

МИНОБРНАУКИ РОССИИ

ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Рабочая программа дисциплины (модуля)  
**ПРОГРАММИРОВАНИЕ НА С**

Направление и направленность (профиль)  
09.03.03 Прикладная информатика. Прикладная информатика

Год набора на ОПОП  
2023

Форма обучения  
очная

Владивосток 2023

Рабочая программа дисциплины (модуля) «Программирование на С» составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.03 Прикладная информатика (утв. приказом Минобрнауки России от 19.09.2017г. №922) и Порядком организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры (утв. приказом Минобрнауки России от 06.04.2021 г. N245).

Составитель(и):

*Васильев Б.К., кандидат химических наук, доцент, Кафедра информационных технологий и систем, boris.vasiliev@vvsu.ru*

*Можаровский И.С., доцент, Кафедра информационных технологий и систем, Mozharovskiy.Igor@vvsu.ru*

Утверждена на заседании кафедры информационных технологий и систем от 31.05.2023 , протокол № 9

СОГЛАСОВАНО:

Заведующий кафедрой (разработчика)

Кийкова Е.В.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ	
Сертификат	1575633692
Номер транзакции	0000000000BB7AC0
Владелец	Кийкова Е.В.

## 1 Цель, планируемые результаты обучения по дисциплине (модулю)

Целью освоения дисциплины «Программирование на С» является получение базовых знаний в области программировании, непосредственное изучение высокоуровневого языка программирования С (как одного из языков общего назначения), сред разработки, правил оформления кода, необходимых выпускнику для решения различных задач практической, научно-исследовательской и педагогической деятельности.

Задачи освоения дисциплины состоят в формировании профессиональных компетенций, позволяющих решать стандартные задачи профессиональной деятельности на основе информационных технологий.

Планируемыми результатами обучения по дисциплине (модулю), являются знания, умения, навыки. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы, представлен в таблице 1.

Таблица 1 – Компетенции, формируемые в результате изучения дисциплины (модуля)

Название ОПОП ВО, сокращенное	Код и формулировка компетенции	Код и формулировка индикатора достижения компетенции	Результаты обучения по дисциплине		
			Код результата	Формулировка результата	
09.03.03 «Прикладная информатика» (Б-ПИ)	ПКВ-1 : Способен осуществлять интеграцию и тестирование работоспособности программных модулей и компонент программного обеспечения	ПКВ-1.1к : Осуществляет сборку программных модулей и компонент в программный продукт	РД1	Знание	основных средств и технологий для разработки программного обеспечения
			РД2	Умение	применять технологии программирования при разработке программного обеспечения
			РД3	Навык	применения технологий программирования при разработке и оформлении программ

## 2 Место дисциплины (модуля) в структуре ОПОП

Отнесение дисциплины к элективной части Блока 1 «Дисциплины (модули)» учебного плана ОПОП определяется спецификой и миссией ВВГУ, а также особенностями взаимодействия ВВГУ с рынком труда и региональными требованиями, выраженными в результатах образования и компетенциях.

Входными требованиями, необходимыми для освоения дисциплины, является наличие у обучающихся компетенций, сформированных на предыдущем уровне образования и при изучении дисциплины «Информатика и основы программирования».

## 3. Объем дисциплины (модуля)

Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу с обучающимися (по видам учебных занятий) и на самостоятельную работу, приведен в таблице 2.

Таблица 2 – Общая трудоемкость дисциплины

Название	Форма	Часть	Семестр (ОФО)	Трудоемкость	Объем контактной работы (час)		Форма
					Аудиторная	Внеаудиторная	

ОПОП ВО	обучения	УП	или курс (ЗФО, ОЗФО)	(З.Е.)	Всего						СРС	аттестации
						лек.	прак.	лаб.	ПА	КСР		
09.03.03 Прикладная информатика	ОФО	Б1.ДВ.Б	2	3	55	18	0	36	1	0	53	Э

## 4 Структура и содержание дисциплины (модуля)

### 4.1 Структура дисциплины (модуля) для ОФО

Тематический план, отражающий содержание дисциплины (перечень разделов и тем), структурированное по видам учебных занятий с указанием их объемов в соответствии с учебным планом, приведен в таблице 3.1

Таблица 3.1 – Разделы дисциплины (модуля), виды учебной деятельности и формы текущего контроля для ОФО

№	Название темы	Код результата обучения	Кол-во часов, отведенное на				Форма текущего контроля
			Лек	Практ	Лаб	СРС	
1	Основные принципы и понятия языка С. Основные встроенные типы данных.	РД1, РД2, РД3, РД4	1	0	2	3	отчет по лабораторной работе
2	Основные операции языка С. Консольный ввод и вывод. Массивы.	РД1, РД2, РД3, РД4	1	0	2	3	отчет по лабораторной работе
3	Управляющие конструкции языка С. Функции в языке С.	РД1, РД2, РД3, РД4	1	0	2	3	отчет по лабораторной работе
4	Область действия переменных и связанные с ней понятия.	РД1, РД2, РД3, РД4	1	0	2	3	отчет по лабораторной работе
5	Создание программ из нескольких модулей.	РД1, РД2, РД3, РД4	1	0	2	3	отчет по лабораторной работе
6	Указатели и динамическая память.	РД1, РД2, РД3, РД4	1	0	2	3	отчет по лабораторной работе
7	Работа с файлами.	РД1, РД2, РД3, РД4	1	0	2	3	отчет по лабораторной работе
8	Переименование типов, перечисляемые типы, структуры, объединения.	РД1, РД2, РД3, РД4	1	0	2	3	отчет по лабораторной работе
9	Работа со строками и памятью.	РД1, РД2, РД3, РД4	1	0	2	3	отчет по лабораторной работе
10	Директивы препроцессора.	РД1, РД2, РД3, РД4	1	0	2	3	отчет по лабораторной работе
11	Версии языка программирования С.	РД1, РД2, РД3, РД4	1	0	2	3	отчет по лабораторной работе
12	Работа с датами и временем.	РД1, РД2, РД3, РД4	1	0	2	3	отчет по лабораторной работе
13	Указатели на функции. Аргументы функции main.	РД1, РД2, РД3, РД4	1	0	2	3	отчет по лабораторной работе
14	Рекурсия.	РД1, РД2, РД3, РД4	1	0	2	3	отчет по лабораторной работе
15	Обработка исключительных ситуаций.	РД1, РД2, РД3, РД4	1	0	2	3	отчет по лабораторной работе
16	Нелокальные переходы.	РД1, РД2, РД3, РД4	1	0	2	3	отчет по лабораторной работе
17	Стандарты оформления программного кода.	РД1, РД2, РД3	2	0	4	5	отчет по лабораторной работе
<b>Итого по таблице</b>			<b>18</b>	<b>0</b>	<b>36</b>	<b>53</b>	

### 4.2 Содержание разделов и тем дисциплины (модуля) для ОФО

*Тема 1 Основные принципы и понятия языка С. Основные встроенные типы данных.*

Содержание темы: Понятие программирования и виды языков программирования. Краткая история возникновения языка программирования. Плюсы и минусы языка программирования С. Процесс компиляции.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, лабораторное занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по лабораторной работе, подготовка к промежуточной аттестации.

*Тема 2 Основные операции языка С. Консольный ввод и вывод. Массивы.*

Содержание темы: Знакомство с одной из сред программирования. Пример написания программ. Основные операции языка программирования С (арифметические, логические, сравнения и т.д.) и особенности их применения. Использование статических массивов для обработки и хранения данных.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, лабораторное занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по лабораторной работе, подготовка к промежуточной аттестации.

*Тема 3 Управляющие конструкции языка С. Функции в языке С.*

Содержание темы: Основные конструкции языка программирования С и ограниченность их применения. Разделение программы на подпрограммы (функции) и их реализация.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, лабораторное занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по лабораторной работе, подготовка к промежуточной аттестации.

*Тема 4 Область действия переменных и связанные с ней понятия.*

Содержание темы: Виды переменных (локальные и глобальные), их создание и уничтожение. Модификация переменных.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, лабораторное занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по лабораторной работе, подготовка к промежуточной аттестации.

*Тема 5 Создание программ из нескольких модулей.*

Содержание темы: Описание процедуры сборки программы из нескольких модулей. Использование механизмов заголовочных файлов. Защита от множественного включения.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, лабораторное занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по лабораторной работе, подготовка к промежуточной аттестации.

*Тема 6 Указатели и динамическая память.*

Содержание темы: Работа с динамическим распределением памяти (выделение и уничтожение). Смежное и несмежное расположение данных в памяти компьютера. Особенности доступа к данным (динамическая и статическая память).

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, лабораторное занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по лабораторной работе, подготовка к промежуточной аттестации.

### *Тема 7 Работа с файлами.*

Содержание темы: Работа с тестовыми и бинарными файлами. Спецификаторы доступа к файлам. Основные операции чтения и записи данных из/в файл.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, лабораторное занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по лабораторной работе, подготовка к промежуточной аттестации.

### *Тема 8 Переименование типов, перечисляемые типы, структуры, объединения.*

Содержание темы: Создание и использование пользовательских типов данных. Особенности объявления типов данных в языке программирования С.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, лабораторное занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по лабораторной работе, подготовка к промежуточной аттестации.

### *Тема 9 Работа со строками и памятью.*

Содержание темы: Представление строки в памяти компьютера. Использование стандартных функций для работы со строками (копирование, конкатенация, поиск символа в строке, разбор строк на лексемы и т.д.) и памятью.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, лабораторное занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по лабораторной работе, подготовка к промежуточной аттестации.

### *Тема 10 Директивы препроцессора.*

Содержание темы: Понятие препроцессора языка программирования С. Управление препроцессором с помощью директив. особенность использования макросов.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, лабораторное занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по лабораторной работе, подготовка к промежуточной аттестации.

### *Тема 11 Версии языка программирования С.*

Содержание темы: Основные этапы развития языка программирования С. Использование препроцессора для получения текущей версии языка. Переключение версий языка программирования в компиляторе (особенность сред программирования).

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, лабораторное занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по лабораторной работе, подготовка к промежуточной аттестации.

### *Тема 12 Работа с датами и временем.*

Содержание темы: Стандартная библиотека языка С для работы с датами и временем. Понятие информационной эры. Виды представлений даты и времени.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, лабораторное занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по лабораторной работе, подготовка к промежуточной аттестации.

### *Тема 13 Указатели на функции. Аргументы функции main.*

Содержание темы: Объявление указателя, способы вызовов. Прототипы функции main. Передача переменных окружения. Передача параметров в программу при запуске.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, лабораторное занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по лабораторной работе, подготовка к промежуточной аттестации.

#### *Тема 14 Рекурсия.*

Содержание темы: Понятие рекурсии, её виды и реализация. Ограничение использование данного способа организации программы.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, лабораторное занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по лабораторной работе, подготовка к промежуточной аттестации.

#### *Тема 15 Обработка исключительных ситуаций.*

Содержание темы: Понятие исключительных ситуаций и их виды. Регистрация и использование обработчиков сигналов.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, лабораторное занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по лабораторной работе, подготовка к промежуточной аттестации.

#### *Тема 16 Нелокальные переходы.*

Содержание темы: Понятие нелокального перехода, ограничения при использовании. Практическое применение.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, лабораторное занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по лабораторной работе, подготовка к промежуточной аттестации.

#### *Тема 17 Стандарты оформления программного кода.*

Содержание темы: Рекомендации при оформлении программного кода. Виды написания составных слов. Пример правил оформления и кода.

Формы и методы проведения занятий по теме, применяемые образовательные технологии: лекция, лабораторное занятие.

Виды самостоятельной подготовки студентов по теме: подготовка отчета по лабораторной работе, подготовка к промежуточной аттестации.

## **5 Методические указания для обучающихся по изучению и реализации дисциплины (модуля)**

### **5.1 Методические рекомендации обучающимся по изучению дисциплины и по обеспечению самостоятельной работы**

В ходе изучения дисциплины «Программирование на С» студенты могут посещать аудиторные занятия (лекции, лабораторные занятия, консультации). Особенность изучения дисциплины «Программирование на С» состоит в выполнении комплекса лабораторных работ, главной задачей которого является получение навыков самостоятельной работы на компьютерах с использованием современных информационных систем для решения различных учебных и профессиональных задач.

Каждая тема должна быть подкреплена выполнением самостоятельной работы.

Ниже приведены примеры заданий для выполнения самостоятельных работ.

1. Создать файл input.txt, в который поместить 2 случайные квадратные матрицы

размера, заданного пользователем с консоли. Закрывать файл. Очистить массивы.

Открыть input.txt файл и прочитать матрицы, произвести их перемножение и сложение (с помощью своих функций), вывести результат в файл output.txt.

Замерить время работы программы и вывести на экран.

2. Создать файл input.txt, в который поместить N случайных чисел в вектор размера N, заданного пользователем с консоли. Очистить массивы. Закрывать файл.

1) Открыть input.txt файл и прочитать значения; все числа, которые делятся на 5, заменить на слово «ПЯТЬ», на 7 – на слово «СЕМЬ». А которые делятся на 7 и 5 на слово «ПЯТЬСЕМЬ». Вывести результат в файл output.txt.

2) Из файла input.txt сформировать матрицу путем преобразования вектора N в M столбцов (лишние цифры не брать, если равных столбцов не получается, то сформировать квадратную матрицу). Посчитать сумму строк и столбцов. Вывести на экран матрицу и результаты сложения.

Замерить время работы программы и вывести на экран.

## **5.2 Особенности организации обучения для лиц с ограниченными возможностями здоровья и инвалидов**

При необходимости обучающимся из числа лиц с ограниченными возможностями здоровья и инвалидов (по заявлению обучающегося) предоставляется учебная информация в доступных формах с учетом их индивидуальных психофизических особенностей:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания, консультации и др.

- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания, консультации и др.

- для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; индивидуальные задания, консультации и др.

## **6 Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине (модулю)**

В соответствии с требованиями ФГОС ВО для аттестации обучающихся на соответствие их персональных достижений планируемым результатам обучения по дисциплине (модулю) созданы фонды оценочных средств. Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 1.

## **7 Учебно-методическое и информационное обеспечение дисциплины (модуля)**

### ***0.1 Основная литература***

### ***0.2 Дополнительная литература***

### ***0.3 Ресурсы информационно-телекоммуникационной сети "Интернет", включая профессиональные базы данных и информационно-справочные системы (при необходимости):***

Отсутствуют

**8 Материально-техническое обеспечение дисциплины (модуля) и перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения**

МИНОБРНАУКИ РОССИИ

ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

Фонд оценочных средств  
для проведения текущего контроля  
и промежуточной аттестации по дисциплине (модулю)

## **ПРОГРАММИРОВАНИЕ НА С**

Направление и направленность (профиль)

09.03.03 Прикладная информатика. Прикладная информатика

Год набора на ОПОП  
2023

Форма обучения  
очная

Владивосток 2023

## 1 Перечень формируемых компетенций

Название ОПОП ВО, сокращенное	Код и формулировка компетенции	Код и формулировка индикатора достижения компетенции
09.03.03 «Прикладная информатика» (Б-ПИ)	ПКВ-1 : Способен осуществлять интеграцию и тестирование работ оспособности программных модулей и компонент программного обеспечения	ПКВ-1.1к : Осуществляет сборку программных модулей и компонент в программный продукт

Компетенция считается сформированной на данном этапе в случае, если полученные результаты обучения по дисциплине оценены положительно (диапазон критериев оценивания результатов обучения «зачтено», «удовлетворительно», «хорошо», «отлично»). В случае отсутствия положительной оценки компетенция на данном этапе считается несформированной.

## 2 Показатели оценивания планируемых результатов обучения

**Компетенция ПКВ-1 «Способен осуществлять интеграцию и тестирование работоспособности программных модулей и компонент программного обеспечения»**

Таблица 2.1 – Критерии оценки индикаторов достижения компетенции

Код и формулировка индикатора достижения компетенции	Результаты обучения по дисциплине			Критерии оценивания результатов обучения
	Код результата	Тип результата	Результат	
ПКВ-1.1к : Осуществляет сборку программных модулей и компонент в программный продукт	РД1	Знание	основных средств и технологий для разработки программного обеспечения	Сформировавшееся знание основных средств и технологий для разработки программного обеспечения
	РД2	Умение	применять технологии программирования при разработке программного обеспечения	Сформировавшееся умение применять технологии программирования при разработке программного обеспечения
	РД3	Навык	применения технологий программирования при разработке и оформлении программ	Сформировавшиеся навыки применения технологий программирования при разработке и оформлении программ

Таблица заполняется в соответствии с разделом 1 Рабочей программы дисциплины (модуля).

## 3 Перечень оценочных средств

Таблица 3 – Перечень оценочных средств по дисциплине (модулю)

Контролируемые планируемые резу	Контролируемые темы	Наименование оценочного средства и представление его в ФОС

Результаты обучения		дисциплины	Текущий контроль	Промежуточная аттестация
Очная форма обучения				
РД1	Знание : основных средств и технологий для разработки программного обеспечения	1.1. Основные принципы и понятия языка С. Основные встроенные типы данных.	Лабораторная работа	Тест
		1.2. Основные операции языка С. Консольный ввод и вывод. Массивы.	Лабораторная работа	Тест
		1.3. Управляющие конструкции языка С. Функции и в языке С.	Лабораторная работа	Тест
		1.4. Область действия переменных и связанные с ней понятия.	Лабораторная работа	Тест
		1.5. Создание программ из нескольких модулей.	Лабораторная работа	Тест
		1.6. Указатели и динамическая память.	Лабораторная работа	Тест
		1.7. Работа с файлами.	Лабораторная работа	Тест
		1.8. Переименование типов, перечисляемые типы, структуры, объединения.	Лабораторная работа	Тест
		1.9. Работа со строками и памятью.	Лабораторная работа	Тест
		1.10. Директивы препроцессора.	Лабораторная работа	Тест
		1.11. Версии языка программирования С.	Лабораторная работа	Тест
		1.12. Работа с датами и временем.	Лабораторная работа	Тест
		1.13. Указатели на функции. Аргументы функции и main.	Лабораторная работа	Тест
		1.14. Рекурсия.	Лабораторная работа	Тест
		1.15. Обработка исключительных ситуаций.	Лабораторная работа	Тест
		1.16. Нелокальные переходы.	Лабораторная работа	Тест
		1.17. Стандарты оформления программного кода.	Лабораторная работа	Тест
РД2	Умение : применять технологии программирования при разработке программного обеспечения	1.1. Основные принципы и понятия языка С. Основные встроенные типы данных.	Лабораторная работа	Тест
		1.2. Основные операции языка С. Консольный ввод и вывод. Массивы.	Лабораторная работа	Тест

		1.3. Управляющие конструкторы языка C. Функции и в языке C.	Лабораторная работа	Тест
		1.4. Область действия переменных и связанные с ней понятия.	Лабораторная работа	Тест
		1.5. Создание программ из нескольких модулей.	Лабораторная работа	Тест
		1.6. Указатели и динамическая память.	Лабораторная работа	Тест
		1.7. Работа с файлами.	Лабораторная работа	Тест
		1.8. Переименование типов, перечисляемые типы, структуры, объединения.	Лабораторная работа	Тест
		1.9. Работа со строками и памятью.	Лабораторная работа	Тест
		1.10. Директивы препроцессора.	Лабораторная работа	Тест
		1.11. Версии языка программирования C.	Лабораторная работа	Тест
		1.12. Работа с датами и временем.	Лабораторная работа	Тест
		1.13. Указатели на функции. Аргументы функции и main.	Лабораторная работа	Тест
		1.14. Рекурсия.	Лабораторная работа	Тест
		1.15. Обработка исключительных ситуаций.	Лабораторная работа	Тест
		1.16. Нелокальные перемены.	Лабораторная работа	Тест
		1.17. Стандарты оформления программного кода.	Лабораторная работа	Тест
РДЗ	Навык : применения технологий программирования при разработке и оформлении программ	1.1. Основные принципы и понятия языка C. Основные встроенные типы данных.	Лабораторная работа	Тест
		1.2. Основные операции языка C. Консольный ввод и вывод. Массивы.	Лабораторная работа	Тест
		1.3. Управляющие конструкторы языка C. Функции и в языке C.	Лабораторная работа	Тест
		1.4. Область действия переменных и связанные с ней понятия.	Лабораторная работа	Тест
		1.5. Создание программ из нескольких модулей.	Лабораторная работа	Тест
		1.6. Указатели и динамическая память.	Лабораторная работа	Тест
		1.7. Работа с файлами.	Лабораторная работа	Тест

		1.8. Переименование типов, перечисляемые типы, структуры, объединения.	Лабораторная работа	Тест
		1.9. Работа со строками и памятью.	Лабораторная работа	Тест
		1.10. Директивы препроцессора.	Лабораторная работа	Тест
		1.11. Версии языка программирования С.	Лабораторная работа	Тест
		1.12. Работа с датами и временем.	Лабораторная работа	Тест
		1.13. Указатели на функции. Аргументы функции и main.	Лабораторная работа	Тест
		1.14. Рекурсия.	Лабораторная работа	Тест
		1.15. Обработка исключительных ситуаций.	Лабораторная работа	Тест
		1.16. Нелокальные переходы.	Лабораторная работа	Тест
		1.17. Стандарты оформления программного кода.	Лабораторная работа	Тест
РД4	Умение : разрабатывать алгоритмы для практической реализации в виде компьютерных программ в области профессиональной деятельности	1.1. Основные принципы и понятия языка С. Основные встроенные типы данных.	Лабораторная работа	Тест
		1.2. Основные операции языка С. Консольный ввод и вывод. Массивы.	Лабораторная работа	Тест
		1.3. Управляющие конструкции языка С. Функции и в языке С.	Лабораторная работа	Тест
		1.4. Область действия переменных и связанные с ней понятия.	Лабораторная работа	Тест
		1.5. Создание программ из нескольких модулей.	Лабораторная работа	Тест
		1.6. Указатели и динамическая память.	Лабораторная работа	Тест
		1.7. Работа с файлами.	Лабораторная работа	Тест
		1.8. Переименование типов, перечисляемые типы, структуры, объединения.	Лабораторная работа	Тест
		1.9. Работа со строками и памятью.	Лабораторная работа	Тест
		1.10. Директивы препроцессора.	Лабораторная работа	Тест
		1.11. Версии языка программирования С.	Лабораторная работа	Тест
		1.12. Работа с датами и временем.	Лабораторная работа	Тест

		1.13. Указатели на функции. Аргументы функции и main.	Лабораторная работа	Тест
		1.14. Рекурсия.	Лабораторная работа	Тест
		1.15. Обработка исключительных ситуаций.	Лабораторная работа	Тест
		1.16. Нелокальные перемены.	Лабораторная работа	Тест

#### 4 Описание процедуры оценивания

Качество сформированности компетенций на данном этапе оценивается по результатам текущих и промежуточных аттестаций при помощи количественной оценки, выраженной в баллах. Максимальная сумма баллов по дисциплине (модулю) равна 100 баллам.

Вид учебной деятельности	Оценочное средство		
	Отчеты по лабораторным работам	Тест	Итого
Лабораторные работы	80		60
Промежуточная аттестация (лекции)		20	20
Итого	80	20	100

Сумма баллов, набранных студентом по всем видам учебной деятельности в рамках дисциплины, переводится в оценку в соответствии с таблицей.

Сумма баллов по дисциплине	Оценка по промежуточной аттестации	Характеристика качества сформированности компетенции
от 91 до 100	«зачтено» / «отлично»	Студент демонстрирует сформированность дисциплинарных компетенций, обнаруживает всестороннее, систематическое и глубокое знание учебного материала, усвоил основную литературу и знаком с дополнительной литературой, рекомендованной программой, умеет свободно выполнять практические задания, предусмотренные программой, свободно оперирует приобретенными знаниями и умениями, применяет их в ситуациях повышенной сложности.
от 76 до 90	«зачтено» / «хорошо»	Студент демонстрирует сформированность дисциплинарных компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
от 61 до 75	«зачтено» / «удовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций: в ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных знаний, умений, навыков по некоторым дисциплинарным компетенциям, студент испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.
от 41 до 60	«не зачтено» / «неудовлетворительно»	У студента не сформированы дисциплинарные компетенции, проявляется недостаточность знаний, умений, навыков.
от 0 до 40	«не зачтено» / «неудовлетворительно»	Дисциплинарные компетенции не сформированы. Проявляется полное или практически полное отсутствие знаний, умений, навыков.

#### 5 Примерные оценочные средства

##### 5.1 Пример заданий на лабораторную работу

- Тема 1. Основные принципы и понятия языка C. Основные встроенные типы данных.
- Тема 2. Основные операции языка C. Консольный ввод и вывод. Массивы.
- Тема 3. Управляющие конструкции языка C. Функции в языке C.
- Тема 4. Область действия переменных и связанные с ней понятия.
- Тема 5. Создание программ из нескольких модулей.
- Тема 6. Указатели и динамическая память.
- Тема 7. Работа с файлами.
- Тема 8. Переименование типов, перечисляемые типы, структуры, объединения.
- Тема 9. Работа со строками и памятью.
- Тема 10. Директивы препроцессора.
- Тема 11. Версии языка программирования C.
- Тема 12. Работа с датами и временем.
- Тема 13. Указатели на функции. Аргументы функции main.
- Тема 14. Рекурсия.
- Тема 15. Обработка исключительных ситуаций.
- Тема 16. Нелокальные переходы.
- Тема 17. Стандарты оформления программного кода.

*Краткие методические указания*

На выполнение одной лабораторной работы отводится не менее одного двухчасового занятия (включая затраты времени на проведение промежуточного теста на последнем в учебном периоде лабораторном занятии). После выполнения каждой лабораторной работы студент должен представить отчет о ее выполнении, а также, по указаниям преподавателя, выполнить дополнительные практические задания по теме лабораторной работы.

*Шкала оценки*

№	Баллы	Описание
5	73–80	Студент демонстрирует умения на итоговом уровне: умеет свободно выполнять практически все задания, предусмотренные программой, свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
4	61–72	Студент демонстрирует умения на среднем уровне: освоил основные умения, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.
3	49–60	Студент демонстрирует умения и навыки на базовом уровне: в ходе контрольных мероприятий допускаются значительные ошибки, проявляется отсутствие отдельных умений, навыков по дисциплинарной компетенции, испытываются значительные затруднения при оперировании умениями и при их переносе на новые ситуации.
2	33–48	Студент демонстрирует умения и навыки на уровне ниже базового: проявляется недостаточность умений и навыков.
1	0–32	Студентом проявляется полное или практически полное отсутствие умений и навыков.

## 5.2 Примеры тестовых заданий

1. Какие из следующих типов данных являются встроенными в C?
  - a) int
  - b) string
  - c) double
  - d) class
2. Переменная в программировании – это \_\_\_\_\_
3. Как объявить константу в C?
  - a) Используя ключевое слово "var"
  - b) Используя знак равенства (=)
  - c) Используя ключевое слово "const"
  - d) Используя ключевое слово "const" и указав тип данных
4. Какой тип данных используется для хранения целых чисел в C?
  - a) float
  - b) int

- c) char  
d) bool
5. Функции scanf() и printf() используются в С, соответственно, для \_\_\_\_\_
6. Для объявления переменной x целого типа (без инициализации значения) следует использовать выражение \_\_\_\_\_
7. Для объявления переменных вещественного типа и вещественного типа двойной точности в С используются, соответственно, типы данных \_\_\_\_\_
8. Какова в С размерность (в байтах) типов данных, соответственно, float и double?  
\_\_\_\_\_
9. Для выполнения математических операций сложения и вычитания в С используются, соответственно, операторы \_\_\_\_\_
10. Для выполнения математических операций умножения и деления в С используются, соответственно, операторы \_\_\_\_\_
11. Для выполнения операций сдвига влево и сдвига вправо в С используются, соответственно, операторы \_\_\_\_\_
12. Как в С объявить и инициализировать переменную x целого типа, чтобы она содержала значение 10? \_\_\_\_\_
13. Для объявления переменных x и y логического типа (без инициализации значения) следует использовать выражение \_\_\_\_\_
14. Для определения в С условия "a больше 0" в операторе if используется выражение \_\_\_\_\_
15. Что означает в С выражение "sizeof(int)"? \_\_\_\_\_
16. Какое из следующих чисел является вещественным числом?  
a) 42  
b) B5  
c) -7  
d) 0.5
17. Объявление в С константы pi вещественного типа с точностью до 2 знаков после запятой \_\_\_\_\_
18. Как в С объявить массив целых чисел с именем arr с 5 элементами? \_\_\_\_\_
19. Как получить значение элемента массива по его индексу в С?  
a) arr(index)  
b) arr[index]  
c) arr.element(index)  
d) arr.get(index)
20. Какой оператор используется для выполнения условных операций в С?  
a) for  
b) if  
c) while  
d) switch
21. Какой цикл используется для выполнения повторяющихся операций, пока условие истинно в С?  
a) do-while  
b) for  
c) while  
d) repeat-until
22. В С при использовании оператора a++ значение переменной a \_\_\_\_\_
23. В С при использовании оператора a-- значение переменной a \_\_\_\_\_
24. Команда break в С используется для \_\_\_\_\_
25. Цикл типа while используется в С для выполнения \_\_\_\_\_
26. Оператор "/" используется в С для выполнения операции \_\_\_\_\_
27. Для проверки двух условий a>0 и b>0, где оба условия должны быть истинными для выполнения действия, используется выражение \_\_\_\_\_

28. Для проверки двух условий  $a > 0$  и  $b > 0$ , где хотя бы одно из условий должно быть истинным для выполнения действия, используется выражение \_\_\_\_\_
29. Какой оператор используется для выполнения определенной операции над каждым элементом массива в C?
- a) apply
  - b) for
  - c) each
  - d) iterate
30. Оператор % в C возвращает \_\_\_\_\_
31. Какой оператор используется для выполнения кода многократно на основе условия в C?
- a) for
  - b) if
  - c) while
  - d) switch
32. Как объявить функцию в C?
- a) `function myFunction();`
  - b) `void myFunction;`
  - c) `int myFunction();`
  - d) `define myFunction();`
33. Перегруженные функции в C – это функции \_\_\_\_\_
34. Что такое область действия переменных в C? \_\_\_\_\_
35. Как объявить локальную переменную внутри функции в C?
- a) `local x;`
  - b) `variable x;`
  - c) `int x;`
  - d) `x = 5;`
36. Глобальная переменная в C – это переменная, которая \_\_\_\_\_
37. Передать значение в функцию в C можно \_\_\_\_\_
38. Для определения условий в C используются операторы \_\_\_\_\_
39. Что такое условная компиляция программы в C? \_\_\_\_\_
40. Как объявить макрос в C с использованием директивы #define?
- a) `macro x = 5;`
  - b) `define x(5);`
  - c) `#define x 5`
  - d) `const x = 5;`
41. Как объявить глобальную переменную в C? \_\_\_\_\_
42. Каким образом можно передать адрес переменной в функцию в C? \_\_\_\_\_
43. Оператор return используется в C \_\_\_\_\_
44. Указатель в языке программирования C – это \_\_\_\_\_
45. Какой оператор используется для получения значения, на которое указывает указатель в C?
- a) \*
  - b) &
  - c) ->
  - d) .
46. Как освободить динамическую память после использования в C? \_\_\_\_\_
47. Что такое memory leak в программировании? \_\_\_\_\_
48. Что такое утечка памяти в программировании? \_\_\_\_\_
49. К чему приводит утечка памяти? \_\_\_\_\_
50. Какие функции используются для работы с динамической памятью в C? \_\_\_\_\_
51. Как можно передать указатель на функцию в качестве параметра в C++? \_\_\_\_\_
52. Операция, невозможная при работе с указателями, это \_\_\_\_\_

53. Какая из следующих функций используется для изменения размера динамически выделенной памяти в C?
- a) `resize()`
  - b) `realloc()`
  - c) `enlarge()`
  - d) Нет такой функции
54. Как передать параметр по ссылке в C?
- a) С помощью указателя
  - b) С помощью оператора "&"
  - c) Объявив функцию с амперсандом (&) после типа параметра
  - d) Объявив параметр с ключевым словом "reference"
55. Как передать параметр по указателю в C?
- a) Объявив функцию с амперсандом (&) после типа параметра
  - b) Объявив параметр с ключевым словом "pointer"
  - c) Объявив параметр с типом указателя
  - d) С помощью оператора "\*"
56. Какой оператор используется для получения адреса переменной в C?
- a) &
  - b) \*
  - c) ->
  - d) .
57. Оператор -> используется в C для доступа к членам структуры \_\_\_\_\_
58. Какой из следующих типов файлов содержит данные в виде текста?
- a) Двоичный файл
  - b) Текстовый файл
  - c) Исполняемый файл
  - d) Системный файл
59. Оператор `FILE *file = fopen("file.txt", "r");` выполняет в C открытие файла \_\_\_\_\_
60. Оператор `FILE *file = fopen("file.txt", "w");` выполняет в C открытие файла \_\_\_\_\_
61. Как закрыть файл после работы с ним в C?
- a) `fcloseFile("file.txt");`
  - b) `close("file.txt");`
  - c) `fclose(file);`
  - d) `exit();`
62. Как проверить, успешно ли открыт файл в C?
- a) `check(file);`
  - b) `if (file != NULL)`
  - c) `open(file);`
  - d) `validate(file);`
63. Как записать строку в текстовый файл в C?
- a) `file.writeLine("Hello");`
  - b) `file.output("Hello");`
  - c) `fprintf(file, "Hello");`
  - d) `file`
64. Оператор `FILE *file = fopen("binary.dat", "rb+");` выполняет в C открытие двоичного файла \_\_\_\_\_
65. Какой режим открытия файла следует использовать для записи двоичных данных в C?
- a) `ios::in`
  - b) `ios::out`
  - c) "wb" (для `fopen`)
  - d) `ios::text`
66. Оператор `feof(file)` выполняет в C проверку достижения \_\_\_\_\_

67. Функция `fseek` используется в C для перемещения внутри файла \_\_\_\_\_
68. В языке C библиотека `string.h` используется для \_\_\_\_\_
69. Символ `\0` (нулевой символ) в C \_\_\_\_\_
70. Для копирования строки `string1` в строку `string2` в C следует использовать оператор \_\_\_\_\_, состоящий из функции с указанием аргументов
71. Оператор `strlen(string)`; в C позволяет найти \_\_\_\_\_
72. Для добавления к строке `string1` строки `string2` в C следует использовать оператор \_\_\_\_\_, состоящий из функции с указанием аргументов
73. Каким образом можно сравнить две строки в C?
- a) `compare(string1, string2)`;
  - b) `strcmp(string1, string2)`;
  - c) `isEqual(string1, string2)`;
  - d) `strcmp(string1, string2)`;
74. Для поиска первого вхождения подстроки `substring` в строке `string` в C следует использовать оператор \_\_\_\_\_, состоящий из функции с указанием аргументов
75. Оператор `atoi(string)`; в C преобразует строку в \_\_\_\_\_
76. Функция `strcpy(char* str1, const char* str2)` в языке C выполняет \_\_\_\_\_
77. Стандартное заголовочное имя `time.h` в C/C++ используется для работы с \_\_\_\_\_
78. Стандартное имя структуры данных `struct tm` используется в C/C++ для представления \_\_\_\_\_
79. Функция `time(NULL)` в C/C++ позволяет получить \_\_\_\_\_
80. Структура `timeInfo.tm_wday` в C/C++ позволяет получить для заданной даты \_\_\_\_\_
81. Как объявить функцию с переменным количеством параметров в C/C++?
- a) `void myFunction(args...)`
  - b) `myFunction(args...)`
  - c) `void myFunction(...)`
  - d) `myFunction(...)`
82. Для передачи переменного количества аргументов в функцию в C/C++ следует использовать \_\_\_\_\_
83. Как получить аргументы с переменным количеством параметров внутри функции в C/C++?
- a) использовать макросы `VAR_ARGS`
  - b) преобразовать их в строку
  - c) использовать функции `va_start` и `va_arg`
  - d) передать их как структуру
84. Каким образом можно создать битовое поле в C/C++? \_\_\_\_\_
85. Какой тип данных используется для определения битового поля в C/C++? \_\_\_\_\_
86. Сколько битов выделяется под битовое поле типа `unsigned int` в следующем объявлении: `unsigned int flag : 4;`?
- a) 1 бит
  - b) 2 бита
  - c) 4 бита
  - d) 8 битов
87. Каким образом можно установить значение битового поля в C/C++?
- a) использовать функцию `setBitfield()`
  - b) присвоить значение с помощью оператора `"="`
  - c) использовать операторы битовых масок и сдвигов
  - d) вызвать функцию `set()`
88. Что такое исключение? \_\_\_\_\_
89. Каким образом можно сгенерировать исключение в языке C?
- a) Используя оператор `throw`
  - b) Используя оператор `catch`
  - c) Используя оператор `try`

- d) Используя функцию `longjmp`
90. Оператор `setjmp-longjmp` используется в языке C для \_\_\_\_\_
91. Что произойдет, если исключение не будет поймано с помощью функции `setjmp` и `longjmp`? \_\_\_\_\_
92. Какой оператор используется для завершения выполнения текущей функции и генерации исключения в языке C?
- `throw`
  - `longjmp`
  - `return`
  - `exit`
93. Как можно передать информацию о произошедшем исключении в языке C? \_\_\_\_\_
94. Что такое сигнал в контексте операционной системы? \_\_\_\_\_
95. Сигнал в языке C можно обработать с помощью \_\_\_\_\_
96. Что такое нелокальный переход (`longjmp`) в языке C? \_\_\_\_\_
97. Как можно выполнить нелокальный переход с использованием функции `longjmp`? \_\_\_\_\_
98. Какие аргументы принимает функция `setjmp` в языке C?
- Ни одного аргумента
  - Один аргумент типа `int`
  - Один аргумент типа `jmp_buf`
  - Два аргумента типа `int`
99. Функция `longjmp` в языке C принимает два аргумента \_\_\_\_\_
100. Каким образом можно использовать нелокальные переходы в языке C? \_\_\_\_\_

*Краткие методические указания*

Промежуточный тест проводится во время последнего в учебном периоде лабораторного занятия. Тест состоит из двух тестовых заданий. На выполнение теста отводится 90 минут. Во время проведения теста использование литературы и других информационных ресурсов допускается только по предварительному согласованию с преподавателем.

*Шкала оценки*

№	Баллы	Описание
5	19–20	Студент демонстрирует умения на итоговом уровне: умеет свободно выполнять тестовые задания, свободно оперирует приобретенными умениями, применяет их в ситуациях повышенной сложности.
4	16–18	Студент демонстрирует умения на среднем уровне: освоил основные умения, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе умений на новые, нестандартные ситуации.
3	13–15	Студент демонстрирует умения и навыки на базовом уровне: в ходе выполнения тестовых заданий допускаются значительные ошибки, проявляется отсутствие отдельных умений, навыков по дисциплинарным компетенциям, испытываются значительные затруднения при оперировании умениями и при их переносе на новые ситуации.
2	9–12	Студент демонстрирует умения и навыки на уровне ниже базового: проявляется недостаточность умений и навыков.
1	0–8	Студентом проявляется полное или практически полное отсутствие умений и навыков.