

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ТАДЖИКИСТАН
МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКО-ТАДЖИКСКИЙ (СЛАВЯНСКИЙ) УНИВЕРСИТЕТ»

«Утверждаю»
Декан ЕНФ 
Муродзода Д.С.
« 08 » _____ 2024 г.



РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Направление подготовки – 09.03.03. “Прикладная информатика”

Профиль – Прикладная информатика в экономике

Форма подготовки - очная

Уровень подготовки – бакалавриат

ДУШАНБЕ 2024

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования, утверждённого приказом Министерства образования и науки РФ от 19 сентября 2017 г. № 922

При разработке рабочей программы учитываются


- требования работодателей, профессиональных стандартов по направлению / специальности (при наличии) (для общепрофессиональных и профессиональных дисциплин);
- содержание программ дисциплин, изучаемых на предыдущих и последующих этапах обучения;
- новейшие достижения в данной предметной области.

Рабочая программа обсуждена на заседании кафедры Информатики и ИТ., протокол № 1 от 28 августа 2024 г.

Рабочая программа утверждена УМС естественнонаучного факультета, протокол № 1 от 29 августа 2024 г.

Рабочая программа утверждена Учёным советом естественнонаучного факультета, протокол № 1 от 30 августа 2024г.

Заведующий кафедрой, к.э.н., доцент  Лешукович А.И.

Зам. председателя УМС факультета
к. ф-м.н., доцент  Халимов И.И.

Разработчик, к.э.н., доцент  Лешукович А.И.

Расписание занятий дисциплины

Ф.И.О. преподавателя	Аудиторные занятия		Приём СРС	Место работы преподавателя
	лекция	Практические занятия (КСР, лаб.)		
Лешукович А. И.	Пятница, 12:40-14:00 второй корпус: Ауд.213	Среда, 11:00-14:00 второй корпус: Ауд.224	Пятница, 10:00-11:00	РТСУ, кафедра информатики и ИТ, Корпус 2, 216 каб.

1. ЦЕЛИ И ЗАДАЧИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ

1.1. Цели изучения дисциплины

Целью дисциплины является приобретение студентами знания по алгоритмизации задач с использованием моделей; расширение кругозора и приобретение навыков по разработке программ; ознакомление с системами программирования и их применения в системах обработки информации; использование стандартных алгоритмов при решении задач прикладного характера; знание способов представления алгоритмов; установление взаимосвязи построения алгоритмов с разработкой программы; освоение способов разработки программ с применением методов структурного и объектно-ориентированного программирования, а также изучение методов современных языков программирования

1.2. Задачи изучения дисциплины

Задачи дисциплины формулируются в соответствии с требованиями ФГОС, предъявляемые к компетенциям обучающегося.

1.3. В результате изучения дисциплины «Основы алгоритмизации и языки программирования» у обучающихся формируются следующие общепрофессиональные компетенции:

Таблица 1.

Код компетенции	Содержание компетенции	Перечень планируемых результатов обучения по дисциплине (индикаторы достижения компетенций)	Виды оценочных средств
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	ИУК-1.1. Анализирует задачу, выделяя ее базовые составляющие. Осуществляет декомпозицию задачи; ИУК-1.2. Находит и критически анализирует информацию, необходимую для решения поставленной задачи; ИУК-1.3. Рассматривает различные варианты решения задачи, оценивая их достоинства и недостатки; ИУК-1.4. Грамотно, логично, аргументированно формирует собственные	Тестирование. Контроль самостоятельной работы. Отчеты по практическим работам. Контрольная работа. Устный опрос.

		суждения и оценки. Отличает факты от мнений, интерпретаций, оценок и т.д. в рассуждениях других участников деятельности; ИУК-1.5. Определяет и оценивает практические последствия возможных решений задачи.	
ПК-8	Способен проводить тестирование компонентов программного обеспечения информационных систем	ИПК-8.1. Использует современные технологии и методы тестирования, специализированное программное обеспечение автоматизации тестирования ИПК-8.2. Разрабатывает программу и методику тестирования, проводит тестирование компонентов программного обеспечения ИС в соответствии с ними ИПК-8.3. Применяет основные инструментальные средства тестирования компонентов программного обеспечения ИС	Тестирование. Контроль самостоятельной работы. Отчеты по практическим работам. Контрольная работа. Устный опрос.
ПК-9	Способен осуществлять ведение баз данных и поддержку информационного обеспечения решения прикладных задач	ИПК-9.1. Применяет современные технологии разработки и ведения баз данных ИПК-9.2. Проектирует и разрабатывает базы данных, использует их для поддержки информационного обеспечения решения прикладных задач ИПК-9.3. Участвует в эксплуатации баз данных, поддержке информационного обеспечения решения прикладных задач	Тестирование. Контроль самостоятельной работы. Отчеты по практическим работам. Контрольная работа. Устный опрос.

2.МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Данная дисциплина входит в базовый цикл вариативной части дисциплины Б1.В.02 ОПОП бакалавриата ФГОС ВО и является обязательной дисциплиной.

Таблица 2.

№ п/п	Наименование дисциплины	Семестр	Место дисциплины в структуре ООП
1.	<i>Программирование</i>	1-2-3	Б1.О.13
2.	<i>Практикум по программированию</i>	2-4	Б1.О.21
3.	<i>Интеллектуальные информационные системы</i>	5	Б1.В.05

4.	<i>Проектирование информационных систем</i>	5-6	Б1.О.26
5.	<i>Информационные системы и технологии</i>	5	Б1.О.25
6.	<i>Теория систем и системный анализ</i>	7	Б1.В.12
7.	<i>Ознакомительная практика</i>	4	Б2.О.01(У)
8.	<i>Технологическая (проектно-технологическая) практика</i>	6	Б2.О.02(П)
9.	<i>Разработка системы электронного документооборота</i>	7	Б1.В.10
10.	<i>Преддипломная практика</i>	8	Б2.В.01(Пд)

При освоении данной дисциплины необходимы умения и готовность («входные» знания) обучающегося по дисциплинам 1-2, указанных в Таблице 2. Дисциплины 3-10 относятся к группе, которые должны использовать «входные» знания данной дисциплины.

3. СТРУКТУРА И СОДЕРЖАНИЕ КУРСА, КРИТЕРИИ НАЧИСЛЕНИЯ БАЛЛОВ

Объем дисциплины составляет 8 зачетных единиц, всего 288 часов, из которых: лекции 30 часов, практические занятия 22 часа, лабораторные работы 30 часов, КСР – 22 часа, всего часов аудиторной нагрузки - 104 часа, самостоятельная работа – 76 часов, контроль – 108 часов.

Экзамен – 3- 4-й семестр

Семестр III

3.1 Структура и содержание теоретической части курса (16 ч.)

Тема 1. Введение в ОАЯП. Общие сведения о предмете основы алгоритмизации и языки моделирования. Связь дисциплины с другими предметами. (2 часа).

Тема 2. Объектно-ориентированные языки программирования. Принципы объектно-ориентированного программирования (ООП). Способы организация программы: код и данные. Общие свойства поддержки принципов ООП: инкапсуляция, полиморфизм, наследование. (2 часа).

Тема 3. Создание, компиляция и выполнение C++ - программ. Режимы работ компилятора языка: прямой режим, программный режим. Форматы программных строк. Исполнительные и неисполнительные операторы. (2 часа).

Тема 4. Основные типы данных. Логический тип. Символьный тип. Целочисленные типы. Типы с плавающей точкой. Размеры типов. Перечисления. Тип данных bool. Тип Void. Указатели и массивы. Литералы. (2 часа).

Тема 5. Использование операторов. Арифметические операторы. Инкремент и декремент. Операторы отношений и логические операторы. Оператор присваивания. (2 часа).

Тема 6. Инструкции управления. Инструкция if. Условное выражение. Вложенные if-инструкции. "Лестничная" конструкция if – else –if. Инструкция switch. Вложенные инструкции switch. Цикл for. (2 часа).

Тема 7. Массивы. Одномерные массивы. Двумерные массивы. Многомерные массивы. Строки. Считывание строк с клавиатуры. Некоторые

библиотечные функции обработки строк: функции `strcpy()`, `strcat()`, `strcmp()`, `strlen()`. (2 часа).

Тема 8. Строки и операции с ними. Строка - массив символов. Нуль-символ. Функции для работы со строками. Заголовочный файл `<cstring>`. Строки `string`. (2 часа).

3.2 Структура и содержание практической части курса (24ч)

Практические занятия (8 час.)

Практическое занятие № 1. Освоение способов организации программы: код и данные. (2 часа).

Практическое занятие № 2. Язык C++ - язык для Windows – программирования. Рассмотрение примера создания оконного приложения. (2 часа).

Практическое занятие № 3. Преобразование типов в операторах присваивания и в выражениях. (2 часа).

Практическое занятие № 4. Использование инструкции `break` для выхода из цикла. Использование инструкции `continue`. Вложенные циклы. Инструкция `goto`. (2 часа).

Лабораторные работы (16 час.)

Лабораторная работа № 1. Решение задач с вещественными числами (2 часа).

Лабораторная работа № 2. Решение задач с условием `if, if...else` (2 часа).

Лабораторная работа № 3. Решение задач с условием `while, do...while` (2 часа).

Лабораторная работа № 4. Решение задач с циклом `for` (2 часа).

Лабораторная работа № 5. Решение задач с использованием `case` – технологии (2 часа).

Лабораторная работа № 6. Решение задач с одномерными массивами (2 часа).

Лабораторная работа № 7. Решение задач с двумерными массивами (2 часа).

Лабораторная работа № 8. Решение задач с использованием строковых переменных (2 часа).

3.3 Структура и содержание КСР (8 час.)

Занятие № 1. Эволюция C++. Связь C++ с языками Java и C#. (2 часа).

Занятие № 2. Шестнадцатеричные и восьмеричные литералы. Строковые литералы. (2 часа).

Занятие № 3. Бесконечный цикл `for`. Циклы без тела. Цикл `while`. (2 часа).

Занятие № 4. Выполнение заданий с использованием функций для работы со строками. (2 часа).

Семестр IV

3.1 Структура и содержание теоретической части курса (14 ч.)

Тема 1. Строки и векторы. Стандартный класс string. Библиотека Cstring. Класс String. Ввод и вывод с помощью класса String. (2 часа).

Тема 2. Указатели и динамические массивы. Указатель - адрес переменной в памяти. Управление памятью. Объявление переменных указателей. (2 часа).

Тема 3. Работа с файлами. Файловый поток ввода и вывода. Функции использования файлового потока ввода и вывода. Файловый ввод-вывод. Директива `fstrem`. Классы `ifstream` и `ofstream`. Функции-члены `open`, `close`. (2 часа).

Тема 4. Структуры. Структуры как тип данных. (2 часа).

Тема 5. Введение в объектно-ориентированное программирование. Структуры и классы. Понятие классов. Класс как тип данных. Члены класса. (2 часа).

Тема 6. Наследование. Простое наследование. Производные классы. Конструкторы в производных классах. Модификатор `protected`. (2 часа).

Тема 7. Полиморфизм. Позднее связывание. Виртуальные функции языка C++ (2 часа).

3.2 Структура и содержание практической части курса (28 ч.)

Практические занятия (14 час.)

Практическое занятие № 1. Функция `getline`. Функции обработки строки и преобразования типов. (2 часа).

Практическое занятие № 2. Арифметика указателей. (2 часа).

Практическое занятие № 3. Локальные и глобальные переменные. Классы памяти. Встраиваемые функции. (2 часа).

Практическое занятие № 4. Функции `eof`, `fail` и `exit`. Добавление данных в файл. Форматирование выходных данных с помощью потоковых функций `setf`. (2 часа).

Практическое занятие № 5. Синтаксис структур. Инициализация структур. Сложные структуры (2 часа).

Практическое занятие № 6. Доступ к закрытым членам класса. Аксессоры и мутаторы. Основные свойства классов. Конструкторы классов. (2 часа).

Практическое занятие № 7. Ненаследуемые функции. Деструкторы и производные классы. (2 часа).

Лабораторные работы (14 час.)

Лабораторная работа № 1. Решение задач с использованием векторной переменной. (2 часа).

Лабораторная работа № 2. Решение задач с использованием указателей (2 часа).

Лабораторная работа № 3. Решение задач с использованием процедурной абстракции и механизмов передачи параметров. (2 часа).

Лабораторная работа № 4. Решение задач с использованием файлового ввода/вывода (2 часа).

Лабораторная работа № 5. Решение задач с использованием инициализации структур и структуры как параметра функции. (2 часа).

Лабораторная работа № 6. Решение задач с применением инкапсуляции (2 часа).

Лабораторная работа № 7. Решение задач с использованием переопределенной функции базового класса (2 часа).

3.3 Структура и содержание КСР (14 час.)

Занятие № 1. Основные функции для работы с векторами. (2 часа).

Занятие № 2. Создание и использования динамических массивов. (2 часа).

Занятие № 3. Обзор стандартных библиотечных функций. Процедурная абстракция. (2 часа).

Занятие № 4. Редактирование текстового файла. Директива `struct` и стандартные символьные функции. (2 часа).

Занятие № 5. Структуры как параметры функций. (2 часа).

Занятие № 6. Функции класса. (2 часа).

Занятие № 7. Множественное наследование. Переопределение функций-членов. (2 часа).

Таблица 3.

№ п/п	Раздел дисциплины	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)					Литература	Кол-во баллов в неделю
		Лек.	Пр.	Лаб.	КСР	СРС		
III семестр								
1.	Тема 1. Введение в ОАЯП. Общие сведения о предмете основы алгоритмизации и языка моделирования. Связь дисциплины с другими предметами.	2					1(5-10) 5 (с.11-16), 9	12,5
	Лабораторная работа № 1. Решение задач с использованием векторной переменной.			2		5	(с.31-49) 10	
2.	Практическое занятие № 1. Освоение способов организации программы: код и данные.		2				(с.19-33) 12,5	12,5
3.	Тема 2. Объектно-ориентированные языки программирования Принципы объектно-ориентированного программирования (ООП). Способы организация программы: код и данные. Общие свойства поддержки принципов ООП: инкапсуляция, полиморфизм, наследование. Объекты и классы.	2					5 (с.173-178), 9 (с.52-69) 10	12,5
	Лабораторная работа №2 Решение задач с условием <i>if, if....else</i>			2			(с.29-34) 2(8-18)	
4.	Эволюция C++. Связь C++ с языками Java и C #				2			12,5

5.	Тема 3. Создание, компиляция и выполнение С++ - программ Режимы работ компилятора языка: прямой режим, программный режим. Форматы программных строк. Исполнительные и неисполнительные операторы.	2				5	1(с.7-16) 5 (с.15-20), 10 (с.50-66)	12,5
	<i>Лабораторная работа № 3.</i> Решение задач с условием <i>while, do...while</i>			2				
6.	Практическое занятие № 2. Язык С++ - язык для Windows – программирования. Рассмотрение примера создания оконного приложения.		2					12,5
7.	Тема 4. Основные типы данных Логический тип. Символьный тип. Целочисленные типы. Типы с плавающей точкой. Размеры типов. Перечисления. Тип данных bool. Тип Void. Указатели и массивы. Литералы.	2				5	1(с.22-23) 5 (с.22-26), 10 (с.67-76)	12,5
	<i>Лабораторная работа № 4.</i> Решение задач с циклом <i>for</i>			2				
8.	Шестнадцатеричные и восьмеричные литералы. Строковые литералы.				2			12,5
9.	Тема 5. Использование операторов Арифметические операторы. Инкремент и декремент. Операторы отношений и логические операторы. Оператор присваивания.	2				5	1(с.25-27) 5 (с.27-37) 9 (с.54-70) 10 (с.67-76) 1(с.35-38)	12,5
	<i>Лабораторная работа № 5.</i> Решение задач с использованием <i>case</i> – технологии			2				
10.	Практическое занятие № 3. Преобразование типов в операторах присваивания и в выражениях.		2					12,5
11.	Тема 6. Инструкции управления Инструкция <i>if</i> . Условное выражение. Вложенные <i>if</i> -инструкции. "Лестничная" конструкция <i>if – else –if</i> . Инструкция <i>switch</i> . Вложенные инструкции <i>switch</i> . Цикл <i>for</i> .	2				5	1(с.30-34;38-45;49-57) 5 (с.40-49), 10 (с.310-366)	12,5
	<i>Лабораторная работа № 6.</i> Решение задач с одномерными массивами			2				
12.	Бесконечный цикл <i>for</i> . Циклы без тела. Цикл <i>while</i> .				2			12,5
13.	Тема 7. Массивы Одномерные массивы. Двумерные массивы. Многомерные массивы. Строки. Считывание строк с клавиатуры. Некоторые библиотечные функции обработки строк: функции <i>strcpy()</i> , <i>strcat()</i> , <i>strcmp()</i> , <i>strlen()</i> .	2				6	1(с.60-66) 5 (с.58-65) 10 (с.447-456)	12,5
	<i>Лабораторная работа № 7.</i> Решение задач с двумерными массивами.			2				

14.	Практическое занятие № 4. Использование инструкции break для выхода из цикла. Использование инструкции continue. Вложенные циклы. Инструкция goto.		2					12,5	
15.	Тема 8. Строки и операции с ними Строка - массив символов. Нуль-символ. Функции для работы со строками. Заголовочный файл <cstring>. Строки string.	2				6	1(с.71-80) 10 (с.515-531) 1(с.45-49)	12,5	
	Лабораторная работа № 8. Решение задач с использованием строковых переменных			2					
16.	Выполнение заданий с использованием функций для работы со строками.				2				12,5
ИТОГО: лек.-16 прак.-8 лаб. - 16 КСР-8 СРС-42 ВСЕГО-90		16	8	16	8	42		200	
IV семестр									
1.	Тема 9. Строки и векторы. Стандартный класс string. Библиотека Cstring. Класс String. Ввод и вывод с помощью класса String.	2				4	1(с.97-101) 10 (с.532-552) 11(с.231-256) 5(с.159-164) 1(с.57-60)	12,5	
	Лабораторная работа № 1. Решение задач с использованием векторной переменной.			2					
2.	Практическое занятие № 1. Функция GetLine. Функции обработки строки и преобразования типов.		2			5	5(с.51-64) 10 (с.556-577) 5(с.151)	12,5	
	Основные функции для работы с векторами.				2				
3.	Тема 10. Указатели и динамические массивы. Указатель - адрес переменной в памяти. Управление памятью. Объявление переменных указателей.	2				5	1(с.57-60)	12,5	
	Лабораторная работа № 2. Решение задач с использованием указателей				2				
4.	Создание и использования динамических массивов				2			12,5	
	Практическое занятие № 2. Арифметика указателей.		2						

5.	Тема 11. Работа с файлами. Файловый поток ввода и вывода. Функции использования файлового потока ввода и вывода. Файловый ввод-вывод. Директива <code>fstrem</code> . Классы <code>ifstream</code> и <code>ofstream</code> . Функции-члены <code>open</code> , <code>close</code> . Функции <code>eof</code> , <code>fail</code> и <code>exit</code> .	2				5	4(57-65) 5(82-83) 10(598-625)	12,5
	<i>Лабораторная работа № 3.</i> Решение задач с использованием процедурной абстракции и механизмов передачи параметров.			2				
6.	<i>Практическое занятие № 3.</i> Локальные и глобальные переменные. Классы памяти. Встраиваемые функции.		2			5	4(66-67)	12,5
	Обзор стандартных библиотечных функций. Процедурная абстракция.				2		1(с.10-1-103)	
7.	Тема 12. Структуры Структуры как тип данных. Синтаксис структур. Инициализация структур. Сложные структуры. Структуры как параметры функций.	2				5	4(с.85-101) 10 (с.192-228)	12,5
	<i>Лабораторная работа № 4.</i> Решение задач с использованием файлового ввода/вывода			2				
8.	<i>Практическое занятие № 4.</i> Функции <code>eof</code> , <code>fail</code> и <code>exit</code> . Добавление данных в файл. Форматирование выходных данных с помощью потоковых функций <code>setf</code> .		2			5	4(85-99) 4(99-101)	12,5
	Редактирование текстового файла. Директива <code>setf</code> и стандартные символьные функции				2			
9.	Тема 13. Введение в объектно-ориентированное программирование Структуры и классы. Понятие классов. Класс как тип данных. Члены класса. Открытые и закрытые члены класса (<code>public</code> , <code>private</code>). Функции класса. Инкапсуляция. Доступ к закрытым членам класса. Аксессоры и мутаторы. Основные свойства классов. Конструкторы классов.	2				5	1(с.108-108) 5(с.67-69)	12,5
	<i>Лабораторная работа № 5.</i> Решение задач с использованием инициализации структур и структуры как параметра функции.			2				
10.	<i>Практическое занятие № 5.</i> Синтаксис структур. Инициализация структур. Сложные структуры		2			5	5(с.144-147) 1(с.112-120)	12,5
	Структуры как параметры функций				2			
11.	Тема 14. Наследование. Простое наследование. Производные классы. Конструкторы в производных классах. Модификатор <code>protected</code> . Множественное наследование. Переопределение функций-членов. Доступ к переопределенной функции базового класса. Ненаследуемые функции. Деструкторы и производные классы.	2				5	5(с.178-188) 10 (с.284-290)	12,5

	<i>Лабораторная работа № 6.</i> Решение задач с применением инкапсуляции			2				
12.	Практическое занятие № 6. Доступ к закрытым членам класса. Аксессуары и мутаторы. Основные свойства классов. Конструкторы классов		2				5(с.24-8-251)	12,5
	Функции класса				2			
13.	Тема 15. Полиморфизм. Позднее связывание. Виртуальные функции языка С++	2				5	5(с.20-0-210) 10 (с.683-704)	12,5
	<i>Лабораторная работа № 7.</i> Решение задач с использованием переопределенной функции базового класса			2				
14.	<i>Практическое занятие № 7.</i> Ненаследуемые функции. Деструкторы и производные классы		2					5(с.25-2-259)
	Множественное наследование. Переопределение функций-членов				2			
ИТОГО: лек.-14 прак.-14 лаб. - 14 КСР-14 СРС-34 ВСЕГО-94		14	14	14	14	34		200
Всего по курсу		30	22	30	22	76		400

Формы контроля и критерии начисления баллов

В случае пропуска студентом занятий по уважительной причине (при наличии подтверждающего документа) в период академической недели, деканат факультета обращается к проректору по учебной работе с представлением об отработке студентом баллов за пропущенные дни по каждой отдельной дисциплине с последующим внесением их в электронный журнал.

Итоговая форма контроля по дисциплине экзамен проводится в форме тестирования. Тестовая форма итогового контроля по дисциплине предусматривает: для естественнонаучных направлений – 10 тестовых вопросов на одного студента, где правильный ответ оценивается в 10 баллов. Тестирование проводится в электронном виде.

Таблица 4.

Неделя	Активное участие на лекционных занятиях, написание конспекта и выполнение других видов работ*	Активное участие на практических (семинарских) занятиях, КСР	СРС Написание реферата, доклада, эссе Выполнение других видов работ	Административный балл за примерное поведение	Балл за рубежный и итоговый контроль	Всего
1	2	3	4	5	6	7
1	3,5	5	4	-	-	12,5
2	3,5	5	4	-	-	12,5

3	3,5	5	4	-	-	12,5
4	3,5	5	4	-	-	12,5
5	3,5	5	4	-	-	12,5
6	3,5	5	4	-	-	12,5
7	3,5	5	4	-	-	12,5
8	3,5	5	4	-	-	12,5
9(первый рубежный контроль)					-	-
Первый рейтинг				-	-	100/2*0.49
10	3,5	5	4	-	-	12,5
11	3,5	5	4	-	-	12,5
12	3,5	5	4	-	-	12,5
13	3,5	5	4	-	-	12,5
14	3,5	5	4	-	-	12,5
15	3,5	5	4	-	-	12,5
16	3,5	5	4	-	-	12,5
17	3,5	5	4	-	-	12,5
18 (второй рубежный контроль)					-	-
Второй рейтинг					-	100/2*0.49
ИТОГОВЫЙ КОНТРОЛЬ (зачет, зачет с оценкой, экзамен)					100	100/2*0.51
ИТОГО:	56	80	64		16+100	300/100

4. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Основы алгоритмизации и языки программирования» включает в себя:

1. план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;
2. характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;
3. требования к представлению и оформлению результатов самостоятельной работы;
4. критерии оценки выполнения самостоятельной работы.

**4.1. План-график выполнения самостоятельной работы по дисциплине
«Основы алгоритмизации и языки программирования»**

Таблица 5.

№ п/п	Объем СРС в ч.	Тема самостоятельной работы	Форма и вид результатов самостоятельной работы	Форма контроля
1	2	Связь дисциплины с другими предметами. Модели и их виды	Конспект, реферат	Контроль самостоятельной работы. Устный опрос.
2	2	Алгоритмические языки высокого уровня. Отличительные черты языков программирования	Конспект, реферат	Контроль самостоятельной работы. Устный опрос.
3	3	Объектно-ориентированные языки программирования. Основные принципы ООП	Презентация, доклад	Контроль самостоятельной работы. Устный опрос.
4	2	Основы языка C++. Эволюция C++. Связь C++ с языками Java и C #.	Конспект, презентация	Контроль самостоятельной работы. Устный опрос.
5	3	Создание, компиляция и выполнение C++ - программ	Выполнение и отладка программы задания 1 (1-10). Подготовка отчета	Контроль самостоятельной работы. Отчеты по практическим работам.
6	2	Основные типы данных	Выполнение и отладка программы задания 1 (11-20). Подготовка отчета	Контроль самостоятельной работы. Отчеты по практическим работам.
7	2	Использование операторов	Выполнение и отладка программы задания 1 (21-30). Подготовка отчета	Контроль самостоятельной работы. Отчеты по практическим работам.
8	2	Инструкция if. Условное выражение. Вложенные if-инструкции.	Выполнение и отладка программы задания 2 (1-30). Подготовка отчета	Контроль самостоятельной работы. Отчеты по практическим работам.
9	2	Инструкция switch. Вложенные ин-	Выполнение и от-	Контроль са-

		струкции switch	ладка программы задания 3 (1-20). Подготовка отчета	мостоятельной работы. Отчеты по практическим работам.
10	3	Цикл for. Объявление управляющей переменной цикла в заголовке инструкции for.	Выполнение и отладка программы задания 4 (1-12). Подготовка отчета	Контроль самостоятельной работы. Отчеты по практическим работам.
11	2	Цикл while (do while).	Выполнение и отладка программы задания 4 (13-35). Подготовка отчета.	Контроль самостоятельной работы. Отчеты по практическим работам.
12	3	Массивы (одномерные, двумерные и многомерные).	Выполнение и отладка программы задания 5 (1-25). Подготовка отчета.	Контроль самостоятельной работы. Отчеты по практическим работам.
13	3	Строки, функции обработки строк	Выполнение и отладка программы задания 6 (1-25). Подготовить отчет.	Контроль самостоятельной работы. Отчеты по практическим работам.
14	3	Указатели. Операторы, используемые с указателями.	Выполнение и отладка программы задания 6 с указателями. Подготовить отчет.	Контроль самостоятельной работы. Отчеты по практическим работам.
15	2	Класс String.	Выполнение и отладка программы задания 7. Подготовить отчет.	Контроль самостоятельной работы. Отчеты по практическим работам.
16	3	Векторы. Объявление векторной переменной. Основные функции для работы с векторами.	Выполнение и отладка программы задания 8 (1-14) с помощью векторов. Подготовить отчет.	Контроль самостоятельной работы. Отчеты по практическим работам.
17	3	Создание и использования динамических массивов.	Выполнение и отладка программы задания 8 (1-14) с помощью динамических массивов. Под-	Контроль самостоятельной работы. Отчеты по практическим

			готовить отчет.	работам.
18	2	Параметры и аргументы функции. Прототип функции. Вызов функции. Механизм передачи параметров.	Выполнение и отладка программы задания 9 (1-13). Подготовить отчет.	Контроль самостоятельной работы. Отчеты по практическим работам.
19	3	Файловый ввод-вывод. Директива <code>fstream</code> . Форматирование выходных данных. Символьный ввод-вывод. Редактирование текстового файла.	Выполнить примеры задания 6 и 7, при условии, что используется файловый поток ввода и вывода. Подготовить отчет.	Контроль самостоятельной работы. Отчеты по практическим работам.
20	2	Потоки в качестве аргументов функций.	Выполнить программы задания 9 передавая потоки (файловые) функции. Подготовить отчет.	Контроль самостоятельной работы. Отчеты по практическим работам.
21	2	Структуры как тип данных. Синтаксис структур. Инициализация структур. Сложные структуры. Структуры как параметры функций	Выполнение и отладка программы задания 10 (1-25). Подготовить отчет.	Контроль самостоятельной работы. Отчеты по практическим работам.
22	3	Понятие классов. Конструкторы и деструкторы классов. Открытые и закрытые члены класса.	Выполнение и отладка программы задания 11 (1-25). Подготовить отчет и презентацию.	Контроль самостоятельной работы. Отчеты по практическим работам.
23	2	Наследование. Конструкторы и деструкторы производных классов.	Выполнение и отладка программы задания 11 (1-25). Подготовить отчет и презентацию.	Контроль самостоятельной работы. Отчеты по практическим работам.

4.2 Характеристика заданий для самостоятельной работы и методические рекомендации по их выполнению

Для выполнения задания, прежде всего, необходимо ознакомиться и изучить основные положения теоретических материалов соответствующей темы из литературных источников. Они указаны в разделе 3 «Содержание и структура дисциплины». Большинство заданий выполняются в виде разработки программы на языке C++ и оформления отчета. Отчет содержит следующие разделы:

1. Стандартный титульный лист
2. Отчет о решенных задачах (с указанным номером варианта):
 - постановку задачи;
 - структурную (обобщенную) схему алгоритма;

- листинг (текст кода) программы с пояснением его отдельных участков;
- результаты (экранные формы), иллюстрирующие работоспособность программы.

Пример оформления отчета

ОТЧЕТ

выполненной работы № задания по дисциплине
Основы алгоритмизации и языки программирования

Фамилия Имя

Группа

Постановка задачи

Задание № варианта

С помощью генератора случайных чисел (`rand()`) образовать вещественный

массив a_0, a_1, \dots, a_n , причем $a_i \in [x_1, x_2]$, $i=1, 2, \dots, n$

1. Найти максимальный элемент.
2. Найти минимальный элемент.
3. Найти сумму элементов.
4. Найти количество нулевых элементов.
5. Образовать новый вещественный массив, элементы которого представляют собой среднюю арифметическую двух рядом идущих элементов исходного массива.
6. Упорядочить.
7. Построить зигзагообразный массив. Массив называется зигзагообразным, если $a_0 \leq (\geq) a_1 \geq (\leq) a_2 \leq (\geq) a_3 \geq (\leq) a_4 \dots$. Имеется множество вариантов построения зигзагообразного массива. Реализовать вариант «равномерного затухания»

Листинг (текст кода) программы

```
int main(int argc, char *argv[])
{
    int n,m,i,j,k=0;
    cout<<"n=";   cin>>n;
    float a[n],a1[n],x1,x2,b,max,min,sum=0.,teta;
    cout<<"x1=";   cin>>x1;   cout<<"x2=";   cin>>x2;
    cout<<"Массив a[i]:"<<endl;
    max=x1;   min=x2;
    for(i=0;i<n;i++)
    {
        m=n*(int)(x2-x1);   teta=(float)(rand()%m);   teta=teta/(float) m;   a[i]=(x2-
x1)*teta+x1;
        if(a[i]>max)   max=a[i];   if(a[i]<min)   min=a[i];   sum+=a[i];
        cout<<a[i];
        (i<n-1)?   cout<<" ";:cout<<endl;
    }
    cout<<"max="<<max<<"\t"<<"   min="<<min<<"\t"<<"Сумма="<<sum<<endl;
    cout<<"New massiv (a[i]+a[i+1])/2"<<endl;
```

```

for(i=0;i<n-1;i++)
{
    cout<<(a[i]+a[i+1])/2;    (i<n-2)? cout<<" "; :cout<<endl;
}
for(i=0;i<n-1;i++)
{
    for(j=i+1;j<n;j++)
    {
        if(a[i]>a[j])
        {
            b=a[i]; a[i]=a[j]; a[j]=b;
        }
    }
}
cout<<"Отсортированный массив: "<<endl;
for(i=0;i<n;i++)
{
    cout<<a[i];
    (i<n-1)? cout<<" "; :cout<<endl;
}
m=(n+1)/2;
cout<<"Зигзаг : "<<endl;
for(i=0;i<m;i++)
{
    a1[k]=a[i]; k++; a1[k]=a[n-(i+1)]; k++;
}
for(int i=0;i<n;i++)
{
    cout<<a1[i];    (i<n-1)? cout<<" "; :cout<<endl;
}
cout<<endl;
system("PAUSE");
return EXIT_SUCCESS;
}

```

Результаты, иллюстрирующие работоспособность программы.

Результаты работы программы для интервала [1,20], n=10

```

D:\CodeBlock\R\zigzag\bin\Debug\zigzag.exe
n=10
x1=1
x2=20
Array a[i]:
5.1; 4.7; 7.4; 10; 17.9; 15.4; 8.8; 10.8; 18.2; 15.4
max=18.2      min=4.7      Summa=113.7
New array (a[i]+a[i+1])/2
4.9; 6.05; 8.7; 13.95; 16.65; 12.1; 9.8; 14.5; 16.8
Sorted array:
4.7; 5.1; 7.4; 8.8; 10; 10.8; 15.4; 15.4; 17.9; 18.2
Zigzag :
4.7; 18.2; 5.1; 17.9; 7.4; 15.4; 8.8; 15.4; 10; 10.8

```

Рис.2. Результат работы программы

Блок-схема (обобщенная) алгоритма

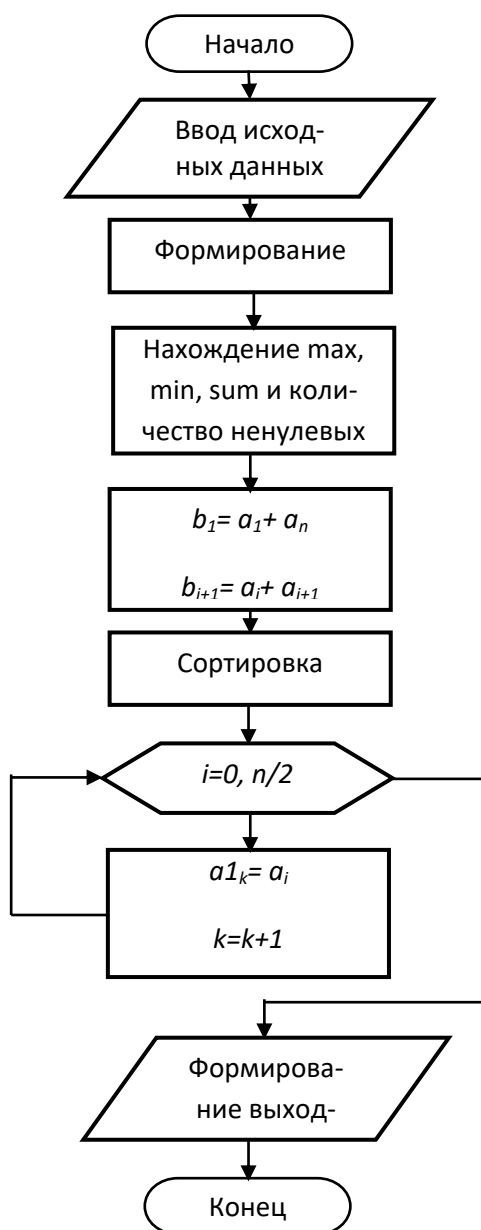


Рис. 1. Блок-схема алгоритма

4.3. Требования к представлению и оформлению результатов самостоятельной работы;

Индивидуальные домашние задания по самостоятельной работе должны быть выполнены в отдельной тетрадке. В каждом задании должны быть приведены постановка задачи и описана последовательность ее решения. В конце решения задачи приводятся результаты выполненной работы.

При выполнении самостоятельной работы студент должен предварительно изучить методы решения задач данного типа и правильно выбрать соответствующий метод ее решения.

По лабораторным работам студенты должны представить отчеты в соответствии с содержанием, приведенным в пункте 4.2, которые должны быть защищены у преподавателя. На защите лабораторных работ студентам зада-

ется один теоретический вопрос и задача, которые он должен самостоятельно подготовить и решить.

4.4. Критерии оценки выполнения самостоятельной работы.

Самостоятельная работа прививает студентам навыки работы с источниками и учебной литературой, помогает повысить уровень знаний по предмету, а также самостоятельно анализировать и проектировать электронные документы для систем электронного документооборота, которые можно использовать на практике.

Оценка «отлично» выставляется студенту, если индивидуальное задание выполнено полностью и по данной теме защищена лабораторная работа.

Оценка «хорошо» выставляется студенту, если лабораторная работа по теме индивидуального задания защищена, а само индивидуальное задание выполнено с отдельными замечаниями.

Оценка «удовлетворительно» выставляется студенту, если лабораторная работа по теме индивидуального задания защищена, а само индивидуальное задание выполнено не до конца, т.е. не полностью.

Оценка «неудовлетворительно» выставляется студенту, если лабораторная работа по теме индивидуального задания не защищена, а само индивидуальное задание выполнено не до конца, т.е. не полностью.

5. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

5.1 Основная литература:

1. Умаров М.А., Касымова М.Д. Основы алгоритмизации и языки программирования (уч. пособие). Душанбе, РТСУ. 2014. -117с
2. Умаров М.А., Касымова М.Д. Сборник задач по программированию. Душанбе, РТСУ. 2013. -102с.
3. Касымова М.Д., Умаров М.А. Методы вычислений и программирование. (Учебное пособие). Душанбе, РТСУ. 2012. -120с.
4. Замонов М.З., Нуров И.Д., Умаров М.А. Нестандартные задачи по программированию. Душанбе, РТСУ. 2012.-110с.

5.2 Дополнительная литература

1. Огнева М. В., Кудрина Е. В. Программирование на языке C++: практический курс. – Москва.:Юрайт, 2022 - 335с.
2. Кувшинов Д. Р. Основы программирования. - Москва.:Юрайт, 2022 - 105с.
3. Трофимов В. В., Павловская Т. А. Основы алгоритмизации и программирования. - Москва.:Юрайт, 2022 - 137с.
4. Трофимов В. В., Павловская Т. А. Алгоритмизация и программирование. Москва.:Юрайт, 2022 - 137с.
5. Савич У. Программирование на C++, 4-е изд. – СПб.: Питер, 2020
6. Страуструп Б. Язык программирования C++. 7-ое издание. - М.: БИНОМ, 2021.
7. Умаров М.А. Методические рекомендации по выполнению курсовой

работы по дисциплине ОАЯМ. Душанбе: - РТСУ, 2011.

8. Шилдт Т. С++ для начинающих. М.: ЭКОМ Паблишерс, 2017.- 640с.

5.3

5.4 Нормативно-правовые материалы (по мере необходимости)

5.5 Перечень ресурсов информационно-телекоммуникационной сети

Интернет

1. [http:// www.citforum.ru](http://www.citforum.ru) – материалы сайта Сервер информационных технологий.
2. <http://www.makasin.info/system/files>

5.5 Перечень информационных технологий и программного обеспечения

Используется программное обеспечение открытого доступа (Open source) - CodeBlock, Dev_C++, Denwer.

6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Студенты, изучающие курс «Основы алгоритмизации и языки программирования», должны в первую очередь обратить внимание на принципиальную разницу между языками программирования высокого уровня и объектно-ориентированного программирования. Для свободной ориентации в информационном пространстве современного общества специалист любого профиля должен уметь получать, грамотно обрабатывать и использовать информацию с помощью средств вычислительной техники и телекоммуникаций.

Студенты должны знать основы языков программирования и их отличительные черты; Алгоритмы и способы представления алгоритмов, основные характеристики и функциональные возможности языков программирования; этапы программирования, основные процессы и подходы к разработке программных продуктов.

Студенты должны уметь проводить анализ и выбирать компоненты аппаратного обеспечения для решения профессиональных задач; применять программные средства для решения экономических задач; использовать современные технологии разработки приложений операционных систем для автоматизации работы с офисными программами.

Студенты должны иметь представление об истории и перспективах развития рынка компьютерных продуктов; об основных понятиях объектно-ориентированного подхода к созданию программных продуктов; о характеристиках основных программных продуктов.

Общую схему изучения предмета «Основы алгоритмизации и языки программирования» можно представить в следующем виде:

- Приобретение необходимых знаний по разработке математической модели задачи с помощью построения алгоритмов.
- Приобретение необходимых знаний и навыков по программированию, созданию и ведению баз данных, в том числе и распределенных.
- Приобретение необходимых знаний и навыков по программиро-

ванию и внедрению современных информационных технологий в конкретную область.

Основа для изучения дисциплины «Основы алгоритмизации и языки программирования» - конспекты лекций, результаты лабораторных занятий и выполненные самостоятельные работы самими студентами.

На лабораторных занятиях с использованием средств вычислительной техники студенты выполняют задания, предусмотренные для приобретения пользовательских навыков, решают задачи вычислительного характера, устанавливают и настраивают программные продукты, разрабатывают алгоритмы и программы для решения прикладных задач, выполняют типовые расчеты. Во время самостоятельной работы студента с преподавателем обсуждаются проблемные лекции, решаются сложные алгоритмы.

Все это может дать положительный результат, если студент активно занимается самостоятельной работой в соответствии с планом-графиком п.4.1. Процесс выполнения СРС с описанием этапов решения примерной задачи описан в п.4.2

Вместе с тем основой обучения являются аудиторные занятия - лекции, практические занятия и лабораторные работы по рассмотрению проблем информационной технологии и решению конкретных задач программирования.

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для реализации дисциплины при кафедре информатики и ИТ РТСУ имеются 4 компьютерных класса, с наличием необходимых программных продуктов: используется программное обеспечение открытого доступа (Open source) - CodeBlock, Dev_C++, Denwer.

В Университете созданы специальные условия обучающихся с ограниченными возможностями здоровья - специальные учебники, учебные пособия и дидактические материалы, специальные технические средства обучения коллективного и индивидуального пользования, предоставление услуг ассистента (помощника), оказывающего обучающимся необходимую техническую помощь, проведение групповых и индивидуальных коррекционных занятий, обеспечение доступа в здания организаций и другие условия, без которых невозможно или затруднено освоение дисциплины обучающимися с ограниченными возможностями здоровья.

Обучающимся с ограниченными возможностями здоровья предоставляются бесплатно специальные учебники и учебные пособия, иная учебная литература, а также обеспечивается:

наличие альтернативной версии официального сайта организации в сети "Интернет" для слабовидящих;

присутствие ассистента, оказывающего обучающемуся необходимую помощь;

обеспечение выпуска альтернативных форматов печатных материалов (крупный шрифт или аудиофайлы);

возможность беспрепятственного доступа обучающихся в учебные помещения, столовые, туалетные и другие помещения организации, а также пребывания в указанных помещениях (наличие пандусов, поручней, расширенных дверных проёмов, лифтов).

8. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Форма итоговой аттестации – экзамен.

Форма промежуточной аттестации - нет

Итоговая система оценок по кредитно-рейтинговой системе с использованием буквенных символов

Оценка по буквенной системе	Диапазон соответствующих наборных баллов	Численное выражение оценочного балла	Оценка по традиционной системе
A	10	95-100	Отлично
A-	9	90-94	
B+	8	85-89	Хорошо
B	7	80-84	
B-	6	75-79	
C+	5	70-74	Удовлетворительно
C	4	65-69	
C-	3	60-64	
D+	2	55-59	
D	1	50-54	
Fx	0	45-49	Неудовлетворительно
F	0	0-44	

Содержание текущего контроля, промежуточной аттестации, итогового контроля раскрываются в фонде оценочных средств, предназначенных для проверки соответствия уровня подготовки по дисциплине требованиям ФГОС ВО.

ФОС по дисциплине является логическим продолжением рабочей программы учебной дисциплины. ФОС по дисциплине прилагается.