

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»**

Радиофизический факультет

УТВЕРЖДЕНО

решением президиума Ученого совета ННГУ

протокол № 1 от 16.01.2024 г.

Рабочая программа дисциплины

Информатика

Уровень высшего образования

Специалитет

Направление подготовки / специальность

11.05.02 - Специальные радиотехнические системы

Направленность образовательной программы

Радиотехнические системы и комплексы сбора и обработки информации

Форма обучения

очная

г. Нижний Новгород

2024 год начала подготовки

1. Место дисциплины в структуре ОПОП

Дисциплина Б1.О.18 Информатика относится к обязательной части образовательной программы.

2. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине (модулю), в соответствии с индикатором достижения компетенции		Наименование оценочного средства	
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	Для текущего контроля успеваемости	Для промежуточной аттестации
ОПК-2: Способен использовать языки и системы программирования, программные средства общего назначения, инструментальные средства компьютерного моделирования для решения различных исследовательских и профессиональных задач	<p>ОПК-2.1: Разбирается в основных понятиях информатики, основах программирования</p> <p>ОПК-2.2: Понимает технологию работы на компьютере в современных операционных средах, основные методы разработки алгоритмов и программ</p> <p>ОПК-2.3: Использует методы и средства компьютерной графики</p> <p>ОПК-2.4: Применяет технологию работы на компьютере и методы разработки алгоритмов и программ для проектирования радиоэлектронных средств</p> <p>ОПК-2.5: Применяет действующие стандарты, положения по оформлению технической документации при помощи средств ПЭВМ</p>	<p>ОПК-2.1: Знает основные понятия информатики и программирования; основы классификации и построения современных компьютерных систем</p> <p>ОПК-2.2: Знает основные алгоритмические конструкции и способы их описания; основы алгоритмизации и принципы использования алгоритмических языков программирования для постановки и решения прикладных задач на ЭВМ</p> <p>ОПК-2.3: Умеет визуализировать этапы решения задач</p> <p>ОПК-2.4: Умеет составить алгоритм построения радиофизической программной модели. Осуществляет создание и трансляцию консольных программ с использованием языков программирования C / C++</p> <p>ОПК-2.5:</p>	Собеседование	<p>Зачёт: Практическое задание</p> <p>Экзамен: Практическое задание</p>

		Сопровождает написанный код пояснениями и комментариями		
ОПК-3: Способен понимать сущность и значение информации в развитии современного общества, осознавать опасности и угрозы, возникающие в этом процессе, соблюдать требования нормативных правовых актов в области информационной безопасности в своей профессиональной деятельности	ОПК-3.1: Понимает сущность и значение информации в развитии современного общества, анализирует современные информационные технологии и правовые акты по информационной безопасности ОПК-3.2: Анализирует основные правовые акты и осуществляет правовую оценку информации, используемых в профессиональной деятельности, предпринимает необходимые меры по восстановлению нарушенных прав	ОПК-3.1: Знать: - основные стандарты современных информационных технологий и языков программирования ОПК-3.2: Может объяснить сущность и значение информации в развитии современного общества, проанализировать современные информационные технологии и правовые акты по информационной безопасности.	Собеседование	Зачёт: Практическое задание Экзамен: Практическое задание
ОПК-4: Способен учитывать современные тенденции развития компьютерных, информационных и телекоммуникационных технологий в своей профессиональной деятельности	ОПК-4.1: Анализирует информационные технологии и информационно-вычислительные системы ОПК-4.2: Применяет информационные технологии и информационно-вычислительные системы для решения научно-исследовательских и проектных задач радиоэлектроники	ОПК-4.1: Знать: - области и особенности применения языков программирования высокого уровня ОПК-4.2: Уметь: - разрабатывать и реализовывать на языке высокого уровня алгоритмы решения типовых профессиональных задач	Собеседование	Зачёт: Практическое задание Экзамен: Практическое задание
ОПК-6: Способен понимать принципы работы информационных технологий, применять основные методы и средства получения, хранения, обработки информации в сфере	ОПК-6.1: Понимает основные методы и средства получения, хранения, обработки информации ОПК-6.2: Применяет основные методы и средства получения, хранения, обработки информации в сфере профессиональной	ОПК-6.1: Может организовать хранение информации и обеспечить ее доступность и целостность. ОПК-6.2: Владеет основным типовым инструментарием для получения, хранения,	Собеседование	Зачёт: Практическое задание Экзамен: Практическое задание

профессиональной деятельности	деятельности	обработки информации в сфере профессиональной деятельности.		
-------------------------------	--------------	---	--	--

3. Структура и содержание дисциплины

3.1 Трудоемкость дисциплины

	очная
Общая трудоемкость, з.е.	6
Часов по учебному плану	216
в том числе	
аудиторные занятия (контактная работа):	
- занятия лекционного типа	32
- занятия семинарского типа (практические занятия / лабораторные работы)	64
- КСР	3
самостоятельная работа	72
Промежуточная аттестация	45 Экзамен, Зачёт

3.2. Содержание дисциплины

(структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий)

Наименование разделов и тем дисциплины	Всего (часы)	в том числе			
		Контактная работа (работа во взаимодействии с преподавателем), часы из них			Самостоятельная работа обучающегося, часы
		Занятия лекционного типа	Занятия семинарского типа (практические занятия/лабораторные работы), часы	Всего	
0 0	0 0	0 0	0 0	0 0	
1.1. Введение в дисциплину «Информатика»	2	1		1	1
1.2. Основы работы в ОС Windows и ОС Linux	4	1	2	3	1
1.3. Введение в язык Си. Структура программы на языке Си. Примеры простых программ	5	1	2	3	2
1.4. Основные этапы трансляции программы на языке Си. Задачи, решаемые препроцессором, компилятором, компоновщиком	3	1		1	2
1.5. Типы данных в языке Си: базовые, производные и составные типы данных	3	1		1	2
1.6. Константы и переменные в языке Си	3	1		1	2
1.7. Представление данных в ЭВМ. Двоичное представление целочисленных данных	4	1	1	2	2

1.8. Двоичное представление вещественных данных. Двоичное представление символической информации	4	1	1	2	2
1.9. Операции в языке Си	7	1	4	5	2
1.10. Операторы языка Си	7	1	4	5	2
1.11. Функции в языке Си	9	1	4	5	4
1.12. Массивы в языке Си	7	1	4	5	2
1.13. Указатели в языке Си. Работа со строками	7	1	4	5	2
1.14. Структуры в языке Си. Средство описания типов typedef. Объединения в языке Си	4	1	2	3	1
1.15. Средства динамического распределения памяти	4	1	2	3	1
1.16. Обзор библиотечных функций языка Си	5	1	2	3	2
2.1. Основы создания программ на языке C++. Компиляция. Директива include. Комментарии. Простейший ввод и вывод. Объявление переменных. Объявление простых функций и их использование	5	1	2	3	2
5 препроцессором, компилятором, компоновщиком. 1.5. Типы данных в языке Си: базовые, производные и составные типы данных. 2 1 1 1.6. Константы и переменные в языке Си. 2 1 1 1.7. Представление данных в ЭВМ. Двоичное представление целочисленных данных. 3 1 1 2 1.8. Двоичное представление вещественных данных. Двоичное представление символической информации. 3 1 1 2 1.9. Операции в языке Си. 7 1 4 5 2 1.10. Операторы языка Си. 7 1 4 5 2 1.11. Функции в языке Си. 8 1 4 5 3 1.12. Массивы в языке Си. 7 1 4 5 2 1.13. Указатели в языке Си. Работа со строками. 7 1 4 5 2 1.14. Структуры в языке Си. Средство описания типов typedef. Объединения в языке Си. 4 1 2 3 1 1.15. Средства динамического распределения памяти. 4 1 2 3 1 1.16. Обзор библиотечных функций языка Си. 5 1 2 3 2 2.1. Основы создания программ на языке C++. Компиляция. Директива include. Комментарии. Простейший ввод и вывод. Объявление переменных. Объявление простых функций и их использование. 5 1 2 3 2 2.2. Типы данных в C++, преобразование типов, арифметические операции	5	1	2	3	2
2.3. Массивы. Строки в стиле C. Строки класса string. Указатели. Управление динамической памятью с помощью new и delete	6	1	2	3	3
2.4. Структуры в C++	6	1	2	3	3
2.5. Циклы for, while, do while	6	1	2	3	3
2.6. Ссылочные переменные. Передача функции аргументов по ссылке. 6 Аргументы по умолчанию.	10	2	4	6	4
2.7. Перегрузка функций. Шаблоны функций. Спецификации шаблонов функций.	11	2	4	6	5
2.8. Раздельная компиляция программ. Продолжительность хранения, область видимости и компоновка.	8	1	2	3	5
2.9. Определение и реализация класса. Открытый и закрытый доступ к классу. Данные- члены класса. Методы класса. Создание и использование объектов класса. Конструкторы и деструкторы класса. Функции-члены const. Указатель this. Создание массивов объектов	11	2	4	6	5
2.10. Перегрузка операций. Дружественные функции. Перегрузка операции для вывода	11	2	4	6	5
2.11. Динамическое выделение памяти для членов класса. Явные и неявные конструкторы копирования. Явные и неявные перегруженные операции присваивания. Статические члены класса. Указатели на объекты	11	2	4	6	5
Аттестация	45				
КСР	3			3	
Итого	216	32	64	99	72

Содержание разделов и тем дисциплины

Системы контроля версий
Базовые алгоритмы и структуры данных
Парадигмы программирования
Основы теории тестирования
Жизненные циклы ПО

4. Учебно-методическое обеспечение самостоятельной работы обучающихся

Самостоятельная работа обучающихся включает в себя подготовку к контрольным вопросам и заданиям для текущего контроля и промежуточной аттестации по итогам освоения дисциплины приведенным в п. 5.

- изучение дополнительных разделов дисциплины с использованием учебной литературы;
- изучение и тестирование библиотек языков программирования.

5. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

5.1 Типовые задания, необходимые для оценки результатов обучения при проведении текущего контроля успеваемости с указанием критериев их оценивания:

5.1.1 Типовые задания (оценочное средство - Собеседование) для оценки сформированности компетенции ОПК-2:

Контрольные вопросы:

1. Команды для работы с файловой системой в командной строке Unix. Примеры использования.
2. Команды для работы с файлами в командной строке Unix. Примеры использования.
3. Команды для работы с файловой системой в командной строке Windows. Примеры использования.
4. Команды для работы с файлами в командной строке Windows. Примеры использования.

5.1.2 Типовые задания (оценочное средство - Собеседование) для оценки сформированности компетенции ОПК-3:

1. Объявление переменных в языке программирования Си. Способы инициализации объявленных переменных.
2. Использование квалификаторов при объявлении переменных.
3. Время жизни и область видимости переменных в языке программирования Си. Статические переменные. Локальные и глобальные переменные.

4. Константы в языке программирования Си, способы задания констант. Целочисленные (десятичные, восьмеричные, шестнадцатеричные) и действительные константы. Именованные константы (директива #define).
5. Массивы в языке программирования Си. Объявление и использование одномерных массивов.
6. Представление строк в языке программирования Си как массивов символов.
7. Типы данных, преобразование типов, арифметические операции.
8. Массивы. Строки в стиле C. Строки класса string. Указатели. Управление динамической памятью с помощью new и delete.

5.1.3 Типовые задания (оценочное средство - Собеседование) для оценки сформированности компетенции ОПК-4:

1. Команды для работы с файловой системой в командной строке Unix. Примеры использования.
2. Команды для работы с файлами в командной строке Unix. Примеры использования.
3. Команды для работы с файловой системой в командной строке Windows. Примеры использования.
4. Команды для работы с файлами в командной строке Windows. Примеры использования.
5. Основные спецификации форматного ввода / вывода (для функций printf и scanf) в библиотеке стандартного ввода / вывода языка программирования Си.
6. Основные функции, используемые при работе с текстовыми файлами (открытие, закрытие, чтение, запись и пр.) в библиотеке стандартного ввода / вывода языка программирования Си.
7. Раздельная компиляция программ. Продолжительность хранения, область видимости и компоновка.
8. Определение и реализация класса. Открытый и закрытый доступ к классу. Данные – члены класса. Методы класса. Создание и использование объектов класса. Конструкторы и деструкторы класса. Функции – члены const. Указатель this. Создание массивов объектов.

5.1.4 Типовые задания (оценочное средство - Собеседование) для оценки сформированности компетенции ОПК-6:

1. Целочисленные простые типы данных в языке программирования Си (char, int). Форматы двоичного представления и диапазоны значений.
2. Вещественнозначные простые типы данных в языке программирования Си (float, double). Форматы двоичного представления и диапазоны значений.
3. Объявление переменных в языке программирования Си. Способы инициализации объявленных переменных.
4. Использование квалификаторов при объявлении переменных.
5. Время жизни и область видимости переменных в языке программирования Си. Статические переменные. Локальные и глобальные переменные.
6. Константы в языке программирования Си, способы задания констант. Целочисленные (десятичные, восьмеричные, шестнадцатеричные) и действительные константы. Именованные константы (директива #define).
7. Константы в языке программирования Си, способы задания констант. Символьные и строковые константы.
8. Арифметические операции в языке программирования Си. Операции инкремента и декремента.
9. Операции отношения и логические операции в языке программирования Си.
10. Битовые операции в языке программирования Си.
11. Операции присваивания в языке программирования Си. Условная операция языка Си.
12. Операторы ветвления в языке программирования Си. Безусловный (goto) и условный (if ... else) переходы, блок-схемное представление.
13. Оператор-переключатель в языке программирования Си (switch ... case). Инструкция break. Множественного ветвление и его блок-схемное представление.
14. Операторы циклов с предусловием (while) и с постусловием (do ... while) в языке программирования Си. Блок-схемное представление циклических конструкций.
15. Оператор цикла for в языке программирования Си. Эквивалентное описание циклической конструкции при помощи оператора while.
16. Использование инструкций break и continue для циклов в языке программирования Си. Примеры.

17. Функции в языке программирования Си. Декларирование (прототипы функций), описание и вызов. Особенности передачи параметров функций.
18. Массивы в языке программирования Си. Объявление и использование одномерных массивов.
19. Представление строк в языке программирования Си как массивов символов.
20. Многомерные массивы в языке программирования Си. Объявление и использование двумерных массивов.
21. Указатели в языке программирования Си. Операции с указателями, адресная арифметика. Взаимосвязь массивов и указателей.
22. Аргументы функции main. Использование массивов строк с информацией об аргументах командной строки и переменных окружения.
23. Основные спецификации форматного ввода / вывода (для функций printf и scanf) в библиотеке стандартного ввода / вывода языка программирования Си.
24. Основные функции, используемые при работе с текстовыми файлами (открытие, закрытие, чтение, запись и пр.) в библиотеке стандартного ввода / вывода языка программирования Си.
25. Основы создания программ на языке C++: компиляция, директива include, комментарии, простейший ввод и вывод. Объявление переменных. Объявление простых функций и их использование.
26. Типы данных, преобразование типов, арифметические операции.
27. Массивы. Строки в стиле C. Строки класса string. Указатели. Управление динамической памятью с помощью new и delete.
28. Структуры в C++.
29. Ссылочные переменные. Передача функции аргументов по ссылке. Аргументы по умолчанию.
30. Перегрузка функций. Шаблоны функций. Спецификации шаблонов функций.
31. Раздельная компиляция программ. Продолжительность хранения, область видимости и компоновка.
32. Определение и реализация класса. Открытый и закрытый доступ к классу. Данные – члены класса. Методы класса. Создание и использование объектов класса. Конструкторы и деструкторы класса. Функции – члены const. Указатель this. Создание массивов объектов.

33. Перегрузка операций. Дружественные функции. Перегрузка операции для вывода.

34. Динамическое выделение памяти для членов класса. Явные и неявные конструкторы копирования. Явные и неявные перегруженные операции присваивания. Статические члены класса. Указатели на объекты.

Критерии оценивания (оценочное средство - Собеседование)

Оценка	Критерии оценивания
зачтено	Дан правильный развернутый ответ. Допустимы помарки, которые в ходе разговора будут устранены или пояснены.
не зачтено	Ответ неверный. Нет базовых знаний предмета.

5.2. Описание шкал оценивания результатов обучения по дисциплине при промежуточной аттестации

Шкала оценивания сформированности компетенций

Уровень сформированности компетенций (индикатора достижения компетенций)	плохо	неудовлетворительно	удовлетворительно	хорошо	очень хорошо	отлично	превосходно
	не зачтено			зачтено			
<u>Знания</u>	Отсутствие знаний теоретического материала. Невозможность оценить полноту знаний вследствие отказа обучающегося от ответа	Уровень знаний ниже минимальных требований. Имели место грубые ошибки	Минимально допустимый уровень знаний. Допущено много негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько негрубых ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Допущено несколько несущественных ошибок	Уровень знаний в объеме, соответствующем программе подготовки. Ошибок нет.	Уровень знаний в объеме, превышающем программу подготовки.
<u>Умения</u>	Отсутствие минимальных умений. Невозможность оценить наличие умений вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы основные умения. Имели место грубые ошибки	Продемонстрированы основные умения. Решены типовые задачи с негрубыми ошибками. Выполнены все задания, но не в	Продемонстрированы все основные умения. Решены все основные задачи с негрубыми ошибками. Выполнены все задания в полном	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания в полном объеме, но некоторые	Продемонстрированы все основные умения. Решены все основные задачи с отдельным и несущественными	Продемонстрированы все основные умения. Решены все основные задачи. Выполнены все задания, в полном объеме без недочетов

			полном объеме	объеме, но некоторые с недочетами	с недочетами	недочетами, выполнены все задания в полном объеме	
<u>Навыки</u>	Отсутствие базовых навыков. Невозможность оценить наличие навыков вследствие отказа обучающегося от ответа	При решении стандартных задач не продемонстрированы базовые навыки. Имели место грубые ошибки	Имеется минимальный набор навыков для решения стандартных задач с некоторым и недочетами	Продемонстрированы базовые навыки при решении стандартных задач с некоторым и недочетами	Продемонстрированы базовые навыки при решении стандартных задач без ошибок и недочетов	Продемонстрированы навыки при решении нестандартных задач без ошибок и недочетов	Продемонстрирован творческий подход к решению нестандартных задач

Шкала оценивания при промежуточной аттестации

Оценка		Уровень подготовки
зачтено	превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне выше предусмотренного программой
	отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично».
	очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо»
	хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо».
	удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
не зачтено	неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно».
	плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

5.3 Типовые контрольные задания или иные материалы, необходимые для оценки результатов обучения на промежуточной аттестации с указанием критериев их оценивания:

5.3.1 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ОПК-2

1. Используя функции математической библиотеки, реализовать программу на языке Си для вычисления значений для $u(x, a)$ в соответствии со своим вариантом. Программа во время своей работы должна принимать значения переменных x и a , определять их принадлежность области допустимых значений. При возможности подсчета u как

действительного числа – вывести его значение, в противном случае – вывести сообщение о причине невозможности нахождения значения переменной *y* (по вариантам).

2. Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран символов текста, располагающихся в заданном файле на нечётных позициях.

3. Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран чётных строк текста, содержащегося в заданном файле.

4. Программе задаются: имя входного текстового файла, начальный символ строк. Программа должна осуществлять вывод на экран содержащихся в заданном файле строк текста, начинающихся с указанного символа.

5. Программе задаются: целочисленные первый член и шаг арифметической прогрессии, количество элементов последовательности, имя выходного текстового файла. Программа должна осуществлять подсчёт и запись в текстовый файл значений элементов арифметической прогрессии.

6. Программе задаётся четырёхбайтовое целое число в шестнадцатеричном представлении. Программа должна осуществлять перемену местами старшего и младшего байтов заданного числа (например, 0xAABVCCDD → 0xDDBVCCAA) и выводить на экран результат в шестнадцатеричном виде.

7. В командной строке программе задаются: первый целочисленный аргумент арифметической операции, символ, определяющий арифметическую операцию ('+', '-', '*', '/'), второй целочисленный аргумент арифметической операции. Программа должна осуществлять заданную арифметическую операцию с заданными целыми числами и выводить на экран результат. При невозможности выполнения операции деления на нуль на экран должно выводиться соответствующее сообщение об ошибке.

8. Программе задаётся буквенный символ из латинского алфавита. Программа должна осуществлять вывод на экран всех переменных окружения (и их значений), название которых начинается с заданной буквы.

11

9. Программе задаются *N* значений элементов целочисленного массива (величина *N* считается константой, определённой на этапе компиляции программы, например, при помощи директивы препроцессора #define *N* 10). Программа должна осуществлять поиск максимального числового значения среди всех *N* заданных целых чисел (элементов массива) с выводом на экран индекса и значения найденного максимального элемента массива.

10. Построить прототипы функций по заданным описаниям:

а) `tofu(...)` принимает аргумент `int` и возвращает `float`;

б) `plot(...)` принимает указатель на структуру `tar` как аргумент и возвращает строку;

в) `func2(...)` принимает структуру `boss` в качестве аргумента и не возвращает ничего.

11. Предположим, что `somearr` – массив из 8 элементов `double`. Объявите указатель на первый элемент массива `somearr` и используйте этот указатель для отображения первого и последнего элементов массива.

12. Чему будет равно `*ptr_arg` после выполнения данного участка программы?

```
int arr[4] = {-2,-1,0,1};
```

```
int *ptr_arg = &arr[2];
```

```
ptr_arg++;
```

13. Вы хотите записать строчное значение в `char carr[100]` с клавиатуры. Вы знаете, что значение может включать пробельные символы. Какая из операций ввода, приведенных ниже, Вам подойдет:

а) `cin << carr;`

- б) `cin.getline(carr,100);`
- в) `cin >> carr;`

5.3.2 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ОПК-3

1. Используя функции математической библиотеки, реализовать программу на языке Си для вычисления значений для $y(x, a)$ в соответствии со своим вариантом. Программа во время своей работы должна принимать значения переменных x и a , определять их принадлежность области допустимых значений. При возможности подсчета y как действительного числа – вывести его значение, в противном случае – вывести сообщение о причине невозможности нахождения значения переменной y (по вариантам).
 2. Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран символов текста, располагающихся в заданном файле на нечётных позициях.
 3. Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран чётных строк текста, содержащегося в заданном файле.
 4. Программе задаются: имя входного текстового файла, начальный символ строк. Программа должна осуществлять вывод на экран содержащихся в заданном файле строк текста, начинающихся с указанного символа.
 5. Программе задаются: целочисленные первый член и шаг арифметической прогрессии, количество элементов последовательности, имя выходного текстового файла. Программа должна осуществлять подсчёт и запись в текстовый файл значений элементов арифметической прогрессии.
 6. Программе задаётся четырёхбайтовое целое число в шестнадцатеричном представлении. Программа должна осуществлять перемену местами старшего и младшего байтов заданного числа (например, `0xAABCCDD` → `0xDDBVCCAA`) и выводить на экран результат в шестнадцатеричном виде.
 7. В командной строке программе задаются: первый целочисленный аргумент арифметической операции, символ, определяющий арифметическую операцию ('+', '-', '*', '/'), второй целочисленный аргумент арифметической операции. Программа должна осуществлять заданную арифметическую операцию с заданными целыми числами и выводить на экран результат. При невозможности выполнения операции деления на нуль на экран должно выводиться соответствующее сообщение об ошибке.
 8. Программе задаётся буквенный символ из латинского алфавита. Программа должна осуществлять вывод на экран всех переменных окружения (и их значений), название которых начинается с заданной буквы.
- 11
9. Программе задаются N значений элементов целочисленного массива (величина N считается константой, определённой на этапе компиляции программы, например, при помощи директивы препроцессора `#define N 10`). Программа должна осуществлять поиск максимального числового значения среди всех N заданных целых чисел (элементов массива) с выводом на экран индекса и значения найденного максимального элемента массива.
 10. Построить прототипы функций по заданным описаниям:
 - а) `tofu(...)` принимает аргумент `int` и возвращает `float`;
 - б) `plot(...)` принимает указатель на структуру `tar` как аргумент и возвращает строку;
 - в) `func2(...)` принимает структуру `boss` в качестве аргумента и не возвращает ничего.
 11. Предположим, что `somearr` – массив из 8 элементов `double`. Объявите указатель на первый элемент массива `somearr` и используйте этот указатель для отображения первого

и последнего элементов массива.

12. Чему будет равно `*ptr_arg` после выполнения данного участка программы?

```
int arr[4] = {-2,-1,0,1};
```

```
int *ptr_arg = &arr[2];
```

```
ptr_arg++;
```

13. Вы хотите записать строчное значение в `char carr[100]` с клавиатуры. Вы знаете, что значение может включать пробельные символы. Какая из операций ввода, приведенных ниже, Вам подойдет:

а) `cin << carr;`

б) `cin.getline(carr,100);`

в) `cin >> carr;`

5.3.3 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ОПК-4

1. Используя функции математической библиотеки, реализовать программу на языке Си для вычисления значений для $y(x, a)$ в соответствии со своим вариантом. Программа во время своей работы должна принимать значения переменных x и a , определять их принадлежность области допустимых значений. При возможности подсчета y как действительного числа – вывести его значение, в противном случае – вывести сообщение о причине невозможности нахождения значения переменной y (по вариантам).

2. Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран символов текста, располагающихся в заданном файле на нечётных позициях.

3. Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран чётных строк текста, содержащегося в заданном файле.

4. Программе задаются: имя входного текстового файла, начальный символ строки.

Программа должна осуществлять вывод на экран содержащихся в заданном файле строк текста, начинающихся с указанного символа.

5. Программе задаются: целочисленные первый член и шаг арифметической прогрессии, количество элементов последовательности, имя выходного текстового файла. Программа должна осуществлять подсчёт и запись в текстовый файл значений элементов арифметической прогрессии.

6. Программе задаётся четырёхбайтовое целое число в шестнадцатеричном представлении. Программа должна осуществлять перемену местами старшего и младшего байтов заданного числа (например, `0xAABCCDD` → `0xDDBVCCAA`) и выводить на экран результат в шестнадцатеричном виде.

7. В командной строке программе задаются: первый целочисленный аргумент арифметической операции, символ, определяющий арифметическую операцию ('+', '-', '*', '/'), второй целочисленный аргумент арифметической операции. Программа должна осуществлять заданную арифметическую операцию с заданными целыми числами и выводить на экран результат. При невозможности выполнения операции деления на нуль на экран должно выводиться соответствующее сообщение об ошибке.

8. Программе задаётся буквенный символ из латинского алфавита. Программа должна осуществлять вывод на экран всех переменных окружения (и их значений), название которых начинается с заданной буквы.

11

9. Программе задаются N значений элементов целочисленного массива (величина N считается константой, определённой на этапе компиляции программы, например, при помощи директивы препроцессора `#define N 10`). Программа должна осуществлять

поиск максимального числового значения среди всех N заданных целых чисел (элементов массива) с выводом на экран индекса и значения найденного максимального элемента массива.

10. Построить прототипы функций по заданным описаниям:

а) `tofu(...)` принимает аргумент `int` и возвращает `float`;

б) `plot(...)` принимает указатель на структуру `tap` как аргумент и возвращает строку;

в) `func2(...)` принимает структуру `boss` в качестве аргумента и не возвращает ничего.

11. Предположим, что `somearr` – массив из 8 элементов `double`. Объявите указатель на первый элемент массива `somearr` и используйте этот указатель для отображения первого и последнего элементов массива.

12. Чему будет равно `*ptr_arr` после выполнения данного участка программы?

```
int arr[4] = {-2,-1,0,1};
```

```
int *ptr_arr = &arr[2];
```

```
ptr_arr++;
```

13. Вы хотите записать строчное значение в `char carr[100]` с клавиатуры. Вы знаете, что значение может включать пробельные символы. Какая из операций ввода, приведенных ниже, Вам подойдет:

а) `cin << carr;`

б) `cin.getline(carr,100);`

в) `cin >> carr;`

5.3.4 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ОПК-6

1. Используя функции математической библиотеки, реализовать программу на языке Си для вычисления значений для $u(x, a)$ в соответствии со своим вариантом. Программа во время своей работы должна принимать значения переменных x и a , определять их принадлежность области допустимых значений. При возможности подсчета u как действительного числа – вывести его значение, в противном случае – вывести сообщение о причине невозможности нахождения значения переменной u (по вариантам).

2. Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран символов текста, располагающихся в заданном файле на нечётных позициях.

3. Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран чётных строк текста, содержащегося в заданном файле.

4. Программе задаются: имя входного текстового файла, начальный символ строки. Программа должна осуществлять вывод на экран содержащихся в заданном файле строк текста, начинающихся с указанного символа.

5. Программе задаются: целочисленные первый член и шаг арифметической прогрессии, количество элементов последовательности, имя выходного текстового файла. Программа должна осуществлять подсчёт и запись в текстовый файл значений элементов арифметической прогрессии.

6. Программе задаётся четырёхбайтовое целое число в шестнадцатеричном представлении. Программа должна осуществлять перемену местами старшего и младшего байтов заданного числа (например, `0xAABCCDD` → `0xDDBCCAA`) и выводить на экран результат в шестнадцатеричном виде.

7. В командной строке программе задаются: первый целочисленный аргумент арифметической операции, символ, определяющий арифметическую операцию ('+', '-', '*', '/'), второй целочисленный аргумент арифметической операции. Программа должна осуществлять заданную арифметическую операцию с заданными целыми числами и

выводить на экран результат. При невозможности выполнения операции деления на нуль на экран должно выводиться соответствующее сообщение об ошибке.

8. Программе задаётся буквенный символ из латинского алфавита. Программа должна осуществлять вывод на экран всех переменных окружения (и их значений), название которых начинается с заданной буквы.

11

9. Программе задаются N значений элементов целочисленного массива (величина N считается константой, определённой на этапе компиляции программы, например, при помощи директивы препроцессора #define N 10). Программа должна осуществлять поиск максимального числового значения среди всех N заданных целых чисел (элементов массива) с выводом на экран индекса и значения найденного максимального элемента массива.

10. Построить прототипы функций по заданным описаниям:

а) tofu(...) принимает аргумент int и возвращает float;

б) plot(...) принимает указатель на структуру tar как аргумент и возвращает строку;

в) func2(...) принимает структуру boss в качестве аргумента и не возвращает ничего.

11. Предположим, что somearr – массив из 8 элементов double. Объявите указатель на первый элемент массива somearr и используйте этот указатель для отображения первого и последнего элементов массива.

12. Чему будет равно *ptr_arr после выполнения данного участка программы?

```
int arr[4] = {-2,-1,0,1};
```

```
int *ptr_arr = &arr[2];
```

```
ptr_arr++;
```

13. Вы хотите записать строчное значение в char carr[100] с клавиатуры. Вы знаете, что значение может включать пробельные символы. Какая из операций ввода, приведенных ниже, Вам подойдет:

а) cin << carr;

б) cin.getline(carr,100);

в) cin >> carr;

Критерии оценивания (оценочное средство - Практическое задание)

Оценка	Критерии оценивания
зачтено	Сданы все практические работы, по особенностям их реализации даны исчерпывающие комментарии.
не зачтено	Сдано меньше половины работ. Или меньше двух третей, но по сданным работам не даны никакие комментарии.

5.3.5 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ОПК-2

1. Используя функции математической библиотеки, реализовать программу на языке Си для вычисления значений для $y(x, a)$ в соответствии со своим вариантом. Программа во время своей работы должна принимать значения переменных x и a , определять их принадлежность области допустимых значений. При возможности подсчета y как действительного числа – вывести его значение, в противном случае – вывести сообщение о причине невозможности нахождения значения переменной y (по вариантам).

2. Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран символов текста, располагающихся в заданном файле на нечётных позициях.
3. Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран чётных строк текста, содержащегося в заданном файле.
4. Программе задаются: имя входного текстового файла, начальный символ строк. Программа должна осуществлять вывод на экран содержащихся в заданном файле строк текста, начинающихся с указанного символа.
5. Программе задаются: целочисленные первый член и шаг арифметической прогрессии, количество элементов последовательности, имя выходного текстового файла. Программа должна осуществлять подсчёт и запись в текстовый файл значений элементов арифметической прогрессии.
6. Программе задаётся четырёхбайтовое целое число в шестнадцатеричном представлении. Программа должна осуществлять перемену местами старшего и младшего байтов заданного числа (например, 0xAABVCCDD → 0xDDVCCAA) и выводить на экран результат в шестнадцатеричном виде.
7. В командной строке программе задаются: первый целочисленный аргумент арифметической операции, символ, определяющий арифметическую операцию ('+', '-', '*', '/'), второй целочисленный аргумент арифметической операции. Программа должна осуществлять заданную арифметическую операцию с заданными целыми числами и выводить на экран результат. При невозможности выполнения операции деления на нуль на экран должно выводиться соответствующее сообщение об ошибке.
8. Программе задаётся буквенный символ из латинского алфавита. Программа должна осуществлять вывод на экран всех переменных окружения (и их значений), название которых начинается с заданной буквы.

11

9. Программе задаются N значений элементов целочисленного массива (величина N считается константой, определённой на этапе компиляции программы, например, при помощи директивы препроцессора #define N 10). Программа должна осуществлять поиск максимального числового значения среди всех N заданных целых чисел (элементов массива) с выводом на экран индекса и значения найденного максимального элемента массива.

10. Построить прототипы функций по заданным описаниям:

- а) tofu(...) принимает аргумент int и возвращает float;
- б) plot(...) принимает указатель на структуру tar как аргумент и возвращает строку;
- в) func2(...) принимает структуру boss в качестве аргумента и не возвращает ничего.

11. Предположим, что somearr – массив из 8 элементов double. Объявите указатель на первый элемент массива somearr и используйте этот указатель для отображения первого и последнего элементов массива.

12. Чему будет равно *ptr_arr после выполнения данного участка программы?

```
int arr[4] = {-2,-1,0,1};
```

```
int *ptr_arr = &arr[2];
```

```
ptr_arr++;
```

13. Вы хотите записать строчное значение в char carr[100] с клавиатуры. Вы знаете, что значение может включать пробельные символы. Какая из операций ввода, приведенных ниже, Вам подойдет:

- а) cin << carr;
- б) cin.getline(carr,100);
- в) cin >> carr;

5.3.6 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ОПК-3

1. Используя функции математической библиотеки, реализовать программу на языке Си для вычисления значений для $y(x, a)$ в соответствии со своим вариантом. Программа во время своей работы должна принимать значения переменных x и a , определять их принадлежность области допустимых значений. При возможности подсчета y как действительного числа – вывести его значение, в противном случае – вывести сообщение о невозможности нахождения значения переменной y (по вариантам).
 2. Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран символов текста, располагающихся в заданном файле на нечётных позициях.
 3. Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран чётных строк текста, содержащегося в заданном файле.
 4. Программе задаются: имя входного текстового файла, начальный символ строки. Программа должна осуществлять вывод на экран содержащихся в заданном файле строк текста, начинающихся с указанного символа.
 5. Программе задаются: целочисленные первый член и шаг арифметической прогрессии, количество элементов последовательности, имя выходного текстового файла. Программа должна осуществлять подсчёт и запись в текстовый файл значений элементов арифметической прогрессии.
 6. Программе задаётся четырёхбайтовое целое число в шестнадцатеричном представлении. Программа должна осуществлять перемену местами старшего и младшего байтов заданного числа (например, $0xAABVCCDD \rightarrow 0xDDVBCCAA$) и выводить на экран результат в шестнадцатеричном виде.
 7. В командной строке программе задаются: первый целочисленный аргумент арифметической операции, символ, определяющий арифметическую операцию ('+', '-', '*', '/'), второй целочисленный аргумент арифметической операции. Программа должна осуществлять заданную арифметическую операцию с заданными целыми числами и выводить на экран результат. При невозможности выполнения операции деления на нуль на экран должно выводиться соответствующее сообщение об ошибке.
 8. Программе задаётся буквенный символ из латинского алфавита. Программа должна осуществлять вывод на экран всех переменных окружения (и их значений), название которых начинается с заданной буквы.
- 11
9. Программе задаются N значений элементов целочисленного массива (величина N считается константой, определённой на этапе компиляции программы, например, при помощи директивы препроцессора `#define N 10`). Программа должна осуществлять поиск максимального числового значения среди всех N заданных целых чисел (элементов массива) с выводом на экран индекса и значения найденного максимального элемента массива.
 10. Построить прототипы функций по заданным описаниям:
 - а) `tofu(...)` принимает аргумент `int` и возвращает `float`;
 - б) `plot(...)` принимает указатель на структуру `tar` как аргумент и возвращает строку;
 - в) `func2(...)` принимает структуру `boss` в качестве аргумента и не возвращает ничего.
 11. Предположим, что `somearr` – массив из 8 элементов `double`. Объявите указатель на первый элемент массива `somearr` и используйте этот указатель для отображения первого и последнего элементов массива.
 12. Чему будет равно `*ptr_arg` после выполнения данного участка программы?
`int arr[4] = {-2,-1,0,1};`

```
int *ptr_arr = &arr[2];
```

```
ptr_arr++;
```

13. Вы хотите записать строчное значение в `char carr[100]` с клавиатуры. Вы знаете, что значение может включать пробельные символы. Какая из операций ввода, приведенных ниже, Вам подойдет:

а) `cin << carr;`

б) `cin.getline(carr,100);`

в) `cin >> carr;`

5.3.7 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ОПК-4

1. Используя функции математической библиотеки, реализовать программу на языке Си для вычисления значений для $y(x, a)$ в соответствии со своим вариантом. Программа во время своей работы должна принимать значения переменных x и a , определять их принадлежность области допустимых значений. При возможности подсчета y как действительного числа – вывести его значение, в противном случае – вывести сообщение о невозможности нахождения значения переменной y (по вариантам).

2. Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран символов текста, располагающихся в заданном файле на нечётных позициях.

3. Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран чётных строк текста, содержащегося в заданном файле.

4. Программе задаются: имя входного текстового файла, начальный символ строк.

Программа должна осуществлять вывод на экран содержащихся в заданном файле строк текста, начинающихся с указанного символа.

5. Программе задаются: целочисленные первый член и шаг арифметической прогрессии, количество элементов последовательности, имя выходного текстового файла. Программа должна осуществлять подсчёт и запись в текстовый файл значений элементов арифметической прогрессии.

6. Программе задаётся четырёхбайтовое целое число в шестнадцатеричном представлении. Программа должна осуществлять переместку местами старшего и младшего байтов заданного числа (например, `0xAABBCCDD` → `0xDDBBCCAA`) и выводить на экран результат в шестнадцатеричном виде.

7. В командной строке программе задаются: первый целочисленный аргумент арифметической операции, символ, определяющий арифметическую операцию ('+', '-', '*', '/'), второй целочисленный аргумент арифметической операции. Программа должна осуществлять заданную арифметическую операцию с заданными целыми числами и выводить на экран результат. При невозможности выполнения операции деления на ноль на экран должно выводиться соответствующее сообщение об ошибке.

8. Программе задаётся буквенный символ из латинского алфавита. Программа должна осуществлять вывод на экран всех переменных окружения (и их значений), название которых начинается с заданной буквы.

11

9. Программе задаются N значений элементов целочисленного массива (величина N считается константой, определённой на этапе компиляции программы, например, при помощи директивы препроцессора `#define N 10`). Программа должна осуществлять поиск максимального числового значения среди всех N заданных целых чисел (элементов массива) с выводом на экран индекса и значения найденного максимального элемента массива.

10. Построить прототипы функций по заданным описаниям:

а) `tofu(...)` принимает аргумент `int` и возвращает `float`;

б) `plot(...)` принимает указатель на структуру `tar` как аргумент и возвращает строку;

в) `func2(...)` принимает структуру `boss` в качестве аргумента и не возвращает ничего.

11. Предположим, что `somearr` – массив из 8 элементов `double`. Объявите указатель на первый элемент массива `somearr` и используйте этот указатель для отображения первого и последнего элементов массива.

12. Чему будет равно `*ptr_arr` после выполнения данного участка программы?

```
int arr[4] = {-2,-1,0,1};
```

```
int *ptr_arr = &arr[2];
```

```
ptr_arr++;
```

13. Вы хотите записать строчное значение в `char carr[100]` с клавиатуры. Вы знаете, что значение может включать пробельные символы. Какая из операций ввода, приведенных ниже, Вам подойдет:

а) `cin << carr;`

б) `cin.getline(carr,100);`

в) `cin >> carr;`

5.3.8 Типовые задания (оценочное средство - Практическое задание) для оценки сформированности компетенции ОПК-6

1. Используя функции математической библиотеки, реализовать программу на языке Си для вычисления значений для $y(x, a)$ в соответствии со своим вариантом. Программа во время своей работы должна принимать значения переменных x и a , определять их принадлежность области допустимых значений. При возможности подсчета y как действительного числа – вывести его значение, в противном случае – вывести сообщение о причине невозможности нахождения значения переменной y (по вариантам).

2. Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран символов текста, располагающихся в заданном файле на нечётных позициях.

3. Программе задаётся имя входного текстового файла. Программа должна осуществлять вывод на экран чётных строк текста, содержащегося в заданном файле.

4. Программе задаются: имя входного текстового файла, начальный символ строк.

Программа должна осуществлять вывод на экран содержащихся в заданном файле строк текста, начинающихся с указанного символа.

5. Программе задаются: целочисленные первый член и шаг арифметической прогрессии, количество элементов последовательности, имя выходного текстового файла. Программа должна осуществлять подсчёт и запись в текстовый файл значений элементов арифметической прогрессии.

6. Программе задаётся четырёхбайтовое целое число в шестнадцатеричном представлении. Программа должна осуществлять перемену местами старшего и младшего байтов заданного числа (например, `0xAABBCCDD` → `0xDDBBCCAA`) и выводить на экран результат в шестнадцатеричном виде.

7. В командной строке программе задаются: первый целочисленный аргумент арифметической операции, символ, определяющий арифметическую операцию ('+', '-', '*', '/'), второй целочисленный аргумент арифметической операции. Программа должна осуществлять заданную арифметическую операцию с заданными целыми числами и выводить на экран результат. При невозможности выполнения операции деления на ноль на экран должно выводиться соответствующее сообщение об ошибке.

8. Программе задаётся буквенный символ из латинского алфавита. Программа должна

осуществлять вывод на экран всех переменных окружения (и их значений), название которых начинается с заданной буквы.

11

9. Программе задаются N значений элементов целочисленного массива (величина N считается константой, определённой на этапе компиляции программы, например, при помощи директивы препроцессора #define N 10). Программа должна осуществлять поиск максимального числового значения среди всех N заданных целых чисел (элементов массива) с выводом на экран индекса и значения найденного максимального элемента массива.

10. Построить прототипы функций по заданным описаниям:

а) tofu(...) принимает аргумент int и возвращает float;

б) plot(...) принимает указатель на структуру tar как аргумент и возвращает строку;

в) func2(...) принимает структуру boss в качестве аргумента и не возвращает ничего.

11. Предположим, что somearr – массив из 8 элементов double. Объявите указатель на первый элемент массива somearr и используйте этот указатель для отображения первого и последнего элементов массива.

12. Чему будет равно *ptr_arg после выполнения данного участка программы?

```
int arr[4] = {-2,-1,0,1};
```

```
int *ptr_arg = &arr[2];
```

```
ptr_arg++;
```

13. Вы хотите записать строчное значение в char carr[100] с клавиатуры. Вы знаете, что значение может включать пробельные символы. Какая из операций ввода, приведенных ниже, Вам подойдет:

а) cin << carr;

б) cin.getline(carr,100);

в) cin >> carr;

Критерии оценивания (оценочное средство - Практическое задание)

Оценка	Критерии оценивания
превосходно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «превосходно», продемонстрированы знания, умения, владения по соответствующим компетенциям на уровне, выше предусмотренного программой
отлично	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «отлично», при этом хотя бы одна компетенция сформирована на уровне «отлично»
очень хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «очень хорошо», при этом хотя бы одна компетенция сформирована на уровне «очень хорошо»
хорошо	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «хорошо», при этом хотя бы одна компетенция сформирована на уровне «хорошо»
удовлетворительно	Все компетенции (части компетенций), на формирование которых направлена дисциплина, сформированы на уровне не ниже «удовлетворительно», при

Оценка	Критерии оценивания
	этом хотя бы одна компетенция сформирована на уровне «удовлетворительно»
неудовлетворительно	Хотя бы одна компетенция сформирована на уровне «неудовлетворительно», ни одна из компетенций не сформирована на уровне «плохо»
плохо	Хотя бы одна компетенция сформирована на уровне «плохо»

6. Учебно-методическое и информационное обеспечение дисциплины (модуля)

Основная литература:

1. Керниган Б. В. Язык программирования С : пер. с англ. - 2-е изд., перераб. и доп. - М. : Вильямс, 2006. - 304 с. - ISBN 5-8459-0891-4 : 138.00., 2 экз.
2. Трофимов В. В. Информатика в 2 т. Том 1 : учебник / В. В. Трофимов. - 3-е изд. ; пер. и доп. - Москва : Юрайт, 2023. - 553 с. - (Профессиональное образование). - ISBN 978-5-534-02518-7. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=843801&idb=0>.
3. Трофимов В. В. Информатика в 2 т. Том 2 : учебник / В. В. Трофимов. - 3-е изд. ; пер. и доп. - Москва : Юрайт, 2023. - 406 с. - (Профессиональное образование). - ISBN 978-5-534-02519-4. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=841921&idb=0>.

Дополнительная литература:

1. Трофимов В. В. Алгоритмизация и программирование : учебник / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. - Москва : Юрайт, 2022. - 137 с. - (Высшее образование). - URL: <https://urait.ru/bcode/491215> (дата обращения: 14.08.2022). - ISBN 978-5-534-07834-3 : 439.00. - Текст : электронный // ЭБС "Юрайт"., <https://e-lib.unn.ru/MegaPro/UserEntry?Action=FindDocs&ids=820098&idb=0>.

Программное обеспечение и Интернет-ресурсы (в соответствии с содержанием дисциплины):

1. Интегрированная среда разработки программного обеспечения Microsoft Visual Studio.
2. Средства компиляции GNU Compiler Collection в составе ОС Linux.
3. Среда разработки программного обеспечения Geany в составе ОС Linux.

7. Материально-техническое обеспечение дисциплины (модуля)

Учебные аудитории для проведения учебных занятий, предусмотренных образовательной программой, оснащены мультимедийным оборудованием (проектор, экран), техническими средствами обучения, компьютерами.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечены доступом в электронную

информационно-образовательную среду.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению подготовки/специальности 11.05.02 - Специальные радиотехнические системы.

Автор(ы): Морозов Никита Сергеевич, кандидат технических наук.

Программа одобрена на заседании методической комиссии от 16.01.2024 г., протокол № №1.