализации алгоритмов	2-4
3.2 Реализация программного кода для управляющей программы	3-4
3.3 Тестирование программы	1-2
ВЫВОДЫ	1
ПЕРЕЧЕНЬ ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	1
ПРИЛОЖЕНИЕ А ФОРМЫ ПРОГРАММЫ	2-4
ПРИЛОЖЕНИЕ Б ПРОГРАММНОЕ КОД	5-6
ПРИЛОЖЕНИЕ В ТЕСТОВЫЙ ПРИМЕР	2-4
ВСЕГО	40-60

После завершения работы преподавателю предоставляется пояснительная записка в объеме 40-60 страниц в печатном и электронном виде в формате Word 2003/2007, оформленная по требованиям стандартов, а также рабочая программа (файлы проекта и скомпилированный *exe-файл*), записанную на CD/DVD – диск.

Исходные данные для тестирования программы студент создает самостоятельно. Для заданий, которые связаны с обработкой статистических данных, размер выборки должен быть не менее 50 записей.

7 Учебно-методическое и информационное обеспечение дисциплины

7.1 Рекомендуемая литература

Основная литература

1. Кузин, А.В. Программирование на языке СИ : учебное пособие для студентов высших учебных заведений, обучающихся по укрупненной группе специальностей 09.00.00 "Информатика и вычислительная техника" (квалификация (степень) "бакалавр") /А.В. Кузин, Е.В. Чумакова — Москва : ФОРУМ, 2022. 143 с. — 1 экз.

Дополнительная литература

1. Брайан Керниган, Деннис Ритчи. Язык программирования С. — Москва: Вильямс, 1992. — 272 с. — 4 экз.

2. Стенли Липпман Язык программирования С++ [Электронный ресурс]: полное руководство/ Стенли Липпман, ЖозиЛажойе— Электрон. текстовые данные.— М.: ДМК Пресс, 2019.— 1104 с. Режим доступа: <http://www.iprbookshop.ru/6899.html> (Дата обращения 26.08.2024).

Учебно-методическое обеспечение

1. Сборник задач к практическим занятиям по курсу «Программирование» : практикум (для студ. специальности 10.05.03 «Информационная безопасность автоматизированных систем» 1 курса очной формы обучения) / сост.: Е.Е. Бизянов, А.А. Гутник, Л.В. Толстова ; Каф. Специализированных компьютерных систем . — Алчевск : ФГБОУ ВО ДонГТУ, 2023 . — 29 с. — URL: <http://library.dstu.education/download.php?rec=132304>.

2. Бизянов, Е.Е. Основы программирования. Часть 1 : лабораторный практикум / Е.Е. Бизянов, Л.В. Толстова. — Алчевск :ФГБОУ ВО "ДонГТУ", 2023. — 134 с. — URL: <http://library.dstu.education/download.php?rec=103329>.

3. Бизянов, Е.Е. Основы программирования. Часть 2 : лабораторный практикум / Е.Е. Бизянов, Л.В. Толстова. — Алчевск : ФГБОУ ВО "ДонГТУ", 2023) – 92 с. — URL: <http://library.dstu.education/download.php?rec=132257>.

7.2 Базы данных, электронно-библиотечные системы, информационно-справочные и поисковые системы

1. Научная библиотека ДонГТУ : официальный сайт.— Алчевск. — URL: library.dstu.education.—Текст : электронный.

2. Научно-техническая библиотека БГТУ им. Шухова : официальный сайт. — Белгород. — URL: <http://ntb.bstu.ru/jirbis2/>.—Текст : электронный.

3. Консультант студента :электронно-библиотечная система.— Москва. — URL: <http://www.studentlibrary.ru/cgi-bin/mb4x>.—Текст : электронный.

4. Университетская библиотека онлайн :электронно-библиотечная система.— URL: http://biblioclub.ru/index.php?page=main_ub_red.—Текст : электронный.

5. Сайт кафедры ИСИБ <http://scs.dstu.education>.

Материально-техническое обеспечение представлено в таблице 9.

Таблица 9 – Материально-техническое обеспечение

Наименование оборудованных учебных кабинетов	Адрес (местополо- жение) учебных кабинетов
Специальные помещения: <i>Мультимедийная аудитория. (60 посадочных мест), оборудованная специализированной (учебной) мебелью (скамья учебная –20 шт., стол– 1 шт., доска аудиторная– 1 шт.), учебное ПК (монитор + системный блок), мультимедийная стойка с оборудованием – 1 шт., широкоформатный экран.</i> Аудитории для проведения лекций: <i>Компьютерные классы (22посадочных места), оборудованный учебной мебелью, компьютерами с неограниченным доступом к сети Интернет, включая доступ к ЭБС:</i> ПК Intel Core 2 DUO 2)5 Ghz, 1024,160 – 11шт.; ПК Intel Celeron 2)0, 256, 40- 1 шт.Доска– 1 шт.	ауд. <u>207</u> корп.<u>4</u> ауд. <u>208</u> корп. <u>4</u> ауд. <u>211</u> корп. <u>4</u>

Лист согласования рабочей программы дисциплины

Разработал:

И.о заведующего кафедрой
интеллектуальных систем
и информационной безопасности
(должность)


(подпись) Е.Е.Бизянов
Ф.И.О.)

(должность)

(подпись) Ф.И.О.)

(должность)

(подпись) Ф.И.О.)

И.о заведующего кафедрой
интеллектуальных систем
и информационной безопасности


(подпись) Бизянов Е.Е.
Ф.И.О.)

Протокол № 1 заседания кафедры СКС от 27.08.2024 г.

И.о.декана факультета

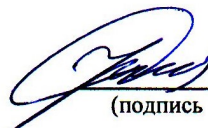

(подпись) В.В. Дьячкова
Ф.И.О.)

Согласовано:

Председатель методической
комиссии по специальности
10.05.03 Информационная безопасность
автоматизированных систем


(подпись) Е.Е. Бизянов
(Ф.И.О.)

Начальник учебно-методического центра


(подпись) О.А.Коваленко
Ф.И.О.)

Лист изменений и дополнений

Номер изменения, дата внесения изменения, номер страницы для внесения изменений	
ДО ВНЕСЕНИЯ ИЗМЕНЕНИЙ:	ПОСЛЕ ВНЕСЕНИЯ ИЗМЕНЕНИЙ:
Основание:	
Подпись лица, ответственного за внесение изменений	