

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Галунин Сергей Александрович  
Должность: проректор по учебной работе  
Дата подписания: 23.12.2025 13:42:26  
Уникальный программный ключ:  
08ef34338325bdb0ac5a47baa5472ce36cc3fc3b

Приложение к ОПОП  
«Разработка программно-  
информационных систем»



**СПбГЭТУ «ЛЭТИ»**  
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

МИНОБРНАУКИ РОССИИ

федеральное государственное автономное образовательное учреждение высшего образования  
**«Санкт-Петербургский государственный электротехнический университет  
«ЛЭТИ» им. В.И.Ульянова (Ленина)»  
(СПбГЭТУ «ЛЭТИ»)**

---

**РАБОЧАЯ ПРОГРАММА**

ДИСЦИПЛИНЫ

**«ПРОГРАММИРОВАНИЕ»**

для подготовки бакалавров

по направлению

09.03.04 «Программная инженерия»

по профилю

**«Разработка программно-информационных систем»**

Санкт-Петербург

2025

## **ЛИСТ СОГЛАСОВАНИЯ**

Разработчики:

доцент, к.т.н., доцент Заславский М.М.

Рабочая программа рассмотрена и одобрена на заседании кафедры МОЭВМ  
20.01.2025, протокол № 1

Рабочая программа рассмотрена и одобрена учебно-методической комиссией  
ФКТИ, 28.01.2025, протокол № 1

Согласовано в ИС ИОТ

Начальник ОМОЛА Загороднюк О.В.

## 1 СТРУКТУРА ДИСЦИПЛИНЫ

Обеспечивающий факультет	ФКТИ
Обеспечивающая кафедра	МОЭВМ

Общая трудоемкость (ЗЕТ)	11
Курс	1
Семестр	2, 1

### **Виды занятий**

Лекции (академ. часов)	34
Электронные лекции (акад. часов)	34
Электронные лабораторные (академ. часов)	68
Иная контактная работа (академ. часов)	6
Все контактные часы (академ. часов)	40
Самостоятельная работа, включая часы на контроль (академ. часов)	254
Всего (академ. часов)	396

### **Вид промежуточной аттестации**

Экзамен (курс)	1
Курсовая работа (курс)	1
Экзамен (курс)	1
Курсовая работа (курс)	1

## **2 АННОТАЦИЯ ДИСЦИПЛИНЫ**

### **«ПРОГРАММИРОВАНИЕ»**

Дисциплина нацелена на изучение и освоение базовых понятий, методов и приемов программирования на языке программирования C / C++ в основном в парадигме процедурного программирования. Представляет программирование как систематическую научно-практическую деятельность, носящую массовый характер (производство программ заданного качества в заданные сроки). Выполняя задания, студенты получают навык компиляции и отладки программ.

### **SUBJECT SUMMARY**

#### **«PROGRAMMING»**

Discipline is aimed at studying and mastering the basic concepts, methods and techniques of programming in the programming language C / C ++, mainly in the paradigm of procedural programming. Represents programming as a systematic scientific and practical activity, which is of a mass nature (production of programs of specified quality at the given time). While completing assignments, students get the compilation and debugging skills.

## **3 ОБЩИЕ ПОЛОЖЕНИЯ**

### **3.1 Цели и задачи дисциплины**

1. Целью дисциплины является получение теоретических знаний и практических навыков реализации и применения основных методов и приемов разработки структурированных алгоритмов и программ; основных процедурных элементов языка C/C++; основ работы с файлами и консольным вводом-выводом; способов записи и документирования алгоритмов и программ, в том числе рекомендуемых правил оформления кода; способов тестирования и отладки разработанных программ; основных методов и средств разработки типовых алгоритмов и программ со структурированными типами данных, определяемыми пользователем, в том числе требующими явного управления памятью; способов представления и реализации линейных списков; структур данных стек, очередь и различных способов реализации; подходов к обработке бинарных данных на примере BMP-файлов.

2. Задачи дисциплины: формирование: знаний по применению методов структурной декомпозиции типовых задач для разделения программы на части и ее реализации как совокупности функций; умений самостоятельно составлять, тестировать, отлаживать и документировать программы; навыков пользования средой программирования для конструирования, тестирования и отладки программ; навыков формулирования функциональных требований к ключевым фрагментам программ, анализа кода с точки зрения выполнения этих требований; навыков использования репозитория для контроля версий кода.

3. Знания:

- основных методов и приемов разработки структурированных алгоритмов и программ;
- основных процедурных элементов языка C/C++: стандартные типы данных;

операторы языка, реализующие основные управляющие структуры; функции (подпрограммы);

- основ работы с файлами и консольным вводом-выводом;
- способов записи и документирования алгоритмов и программ, в том числе рекомендуемых правил оформления кода;
- способов тестирования и отладки разработанных программ;
- основных методов и средств разработки типовых алгоритмов и программ со структурированными типами данных, определяемыми пользователем, в том числе требующими явного управления памятью;
- способов представления и реализации линейных списков;
- структур данных стек, очередь и различных способов реализации;
- подходов к обработке бинарных данных на примере BMP-файлов.

#### 4. Формирование умений:

- по применению методов структурной декомпозиции типовых задач для разделения программы на части и ее реализации как совокупности функций;
- самостоятельно составлять, тестировать, отлаживать и документировать программы;
- пользования средой программирования для конструирования, тестирования и отладки программ;
- формулирования функциональных требований к ключевым фрагментам программ, анализа кода с точки зрения выполнения этих требований;
- использования репозитория для контроля версий кода.

#### 5. Освоение языка программирования в необходимом для решения типовых задач объеме.

##### Освоение навыков:

- чтения, анализа, выделения главной идеи, понимания и модификации типовых программ, включающих изучаемые конструкции языка программирования и работающих со структурированными данными;

- работы со стандартным окружением разработчика в ОС Linux;
- работы со справочными руководствами (man-страницами), посвященными стандартной библиотеке языка Си.

### **3.2 Место дисциплины в структуре ОПОП**

Дисциплина изучается на основе знаний, полученных при освоении школьной программы.

и обеспечивает изучение последующих дисциплин:

1. «Алгоритмы и структуры данных»
2. «Объектно-ориентированное программирование»
3. «Организация ЭВМ и систем»
4. «Основы программирования на языке Ассемблера»
5. «Основы тестирования программного обеспечения»
6. «Вычислительная математика»
7. «Низкоуровневое программирование»
8. «Операционные системы»
9. «Построение и анализ алгоритмов»
10. «Тестирование программного обеспечения»
11. «Учебная практика (технологическая (проектно-технологическая) практика)»
12. «Web-технологии»
13. «Базы данных»
14. «Компьютерная графика»
15. «Мобильная разработка на Android ч.1»

16. «Основы бэкенд-разработки на Python ч.1»
17. «Основы фронтенд-разработки ч.1»
18. «Параллельные алгоритмы»
19. «Программирование логических интегральных схем»
20. «Сети и телекоммуникации»
21. «Логическое программирование»
22. «Мобильная разработка на Android ч.2»
23. «Основы промышленной разработки программного обеспечения»
24. «Промышленное тестирование программного обеспечения»
25. «Распределенные алгоритмы»
26. «Технологии хранения данных»
27. «Основы компьютерного зрения»
28. «Проектирование человеко-машинного интерфейса»
29. «Спецификация, проектирование и архитектура программных систем»
30. «Визуализация данных»
31. «Инженерный документооборот»
32. «Интеллектуальные системы»
33. «Качество и метрология программного обеспечения»
34. «Разработка приложений для мобильных платформ»
35. «Управление разработкой и экономика программного проекта»
36. «Цифровая обработка изображений»



### 3.3 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения образовательной программы обучающийся должен достичь следующие результаты обучения по дисциплине:

Код компетенции/ индикатора компетенции	Наименование компетенции/индикатора компетенции
ОПК-5	Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем;
<i>ОПК-5.3</i>	<i>Имеет навыки установки программного и аппаратного обеспечения информационных и автоматизированных систем</i>
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов;
<i>ОПК-6.1</i>	<i>Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий</i>
ПК-4	Способен оценивать временную и емкостную сложность программного обеспечения
<i>ПК-4.1</i>	<i>Знает методы оценки временной и емкостной сложности программного обеспечения</i>
<i>ПК-4.2</i>	<i>Умеет вычислять временную и емкостную сложность</i>
<i>ПК-4.3</i>	<i>Имеет навыки оценки временной и емкостной сложности</i>

## 4 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

### 4.1 Содержание разделов дисциплины

#### 4.1.1 Наименование тем и часы на все виды нагрузки

№ п/п	Наименование темы дисциплины	Лек, ач	ЭЛек, ач	Элаб, ач	ИКР, ач	СР, ач
1	Введение	1	1			
2	Тема 1. Способы написания программ, компиляция, используемые инструменты	1	1			10
3	Тема 2. Стандартные (базовые) типы данных, операторы, выражения	1	1	8		12
4	Тема 3. Ввод и вывод в языке	1	1			10
5	Тема 4. Основные управляющие структуры программирования и управляющие инструкции в языке С	1	1	10		12
6	Тема 5. Функции и структура программы	2	2			10
7	Тема 6. Обработка строк и текстов	2	2	8		10
8	Тема 7. Сборка программ в языке С	1	1			10
9	Тема 8. Указатели и массивы	2	2	8		10
10	Тема 9. Введение в структуры в С	2	2			10
11	Тема 10. Стандартная библиотека языка С	2	2			10
12	Заключение	1	1		3	35
13	2 семестр					
14	Введение	1	1			
15	Тема 1. Регулярные выражения, их структура и применение для работы с текстовыми данными	2	2			10
16	Тема 2. Записи (структуры) и их реализация в языке программирования	2	2	8		10
17	Тема 3. Разновидности линейных списков	1	1			10
18	Тема 4. Указатели, структуры рекурсивные типы данных	2	2	8		10
19	Тема 5. Абстрактные типы данных (АТД)	2	2			10
20	Тема 6. Организация ввода/вывода и работа с файлами	2	2	8		10
21	Тема 7. Жизненный цикл разработки программ	2	2			10
22	Тема 8. Идея абстрактных структур данных. Реализации АТД при помощи классов С++	2	2	10		10
23	Заключение	1	1		3	35
	Итого, ач	34	34	68	6	254
	Из них ач на контроль	0	0	0	0	70
	Общая трудоемкость освоения, ач/зе	396/11				

### 4.1.2 Содержание

№ п/п	Наименование темы дисциплины	Содержание
1	Введение	Предмет дисциплины и ее задачи. Содержание и форма проведения занятий. Связь с другими дисциплинами учебного плана направления.
2	Тема 1. Способы написания программ, компиляция, используемые инструменты	Языки программирования и системы программирования. История создания языков С и С++. Краткая характеристика языка С. Структура и основные элементы программы. Пример программы на языке С. Компиляция и интерпретация программ. Выполнение программы. Стандарты языка С. Системы программирования на языке С для ОС Linux.
3	Тема 2. Стандартные (базовые) типы данных, операторы, выражения	Простые стандартные типы данных (схема рассмотрения: множество значений, переменные, константы, набор операций, выражения, битовое представление, побитовые операторы, оператор присваивания). Целочисленные типы, особенности представления и вычислений. Символьный тип. Особенности реализаций. Способы использования символьного типа в программах. Булевский (логический) тип, особенности логических выражений. Сокращенное (неполное) вычисление логических выражений. Вещественные типы с плавающей точкой. Особенности машинного представления вещественных чисел. Свойства машинной арифметики. Машинное epsilon. Выражения. Преобразования типов. Приведение типов. Специальные операторы в языке С (инкремент и декремент, составной оператор присваивания, тернарный условный оператор, оператор запятая и т.п.). Объявления и описания переменных, инициализация переменных, квалификатор const. Поразрядные операции.
4	Тема 3. Ввод и вывод в языке	Ввод и вывод значений стандартных типов с использованием стандартных потоков. Простейшая отладка.
5	Тема 4. Основные управляющие структуры программирования и управляющие инструкции в языке С	Основные управляющие структуры программирования: последовательность, выбор (ветвление), итерации (циклы). Семантика управляющих структур и инструкции языка С++. Операторы и блоки. Выбор вариантов: оператор (инструкция) if, расширение оператора if, множественный выбор. Итерации: цикл с предусловием, цикл с постусловием, цикл со счетчиком. Другие управляющие операторы. Примеры преобразования управляющих структур. Структурное программирование.

№ п/п	Наименование темы дисциплины	Содержание
6	Тема 5. Функции и структура программы	Создание и использование функций. Вызов функции (аргументы функции) и возврат значения. Области видимости (области действия). Глобальные и локальные переменные. Области видимости и классы памяти. Внешние переменные. Автоматические переменные. Статические переменные. Представление программы в виде набора функций.
7	Тема 6. Обработка строк и текстов	Строки. Определение, операции над строками. Представление строк (с явной длиной, с символом ограничителем). Реализация типовых операций над строками (анализ и редактирование строк).
8	Тема 7. Сборка программ в языке C	Многофайловая структура программы. Семейство компиляторов gcc. Этапы сборки программы: препроцессор, компилятор, линковщик. Директивы препроцессора. Частые ошибки сборки программ. Утилита make. Создание Makefile.
9	Тема 8. Указатели и массивы	Массивы. Индексирование. Объявление массивов. Двумерные и многомерные массивы. Ввод-вывод массивов. Строки и тексты как массивы символов. Введение в указатели. Связь массивов и указателей. Арифметика указателей. Передача аргумента в функцию по ссылке и по значению. Статическая и динамическая память. Динамические массивы строк. Размещение данных в памяти: статическое, автоматическое и динамическое выделение памяти. Примеры. Управление памятью во время выполнения программы в языке C.
10	Тема 9. Введение в структуры в C	Структуры struct. Определение структуры. Доступ к полям структуры. Записи со структурированными полями. Указатель на структуру. Массивы структур. Примеры использования.
11	Тема 10. Стандартная библиотека языка C	Линейный и бинарный поиск в массиве. Оптимальность алгоритма бинарного поиска. Сортировка массива чисел. Сортировка массива структур. Компараторы. Функции обработки строк стандартной библиотеки. Функции работы со временем. Генерация псевдослучайной величины. Функции для управления процессом выполнения программы.
12	Заключение	Повторение основных тем семестра. Связь с учебной дисциплиной по программированию следующего семестра.
13	2 семестр	
14	Введение	Предмет дисциплины и ее задачи. Содержание и форма проведения занятий. Связь с дисциплиной «Программирование» (1-го семестра). Обзор структуры курса 2-го семестра. Связь дисциплины «Программирование» с другими дисциплинами учебного плана направления.

№ п/п	Наименование темы дисциплины	Содержание
15	Тема 1. Регулярные выражения, их структура и применение для работы с текстовыми данными	Область применения регулярных выражений, примеры практического использования. Структура регулярных выражений, метасимволы, группы. Разновидности регулярных выражений. Использование регулярных выражений для поиска файлов средствами ОС Linux. Использование регулярных выражений при работе с текстом в программах на языке C, примеры.
16	Тема 2. Записи (структуры) и их реализация в языке программирования	Типы записей в языке (структуры struct и объединения union). Определение структуры. Доступ к полям структуры. Записи со структурированными полями. Массивы записей, поиск данных по составным критериям в массиве записей. Примеры использования. Размеченное объединение множеств, записи с вариантами в языках программирования, объединение union в языке C. Поле тега в структуре, содержащей объединение. Рекомендации по применению.
17	Тема 3. Разновидности линейных списков	Разновидности линейных списков: однонаправленные (Л1-списки), двунаправленные (Л2-списки), циклические списки. Реализация на языке C. Изучение основных операций над списками: удаление, вставка. Особенности операций в зависимости от места в списке. Примеры решения задач с линейными списками.
18	Тема 4. Указатели, структуры рекурсивные типы данных	Структуры и указатели при работе с динамической памятью. Определяемые рекурсивные типы данных. Цепное представление последовательностей. Линейный список с произвольным доступом. Представление в связанной памяти. Программирование основных операций.
19	Тема 5. Абстрактные типы данных (АТД)	Способы реализации списка как АТД в языке C с использованием структур, указателей и рекурсивных типов данных (в парадигме процедурного программирования). Функциональная спецификация Л1-списка. Типовые способы представления и реализации линейных списков: непрерывная реализация на базе массива, ссылочная реализация в динамической памяти. Представление и реализация Л1-списка на языке C
20	Тема 6. Организация ввода/вывода и работа с файлами	Файловые системы языков C. Потоки и файлы. Последовательность, как модель потока. Стандартные потоки ввода и вывода. Ввод и вывод встроенных типов. Файловые потоки. Типовые действия с файлами: генерация, чтение, копирование. Форматирование ввода и вывода. Схема однократных алгоритмы обработки входных потоков (файлов) – вычисление функций на последовательностях. Примеры. Работа с файловой системой из программ на языке C.

№ п/п	Наименование темы дисциплины	Содержание
21	Тема 7. Жизненный цикл разработки программ	Жизненный цикл программы. Анализ требований, проектирование, реализация, сопровождение. Анализ требований и постановка задачи. Формализация постановки задачи. Спецификация задачи (программы). Анализ и описание диалога. Сценарий диалога. Пример спецификации диалоговой программы. Особенности и этапы проектирования. Верификация и аттестация программ. Испытание (тестирование и отладка) программ. Модульное тестирование. Сквозные мероприятия: документирование, подготовка и планирование испытаний. Тестирование программ, использующих динамическую память, структуры и указатели. Разработка интерфейса командной строки.
22	Тема 8. Идея абстрактных структур данных. Реализации АДТ при помощи классов C++	Общие идеи классов как способа представления и реализации АДТ или типов, определяемых пользователем. Классы и объекты. Определение класса, использование класса, вызов методов класса. Конструкторы и деструкторы. Отделение интерфейса от реализации. Абстракция данных и скрытие информации. Способы реализации стека и очереди на основе линейного списка и массива как АДТ в языке C++. Класс Node, List, Queue. Разработка класса линейных списков. Базовые операции Л1-списка.
23	Заключение	Развитие и реализация идей объектно-ориентированного программирования в языках программирования. Инкапсуляция. Наследование. Полиморфизм. Параметризованные типы данных. Связь с последующими изучаемыми дисциплинами по программированию.

## 4.2 Перечень лабораторных работ

Наименование лабораторной работы	Количество ауд. часов
1. Управляющие конструкции языка C (1 семестр)	8
2. Сборка программ в языке C	10
3. Указатели и массивы в языке C	8
4. Стандартная библиотека языка C	8
5. Использование регулярных выражений в программах на языке C (2 семестр)	8
6. Реализация линейных списков на языке C	8
7. Работа с файлами и файловой системой в программах на языке C	8
8. Реализация динамических структур данных стек и очередь на языке C	10
Итого	68

### **4.3 Перечень практических занятий**

Практические занятия не предусмотрены.

### **4.4 Курсовое проектирование**

Цель работы (проекта): Приобретение навыков разработки, тестирования и отладки комплектной программы на языке программирования Си, выполняющую действия и имеющую функции, определенные индивидуальным заданием

Содержание работы (проекта): В процессе освоения дисциплины обучающиеся выполняют 2 курсовых работы в 1 и 2 семестрах.

Курсовая работа состоит из следующих этапов:

1. Анализ задачи, цели, алгоритм решения задачи;
2. Разработка программы;
3. Выполнение исследуемых алгоритмов на сгенерированных наборах данных;
4. Фиксация результатов тестовых испытаний программы;
5. Отладка программы.

Студенту на шестой неделе семестра выдается задание на определенную тему. Защита курсовой работы происходит на последней неделе семестра. Результаты выполнения курсовой работы представляются в виде пояснительной записки.

Пояснительная записка к курсовой работе оформляется в соответствии с шаблоном оформления курсовой работы, размещённым на сайте вуза. Помимо листов, предусмотренных шаблоном (титульный, лист задания, аннотация, содержание, введение, выводы), в отчёте должны быть:

- 1) Аннотация - краткое содержание/краткая характеристика работы. В аннотации к курсовой работе необходимо указать следующее: краткое описание задания, информацию о демонстрации работы программы.

2) Цель - одно предложение, кратко характеризующее желаемый конечный результат, который планируется достигнуть в ходе работы. Примерный шаблон: “Целью данной работы является изучение... создание... разработка...”.

3) Задачи - список действий для достижения цели работы.

4) Пояснительная записка (ПЗ) должна содержать подробное описание выполненной вами работы, программной реализации той или иной функции. ПЗ должна включать в себя обоснование принятых вами решений, изображения и, при необходимости, таблицы для более наглядного объяснения.

5) Примеры работы программы. Каждый пример должен содержать заголовок, в котором описано, какая конкретно функция проверяется. Все функции, реализованные в курсовой работе, должны быть представлены в примерах работы программы. Примеры работы программы должны быть представлены хорошо читаемым кодом.

Критерии оценивания курсовой работы:

Неудовлетворительно - студент испытывает серьезные трудности при ответе на вопросы преподавателя, не может объяснить ход решения задачи курсовой работы, не может решить дополнительную задачу, выданную преподавателем.

Удовлетворительно - студент в целом объясняет ход решения задачи, но на некоторые вопросы преподавателя отвечает неточно, допускает несущественные ошибки при решении дополнительной задачи.

Хорошо - студент объясняет ход решения задачи курсовой работы, но испытывает затруднения в объяснении выбора некоторых решений. Умеет решать дополнительные задачи.

Отлично - студент свободно объясняет ход решения задачи курсовой работы, способен применять полученные знания при решении дополнительных задач, аргументирует выбор решений, свободно владеет теорией

Темы:



№ п/п	Название темы	Перевод темы
1	Темы 1 семестра	
2	Обработка текстовой информации. Реализовать программу, которая должна найти и удалить все повторно встречающиеся предложения	Text data processing. Implement a program that should find and remove all duplicate sentences
3	Обработка текстовой информации. Реализовать программу, которая должна заменить все цифры в предложениях на введенную строку.	Text data processing. Implement a program that should replace all numbers in sentences with the entered string
4	Обработка текстовой информации. Реализовать программу, которая должна удалить все предложения, в которых есть три подряд идущие буквы в верхнем регистре	Text data processing. Implement a program that should remove all sentences that have three consecutive uppercase letters
5	Обработка текстовой информации. Реализовать программу, которая должна отсортировать по уменьшению количества слов начинающихся с гласной буквы	Text data processing. Implement a program that should sort by decreasing the number of words starting with a vowel
6	Обработка текстовой информации. Реализовать программу, которая должна отсортировать предложения по уменьшению количества слов длина которых равняется 3	Text data processing. Implement a program that should sort sentences by decreasing the number of words whose length is 3
7	Темы 2 семестр	
8	Обработка изображений. Реализовать программу, которая заменяет все пиксели одного заданного цвета на другой цвет	Image processing. Implement a program that replaces all pixels of one given color with another color
9	Обработка изображений. Реализовать программу, которая разделяет изображение на N*M частей	Image processing. Implement a program that divides an image into N*M parts
10	Обработка изображений. Реализовать программу, которая осуществляет поворот изображения на 90/180/270 градусов	Image processing. Implement a program that rotates an image 90/180/270 degrees
11	Обработка изображений. Реализовать программу, которая находит самый часто встречаемый цвет и заменяет его на другой заданный цвет	Image processing. Implement a program that finds the most frequently encountered color and replaces it with another given color
12	Обработка изображений. Реализовать программу, которая находит самый большой прямоугольник заданного цвета и перекрашивает его в другой цвет	Image processing. Implement a program that finds the largest rectangle of a given color and recolors it to a different color

## 4.5 Реферат

Реферат не предусмотрен.

#### **4.6 Индивидуальное домашнее задание**

Индивидуальное домашнее задание не предусмотрено.

#### **4.7 Доклад**

Доклад не предусмотрен.

#### **4.8 Кейс**

Кейс не предусмотрен.

#### **4.9 Организация и учебно-методическое обеспечение самостоятельной работы**

Изучение дисциплины сопровождается самостоятельной работой студентов с рекомендованными преподавателем литературными источниками и информационными ресурсами сети Интернет.

Планирование времени для изучения дисциплины осуществляется на весь период обучения, предусматривая при этом регулярное повторение пройденного материала. Обучающимся, в рамках внеаудиторной самостоятельной работы, необходимо регулярно дополнять сведениями из литературных источников материал, законспектированный на лекциях. При этом на основе изучения рекомендованной литературы целесообразно составить конспект основных положений, терминов и определений, необходимых для освоения разделов учебной дисциплины.

Особое место уделяется консультированию, как одной из форм обучения и контроля самостоятельной работы. Консультирование предполагает особым образом организованное взаимодействие между преподавателем и студентами, при этом предполагается, что консультант либо знает готовое решение, которое он может предписать консультируемому, либо он владеет способами деятель-

ности, которые указывают путь решения проблемы.

Самостоятельное изучение студентами теоретических основ дисциплины обеспечено необходимыми учебно-методическими материалами (учебники, учебные пособия, конспект лекций и т.п.), выполненными в печатном или электронном виде.

Изучение студентами дисциплины сопровождается проведением регулярных консультаций преподавателей, обеспечивающих практические занятия по дисциплине, за счет бюджета времени, отводимого на консультации (внеаудиторные занятия, относящиеся к разделу «Самостоятельные часы для изучения дисциплины»).

*В случае применения ДОТ с заменой аудиторных занятий:*

Самостоятельной записи на курс нет. Студент заходит на курс, используя логин/пароль от единой учетной записи университета (единый логин и пароль). Каждую неделю будет доступна новая тема курса: видеолекции, кратко раскрывающие содержание каждой темы, презентации и конспекты, с которыми обучающиеся смогут ознакомиться в любое удобное время. Все темы включают практические занятия, которые предусматривают самостоятельное выполнение заданий, а также задания с автоматической проверкой, результаты которых учитываются при общей аттестации полученных знаний. В конце каждой лекции необходимо пройти небольшой контрольный тест, который покажет насколько усвоен предложенный материал. Рекомендуем изучать материал последовательно, что существенно облегчит работу. У каждого контрольного задания имеется своя форма (тест или практическое задание) есть срок выполнения (окончательный срок), по истечении которого даже правильные ответы система принимать не будет! В расписании курса указан окончательный срок каждого задания, который варьируется от двух до четырех недель в зависимости от его сложности. Весь учебный курс рассчитан на 16 недель. Его итоги будут подведены в течение нескольких недель после его окончания.

<b>Текущая СРС</b>	<b>Примерная трудоемкость, ач</b>
Работа с лекционным материалом, с учебной литературой	60
Опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	0
Самостоятельное изучение разделов дисциплины	0
Выполнение домашних заданий, домашних контрольных работ	0
Подготовка к лабораторным работам, к практическим и семинарским занятиям	54
Подготовка к контрольным работам, коллоквиумам	0
Выполнение расчетно-графических работ	0
Выполнение курсового проекта или курсовой работы	70
Поиск, изучение и презентация информации по заданной проблеме, анализ научных публикаций по заданной теме	0
Работа над междисциплинарным проектом	0
Анализ данных по заданной теме, выполнение расчетов, составление схем и моделей, на основе собранных данных	0
Подготовка к зачету, дифференцированному зачету, экзамену	70
<b>ИТОГО СРС</b>	<b>254</b>

## 5 Учебно-методическое обеспечение дисциплины

### 5.1 Перечень основной и дополнительной литературы, необходимой для освоения дисциплины

№ п/п	Название, библиографическое описание	К-во экз. в библ.
Основная литература		
1	Ивановский, Сергей Алексеевич. Разработка корректных программ [Текст] : Практикум по программированию / С.А.Ивановский, В.А.Калмычков, А.А.Лисс, 2001. -71 с.	93
2	Представление и обработка структурированных данных [Текст] : Практикум по программированию / С.А.Ивановский, В.А.Калмычков, А.А.Лисс, В.П.Самойленко, 2002. -95 с.	192
3	Программирование [Электронный ресурс] : электрон. учеб.-метод. пособие / К. В. Кринкин [и др.], 2018. -1 эл. опт. диск (CD-ROM)	неогр.
Дополнительная литература		
1	Керниган, Брайан В. Язык программирования Си [Текст] : монография / Б.Керниган, Д.Ритчи; Пер. с англ. под ред. В.С.Штаркмана, 2001. -351 с	7
2	Керниган Брайан У. Язык программирования С [Текст] / Б.У. Керниган, Д.М. Ритчи, 2007. -289 с	3
3	Войтов Н. М. Основы работы с Linux. Учебный курс [Электронный ресурс], 2010. -216 с.	неогр

### 5.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», используемых при освоении дисциплины

№ п/п	Электронный адрес
1	Б.В. Керниган, Д.М. Ричи. Язык C <a href="https://nsu.ru/xmlui/bitstream/handle/nsu/9058/kr.pdf">https://nsu.ru/xmlui/bitstream/handle/nsu/9058/kr.pdf</a>

### 5.3 Адрес сайта курса

Адрес сайта курса: <https://vec.etu.ru/moodle/course/view.php?id=10635>

## 6 Критерии оценивания и оценочные материалы

### 6.1 Критерии оценивания

Для дисциплины «Программирование» предусмотрены следующие формы промежуточной аттестации: экзамен.

#### Экзамен

Оценка	Описание
Неудовлетворительно	Курс не освоен. Студент испытывает серьезные трудности при ответе на ключевые вопросы дисциплины
Удовлетворительно	Студент в целом овладел курсом, но некоторые разделы освоены на уровне определений и формулировок
Хорошо	Студент овладел курсом, но в отдельных вопросах испытывает затруднения. Умеет решать задачи
Отлично	Студент демонстрирует полное овладение курсом, способен применять полученные знания при решении конкретных задач.

## Особенности допуска

Допуск к промежуточной аттестации:

- выполнение и защита 4 лабораторных работ в каждом семестре;
- выполнение и защита курсовой работы на оценку не ниже ”удовлетворительно”.

Экзамен проводится по билетам, оценивается в соответствии с критериями оценивания экзамена.

## 6.2 Оценочные материалы для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине

### Вопросы к экзамену

№ п/п	Описание
1	1 семестр
2	Структура и основные элементы программы
3	Стандарты языка C
4	Булевский (логический) тип, особенности логических выражений
5	Машинное эpsilon
6	Ввод и вывод значений стандартных типов с использованием стандартных потоков
7	Операторы и блоки
8	Итерации: цикл с предусловием, цикл с постусловием, цикл со счетчиком
9	Примеры преобразования управляющих структур
10	Глобальные и локальные переменные
11	Автоматические переменные
12	Строки
13	Реализация типовых операций над строками (анализ и редактирование строк)
14	Семейство компиляторов gcc.
15	Утилита make
16	Массивы. Индексирование
17	Арифметика указателей
18	Структуры struct. Определение структуры
19	Линейный и бинарный поиск в массиве
20	Функции обработки строк стандартной библиотеки
21	Функции для управления процессом выполнения программы
22	2 семестр
23	Использование регулярных выражений для поиска файлов средствами ОС Linux
24	Доступ к полям структуры

25	Размеченное объединение множеств, записи с вариантами в языках программирования, объединение union в языке C
26	Разновидности линейных списков: однонаправленные (Л1списки), двунаправленные (Л2списки)
27	Особенности операций в зависимости от места в списке
28	Структуры и указатели при работе с динамической памятью
29	Линейный список с произвольным доступом
30	Способы реализации списка как АД в языке C с использованием структур, указателей и рекурсивных типов данных
31	Функциональная спецификация Л1списка
32	Представление и реализация Л1списка на языке C
33	Тестирование программ, использующих динамическую память, структуры и указатели
34	Ввод и вывод встроенных типов.
35	Файловые потоки.
36	Типовые действия с файлами: генерация, чтение, копирование.
37	Форматирование ввода и вывода
38	Класс Node, List, Queue

### Форма билета

Министерство науки и высшего образования Российской Федерации  
 ФГАОУ ВО «Санкт-Петербургский государственный электротехнический  
 университет «ЛЭТИ» имени В.И. Ульянова (Ленина)»

---

### ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

Дисциплина **Программирование ФКТИ**

1. Доступ к полям структуры.
2. Форматирование ввода и вывода.
3. Задача.

УТВЕРЖДАЮ

Заведующий кафедрой



## **Образцы задач (заданий) для контрольных (проверочных) работ**

### **Примеры задач, выдаваемых на экзамене к билету**

1. Найти и вывести максимальное число из трех целых чисел (числа вводятся с консоли).
2. Найти разницу между числом строчных и прописных (заглавных) букв в строке.
3. Подсчитать количество слов во введенной пользователем строке. Для упрощения задачи считать, что слова разделяются исключительно одним пробелом, а первый и последний символы строки не являются пробелами.

### **Примеры задач, выдаваемых дополнительно при защите лабораторных работ и курсовой работы**

1. Вывести число в поле шириной 10 символов, выровняв его по правому краю
2. Напишите программу, которая в текущей директории находит все одинаковые (по содержимому) текстовые файлы
3. Напишите функцию удаления/вставки элемента в произвольном месте списка

### **Примеры вопросов при защите лабораторных работ и курсовой работы**

1. В каком заголовочном файле стандартной библиотеки расположены функции для работы с символами?
2. Почему частое использование функции `realloc` может замедлять выполнение программы?
3. Как выполняется процесс сборки программы на языке C?

Весь комплект контрольно-измерительных материалов для проверки сформированности компетенции (индикатора компетенции) размещен в закрытой части по адресу, указанному в п. 5.3

### 6.3 График текущего контроля успеваемости

Неделя	Темы занятий	Вид контроля
1	Тема 1. Способы написания программ, компиляция, используемые инструменты Тема 2. Стандартные (базовые) типы данных, операторы, выражения Тема 3. Ввод и вывод в языке	
2		
3		
4		Отчет по лаб. работе
5	Тема 4. Основные управляющие структуры программирования и управляющие инструкции в языке С Тема 5. Функции и структура программы	
6		
7		
8		Отчет по лаб. работе
9	Тема 6. Обработка строк и текстов Тема 7. Сборка программ в языке С	
10		
11		
12		Отчет по лаб. работе
13	Тема 8. Указатели и массивы Тема 10. Стандартная библиотека языка С	
14		
15		
16		Отчет по лаб. работе
17	Заключение	Защита КР / КП
18	Тема 1. Регулярные выражения, их структура и применение для работы с текстовыми данными Тема 2. Записи (структуры) и их реализация в языке программирования	
19		
20		
21		Отчет по лаб. работе
22	Тема 3. Разновидности линейных списков Тема 4. Указатели, структуры рекурсивные типы данных	
23		
24		
25		Отчет по лаб. работе
26	Тема 5. Абстрактные типы данных (АТД) Тема 6. Организация ввода/вывода и работа с файлами	
27		
28		
29		Отчет по лаб. работе
30	Тема 7. Жизненный цикл разработки программ Тема 8. Идея абстрактных структур данных. Реализации АТД при помощи классов С++	
31		
32		
33		Отчет по лаб. работе
34	Заключение	Защита КР / КП

### 6.4 Методика текущего контроля

#### на лекционных занятиях

Текущий контроль включает в себя:

– контроль посещаемости (не менее 80% занятий).

### **на лабораторных занятиях**

В процессе обучения по дисциплине студент обязан выполнить и успешно защитить по 4 лабораторных работы в 1 и 2 семестре. Под выполнением лабораторных работ подразумевается подготовка к работе, решение задачи, подготовка отчета и его защита. Отчет оформляется после решения задачи и представляется преподавателю на проверку в электронном виде. После проверки отчет либо возвращается (при наличии замечаний) на доработку, либо допускается к защите.

Лабораторные работы защищаются студентами индивидуально, в часы, отведенные для лабораторных работ. На защите лабораторной работы студент должен показать: понимание постановки задачи, подхода к ее решению, умение объяснять ход решения, выбор тех или иных методик решения задачи. Преподаватель задает вопросы, позволяющие определить глубину понимания теоретического материала, который лежит в основе решения задачи лабораторной работы, а также самостоятельность ее выполнения. Преподаватель может выдать для решения дополнительную задачу.

Текущий контроль включает в себя контроль выполнения лабораторной работы и сдачи в срок отчета по лабораторной работе.

Критерии оценивания: «не зачтено» - ставится, если основное содержание материала работы не раскрыто, не даны ответы на вопросы преподавателя, допущены грубые ошибки в определении понятий и в использовании терминологии, дополнительная задача не решена, либо решена с ошибками; «зачтено» ставится, если продемонстрировано усвоение основного содержания материала, работа выполнена полностью, самостоятельно и оформлена в соответствии с требованиями, решена дополнительная задача.

### **самостоятельной работы студентов**

Контроль самостоятельной работы студентов осуществляется на осуществляется на лекционных, лабораторных занятиях по методикам, описанным выше.

#### **при выполнении курсовой работы**

Текущий контроль и оценивание при выполнении курсовой работы осуществляется по методике, описанной в разделе 4.4.

## 7 Описание информационных технологий и материально-технической базы

Тип занятий	Тип помещения	Требования к помещению	Требования к программному обеспечению
Лекция	Лекционная аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, IBM-совместимый компьютер Pentium или выше проектор, экран, меловая или маркерная доска	1) Linux Альт Образование 10 и выше; 2) P7-Офис 7 и выше либо LibreOffice 7 и выше
Лабораторные работы	Лаборатория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя. Оснащено компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.	1) Linux Альт Образование 10 и выше; 2) P7-Офис 7 и выше либо LibreOffice 7 и выше
Самостоятельная работа	Помещение для самостоятельной работы	Оснащено компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.	1) Linux Альт Образование 10 и выше; 2) P7-Офис 7 и выше либо LibreOffice 7 и выше

## **8 Адаптация рабочей программы для лиц с ОВЗ**

Адаптированная программа разрабатывается при наличии заявления со стороны обучающегося (родителей, законных представителей) и медицинских показаний (рекомендациями психолого-медико-педагогической комиссии). Для инвалидов адаптированная образовательная программа разрабатывается в соответствии с индивидуальной программой реабилитации.

## ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

<b>№ п/п</b>	<b>Дата</b>	<b>Изменение</b>	<b>Дата и номер протокола заседания УМК</b>	<b>Автор</b>	<b>Начальник ОМОЛА</b>