

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Галунин Сергей Александрович  
Должность: проректор по учебной работе  
Дата подписания: 23.12.2025 12:07:09  
Уникальный программный ключ:  
08ef34338325bdb0ac5a47baa5472ce36cc3fc3b

Приложение к ОПОП  
«Информационно-управляющие  
системы»



**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

09.03.02 «Информационные системы и технологии»

по профилю

**«Информационно-управляющие системы»**

Санкт-Петербург

2025

## **ЛИСТ СОГЛАСОВАНИЯ**

Разработчики:

доцент, к.т.н. Синев В.Е.

Рабочая программа рассмотрена и одобрена на заседании кафедры ИС  
16.01.2025, 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	ИС
Общая трудоемкость (ЗЕТ)	11
Курс	1
Семестр	2, 1

## Виды занятий

Лекции (академ. часов)	68
Практические занятия (академ. часов)	85
Иная контактная работа (академ. часов)	6
Все контактные часы (академ. часов)	159
Самостоятельная работа, включая часы на контроль (академ. часов)	237
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»**

Целью дисциплины является формирование общих и профессиональных компетенций в области современного программирования, основанные на использовании методологии структурного программирования. Изучаются методы структурной декомпозиции типовых задач для разделения программы на части и ее реализации как совокупности функций. На основе использования языка программирования C++ формируются умения чтения, анализа, понимания и модификации программ, а также навыки алгоритмизации, кодирования, тестирования и отладки программ при решении типовых задач

### **SUBJECT SUMMARY**

### **«PROGRAMMING»**

The aim of the discipline is the formation of General and professional competencies in the field of modern programming, based on the use of the methodology of structural programming. We study the methods of structural decomposition of typical tasks to divide the program into parts and its implementation as a set of functions. Through the use of the programming language C++ to communicate the skills of reading, analysis, understanding and modification of programs and also skills of algorithms, coding, testing and debugging of programs in the solution of typical problems.

## **3 ОБЩИЕ ПОЛОЖЕНИЯ**

### **3.1 Цели и задачи дисциплины**

1. Целью изучения дисциплины является получение знаний, умений и навыков использования языка С++ для решения базовых задач программирования.
2. Задачи изучения дисциплины -получение знаний о методах структурной (функциональной) декомпозиции типовых задач для разделения программы на части и ее реализации как совокупности функций, формирование практических умений и навыков самостоятельно составлять, тестировать, отлаживать и документировать программы на языке С++.
3. Получение знаний о способах постановки и описания задач для решения на ЭВМ, основных методах и приемах разработки структурированных алгоритмов и программ, процедурных элементах языка С++.
4. Формирование умений самостоятельно составлять, тестировать, отлаживать и документировать программы на языке С++.
5. Получение навыков использования языка программирования в необходимом для решения типовых задач объеме, навыков чтения, анализа, понимания и модификации простых программ, включающих базовые конструкции программирования и стандартные типы данных, а также навыков алгоритмизации, кодирования, тестирования и отладки программ при программировании типовых задач.

### **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. «Архитектура информационных систем»
7. «Низкоуровневое программирование»
8. «Операционные системы»
9. «Теория информации, данные, знания»
10. «Тестирование программного обеспечения»
11. «Учебная практика (научно-исследовательская работа (получение первичных навыков научно-исследовательской работы))»
12. «Web-технологии»
13. «Анализ требований ч.1»
14. «Методы оптимизации»
15. «Мобильная разработка на Android ч.1»
16. «Основы бэкенд-разработки на Python ч.1»
17. «Основы фронтенд-разработки ч.1»
18. «Анализ требований ч.2»
19. «Мобильная разработка на Android ч.2»
20. «Производственная практика (технологическая (проектно-технологическая) практика)»
21. «Промышленное тестирование программного обеспечения»
22. «Компьютерная графика»

23. «Моделирование систем»
24. «Основы компьютерного зрения»
25. «Технологии баз данных»
26. «Администрирование информационных систем»
27. «Визуализация данных»
28. «Инженерный документооборот»
29. «Качество и метрология программного обеспечения»
30. «Системы реального времени»

### **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:

<b>Код компетенции/ индикатора компетенции</b>	<b>Наименование компетенции/индикатора компетенции</b>
ОПК-1	Способен применять естественнонаучные и общепрофессиональные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности;
ОПК-1.1	<i>Знает основы высшей математики, физики, основы вычислительной техники и программирования</i>
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий;
ОПК-6.1	<i>Знает методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий</i>
ОПК-6.2	<i>Умеет применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий</i>
ОПК-6.3	<i>Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач</i>

## **4 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ**

### **4.1 Содержание разделов дисциплины**

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

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>Лек, ач</b>	<b>Пр, ач</b>	<b>ИКР, ач</b>	<b>СР, ач</b>
1	Введение. Предмет дисциплины и ее задачи	1			
2	Этапы и проблемы решения задач с использованием компьютера	3			5
3	Общие сведения о языке программирования C++ и системе программирования	3			5
4	Стандартные (базовые) типы данных, операции и выражения	4	5		13
5	Основные управляющие структуры программирования и управляющие конструкции в языке C++	4	10		10
6	Приемы программирования циклов	5	10		25
7	Массивы	5	10		25
8	Указатели и ссылки	5	10	2	35
9	Функции и структура программы	7	10		34
10	Структуры, объединения, перечисления	7	10		30
11	Организация работы с файлами	7	10	2	30
12	Работа с динамической памятью	10	10	2	25
13	Введение в классы	6			
14	Заключение	1			
	Итого, ач	68	85	6	237
	Из них ач на контроль	0	0	0	70
	Общая трудоемкость освоения, ач/зе				396/11

#### **4.1.2 Содержание**

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>Содержание</b>
1	Введение. Предмет дисциплины и ее задачи	Предмет дисциплины и ее задачи. Содержание и форма проведения занятий. Связь с другими дисциплинами учебного плана направления (специальности).

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>Содержание</b>
2	Этапы и проблемы решения задач с использованием компьютера	Решение задач на компьютере. Уровни вычислительных систем: архитектура компьютера, операционные системы, методы и технологии программирования, прикладные технологии, информационные системы. Виды программного обеспечения (ПО): программа, программный комплекс, программный продукт (изделие), программная система. Основные показатели качества ПО. Жизненный цикл ПО: основные этапы и процессы, их соотношение с содержанием и видами занятий учебной дисциплины
3	Общие сведения о языке программирования C++ и системе программирования	Языки программирования и системы программирования. История создания языков С и С++. Краткая характеристика языка С++. Структура и основные элементы программы. Классификация действий и данных. Пример программы на языке С++. Система программирования. Трансляция программ и выполнение программы. Стандарты С и С++. Системы программирования С/С++ для различных вычислительных платформ и операционных сред.
4	Стандартные (базовые) типы данных, операции и выражения	Типы данных, переменные и константы. Понятие типов данных. Классификация простых предопределенных типов данных. Переменные, константы. Целочисленные типы данных. Вещественные типы данных. Логический тип данных. Символьный тип данных. Операции и выражения Преобразования типов данных. Операция присваивания. Арифметические операции. Операции отношения. Логические операции. По-разрядные (битовые) операции. Операции составного присваивания. Условная операция. Операция sizeof. Приоритеты рассмотренных операций. Ввод и вывод простых типов данных. Форматирование ввода / вывода.
5	Основные управляющие структуры программирования и управляющие конструкции в языке С++	Структурное программирование. Основные управляющие структуры программирования: последовательность, выбор (ветвление), итерации (циклы). Примеры преобразования структур. Семантика управляющих структур и инструкции языка С++. Операторы и блоки. Выбор вариантов: оператор if, расширение оператора if, множественный выбор. Итерации: цикл с предусловием, цикл с постусловием. Инструкции перехода.
6	Приемы программирования циклов	Итерация как базисная вычислительная схема (рекуррентные вычисления). Рекуррентные вычисления с целочисленными типами. Рекуррентные вычисления с вещественными типами. Программирование циклов в языке С++. Вложенные циклы. Циклы со сложным условием продолжения (выхода). Примеры.

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>Содержание</b>
7	Массивы	Понятие массива. Индексирование. Объявление массивов. Двумерные и многомерные массивы. Ввод-вывод массивов. Строки и тексты как массивы символов. Обработка текстовых строк. Массивы текстовых строк. Задачи разделения и слияния массивов, перестановка сегментов массива (циклический сдвиг) и т.п. Линейный и бинарный поиск в массиве. Простые алгоритмы сортировки (выбором, вставками, обменами). Работа с двумерными и многомерными массивами.
8	Указатели и ссылки	Понятие указателя. Работа с указателями. Арифметика указателей. Ошибки при работе с указателями. Указатели и массивы. Индексация указателей. Работа с массивами через указатели. Понятие ссылки. Использование ссылок.
9	Функции и структура программы	Процедурный подход к разработке программ. Определение функций в программе. Вызов функции (аргументы функции). Завершение работы функции и возврат значения. Передача параметров по значению, через указатели и по ссылке. Обращение к функциям в программе. Перегрузка функций. Параметры по умолчанию. Функции с переменным количеством аргументов. Использование функции как параметра другой функции. Рекурсия. Встраиваемые функции. Прототипы функций. Представление программы в виде набора функций. Многофайловая структура программы. Глобальные и локальные переменные. Область видимости и область жизни. Классы памяти. Автоматические переменные. Внешние переменные. Статические переменные. Внешние статические переменные. Регистровые переменные.
10	Структуры, объединения, перечисления	Понятие структур и их определение в программе. Доступ к полям структур. Указатели на структуры. Структурные параметры функций. Битовые поля структур. Объединения. Обычные объединения. Анонимные объединения. Перечисления. Примеры использования структур и объединений.
11	Организация работы с файлами	Потоки для работы с файлами. Создание потока, открытие и закрытие файлов. Запись и чтение данных в текстовых файлах. Запись и чтение данных в двоичном режиме. Обнаружение конца файла. Прямой доступ при работе с файлами. Статус потоков ввода-вывода. Функции управления потоками ввода-вывода. Примеры по работе с файлами.

<b>№ п/п</b>	<b>Наименование темы дисциплины</b>	<b>Содержание</b>
12	Работа с динамической памятью	Распределение памяти при работе программы. Динамическое выделение и освобождение памяти в стиле C++. Динамическое выделение и освобождение памяти в стиле С. Возможные ошибки при работе с динамической памятью. Динамические массивы. Способы создания одномерных динамических массивов. Многомерные динамические массивы. Способы создания. Примеры реализации. Динамические структуры. Одномерные однонаправленные и двунаправленные списки. Создание и удаление одномерных списков. Типовые операции со списками (доступ к элементам, удаление, перемещение, вставка элементов списка). Достоинства и недостатки. Многомерные списки. Стек. Возможные способы реализации и методы работы со стеками. Другие динамические структуры.
13	Введение в классы	Понятие класса. Инкапсуляция и сокрытие. Определение классов. Члены данных и функции-члены классов. Сопоставление структур и классов. Открытые и закрытые члены классов. Определение функций членов. Встроенные члены функции. Экземпляры классов. Обращение к членам экземпляров классов в программе. Массивы экземпляров классов. Конструкторы и деструкторы. Перегрузка конструкторов и членов функций. Дружественные функции. Присвоение экземпляров классов. Передача экземпляров классов в качестве параметров. Конструкторы копий. Перегрузка операторов. Примеры разработки классов
14	Заключение	Технологии программирования и дальнейшие перспективы изучения программирования

## 4.2 Перечень лабораторных работ

Лабораторные работы не предусмотрены.

## 4.3 Перечень практических занятий

<b>Наименование практических занятий</b>	<b>Количество ауд. часов</b>
1. Освоение работы со средой разработки программ (системой программирования) на языке С++. Работа с готовыми программами	2
2. Ввод – вывод простых типов данных. Форматирование ввода – вывода	2
3. Программирование ветвлений	2
4. Рекуррентные вычисления (схема итерации), операторы циклов	3
5. Рекуррентное вычисление суммы ряда, вещественные типы	3

<b>Наименование практических занятий</b>	<b>Количество ауд. часов</b>
6. Побитовая обработка данных	3
7. Программирование задач с одномерными массивами	4
8. Программирование задач с двумерными массивами	4
9. Функции. Механизмы передачи данных	4
10. Рекурсии	4
11. Функции-параметры. Функции с переменным количеством аргументов	4
12. Перегрузка и шаблоны функций	4
13. Средства разработки интерфейсов консольных приложений	4
14. Структуры и объединения. Массивы структур. Указатели на структуры	4
15. Структурные параметры функций	4
16. Текстовый режим работы с файлами	4
17. Бинарный режим работы с файлами	4
18. Одномерные и многомерные динамические массивы. Способы организации многомерных динамических массивов	4
19. Динамические списки. Операции со списками	2
20. Стеки. Способы реализации. Операции	2
21. Разработка класса для работы со списками	4
22. Конструкторы и деструкторы классов	2
23. Работа функций с экземплярами классов. Конструкторы копий	4
24. Перегрузка операций	4
25. Шаблоны классов	4
<b>Итого</b>	<b>85</b>

#### 4.4 Курсовое проектирование

Цель работы (проекта): Цель работы 1-го семестра: Законченное поэтапное решение содержательной задачи (постановка задачи, спецификация, выбор структур данных и разработка алгоритма, программная реализация, тестирование).

Цель работы 2-го семестра: Законченное поэтапное решение содержательной задачи (постановка задачи, спецификация, выбор структур данных и разработка алгоритма, программная реализация, тестирование) на основе использования структурированных типов данных (структур и классов)

Содержание работы (проекта): Содержание работы 1–го семестра:

Тема: "Решение содержательной задачи методами функционального програм-

мирования”

Курсовая работа 1–го семестра представляет собой набор из 8 отдельных задач, связанных с обработкой различных типов данных, каждая из которых реализуется в виде отдельной программы. Предлагаемые для решения задачи охватывают все темы изучаемой дисциплины. Типы заданий содержательно ориентированы на обработку различных типов данных, в том числе векторов и матриц, реализацию действий с полиномами (представляемыми массивами коэффициентов), с использованием всех управляющих конструкций языка программирования.

Содержание работы 2-го семестра:

Тема: ”Решение содержательной задачи методами процедурного программирования”

Курсовая работа 2-го семестра представляет собой разработку достаточно сложной интерактивной программы с диалоговым интерфейсом, ориентированной на использование практически всех возможностей языка программирования (динамических структур, работы с файлами, модулей). Тематика программ самая разнообразная – это программы для решения математических задач (решение линейных систем уравнений фиксированной и произвольной размерности, операции над многочленами фиксированного и произвольного порядка, алгебра матриц и т.п.), информационные системы с файловой системой хранения данных, игровые программы и т.д. Особое внимание уделяется надежности и дружественности программного интерфейса.

Оформление отчета по курсовой работе происходит в соответствии с шаблонами, принятыми в СПбГЭТУ ”ЛЭТИ”.

Количество страниц - от 10 до 50.

Количество использованных источников - от 3 до 7.

Отчет сдается преподавателю в электронном виде на почту.

Защита отчета происходит по методикам, приведенным в пункте 6.4.

В отчете должно быть отражено:

- тема
- постановка задачи
- спецификация предметной области
- выбор структур данных и разработка алгоритма
- программная реализация
- тестирование

Темы:

№ п/п	Название темы	Перевод темы
1	Решение линейных систем уравнений произвольного порядка	Solving linear systems of equations of arbitrary order

#### **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.

Особое место уделяется консультированию, как одной из форм обучения и контроля самостоятельной работы.

<b>Текущая СРС</b>	<b>Примерная трудоемкость, ач</b>
Работа с лекционным материалом, с учебной литературой	30
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	37
Подготовка к контрольным работам, коллоквиумам	0
Выполнение расчетно-графических работ	0
Выполнение курсового проекта или курсовой работы	100
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, дифференциированному зачету, экзамену	70
<b>ИТОГО СРС</b>	<b>237</b>

## **5 Учебно-методическое обеспечение дисциплины**

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

<b>№ п/п</b>	<b>Название, библиографическое описание</b>	<b>К-во экз. в библ.</b>
<b>Основная литература</b>		
1	Калмычков, Виталий Анатольевич. Представление и обработка математических данных на языке С++ [Текст] : учеб. пособие / В.А. Калмычков, Л.А. Чугунов, 2010. -62 с.	111
2	Дейтел Х.М. Как программировать на С++ [Текст] / Х.М. Дейтел, П.Дж. Дейтел ; пер. с англ. под ред. В.Г. Кузьменко, 2005. -1244 с.	100
<b>Дополнительная литература</b>		
1	Страуструп, Бьерн. Язык программирования С++ [Текст] : монография / Б. Страуструп; Пер. с англ. С.Анисимова, М.Кононова; Под ред. Ф.Андреева, А.Ушакова, 2001. -1098 с.	58
2	Саттер, Герб. Стандарты программирования на С++ [Текст] : 101 правило и рекомендация / Г. Саттер, А. Александреску, 2005. -223 с.	50
3	Керниган, Брайн В. Язык программирования Си [Текст] : монография / Б.В. Керниган; Пер. с англ. В.С.Штаркмана; Под ред. и с предисл. В.С.Штаркмана, 1992. -271 с.	45
4	Вирт, Никлаус. Алгоритмы + структуры данных = программы [Текст] : науч. изд. / Н. Вирт ; пер. с англ. Л.Ю. Иоффе ; под ред. Д.Б. Подшивалова, 1985. -406 с.	61

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

<b>№ п/п</b>	<b>Электронный адрес</b>
1	Брюс Эккель. Философия С++. Введение в стандартный С++: Питер, 2004http://er.jimr.ru/_downloads/ekkel.pdf
2	Бьерн Страуструп. Язык программирования С++http://lib.ru/CPPHB/cpptut.txt
3	Бьерн Страуструп. Программирование: Принципы и практика использования С++ (на англ.)https://www.stroustrup.com/programming.html
4	Основы программирования, Автор: В.В. Борисенкоhttps://intuit.ru/studies/courses/2193/67/info

### **5.3 Адрес сайта курса**

Адрес сайта курса: <https://vec.etu.ru/moodle/course/view.php?id=18900>

## **6 Критерии оценивания и оценочные материалы**

### **6.1 Критерии оценивания**

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

#### **Экзамен**

<b>Оценка</b>	<b>Описание</b>
Неудовлетворительно	Студент демонстрирует полное овладение курсом, способен применять полученные знания при решении конкретных задач
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**

К экзамену допускаются студенты, посетившие не менее 70% лекционных и практических занятий, выполнивших не менее 4-х и 3-х практических работ (в 1-м и 2-м семестрах соответственно), а также вовремя сдавших и защитивших курсовую работу, предусмотренную в соответствующем семестре. Экзамен проводится по билетам.

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

### **Вопросы к экзамену**

<b>№ п/п</b>	<b>Описание</b>
1	Уровни вычислительных систем
2	Виды программного обеспечения (ПО)
3	Жизненный цикл ПО
4	Основные показатели качества ПО
5	Структура и основные элементы программы
6	Типы данных, переменные и константы
7	Переменные, константы
8	Целочисленные типы данных. Вещественные типы данных. Логический тип данных. Символьный тип данных
9	Арифметические операции. Операции отношения. Логические операции. Поразрядные (битовые) операции. Операции составного присваивания
10	Ввод и вывод простых типов данных. Форматирование ввода / вывода.
11	Основные управляющие структуры программирования
12	Операторы и блоки
13	Выбор вариантов: оператор if, расширение оператора if, множественный выбор
14	Итерации: цикл с предусловием, цикл с постусловием
15	Рекуррентные вычисления с целочисленными типами
16	Рекуррентные вычисления с вещественными типами
17	Программирование циклов в языке C++. Вложенные циклы. Циклы со сложным условием продолжения (выхода). Примеры
18	Понятие массива. Индексирование. Объявление массивов
19	Двумерные и многомерные массивы
20	Ввод-вывод массивов
21	Задачи разделения и слияния массивов, перестановка сегментов массива
22	Линейный и бинарный поиск в массиве
23	Простые алгоритмы сортировки (выбором, вставками, обменами)
24	Понятие указателя. Работа с указателями

25	Указатели и массивы.
26	Индексация указателей. Работа с массивами через указатели.
27	Индексация указателей. Работа с массивами через указатели.
28	Определение функций в программе. Вызов функции (аргументы функции). Завершение работы функции и возврат значения
29	Передача параметров по значению, через указатели и по ссылке
30	Понятие структур и их определение в программе. Доступ к полям структур. Указатели на структуры
31	Потоки для работы с файлами. Создание потока, открытие и закрытие файлов. Запись и чтение данных в текстовых файлах.
32	Динамическое выделение и освобождение памяти в стиле C++
33	Понятие класса.
34	Конструкторы и деструкторы

## 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. Основные показатели качества ПО

2. Инdexация указателей. Работа с массивами через указатели.

УТВЕРЖДАЮ

Заведующий кафедрой

В.В.Цехановский

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**

<b>Неделя</b>	<b>Темы занятий</b>	<b>Вид контроля</b>
2	Общие сведения о языке программирования C++ и системе программирования	
3		Практическая работа
4	Стандартные (базовые) типы данных, операции и выражения	
5		Практическая работа
6	Основные управляющие структуры программирования и управляющие конструкции в языке C++	
7		Практическая работа
8	Приемы программирования циклов	
9		Практическая работа
10	Массивы	
11		Практическая работа
12	Указатели и ссылки	
13		Практическая работа
14	Стандартные (базовые) типы данных, операции и выражения	
15		
16	Основные управляющие структуры программирования и управляющие конструкции в языке C++	
17		Защита КР / КП
18	Приемы программирования циклов Массивы Указатели и ссылки	
19		
20	Функции и структура программы	
21		Практическая работа
22	Структуры, объединения, перечисления	
23		Практическая работа
24	Организация работы с файлами	
25		Практическая работа
26	Работа с динамической памятью	
27		Практическая работа
28	Функции и структура программы Структуры, объединения, перечисления Организация работы с файлами Работа с динамической памятью	
29		
30		
31		
32		
33		
34		Защита КР / КП

### **6.4 Методика текущего контроля**

#### **на лекционных занятиях**

Текущий контроль включает в себя контроль посещаемости (не менее **70** % занятий), по результатам которого студент получает допуск на экзамен.

## **на практических (семинарских) занятиях**

Текущий контроль включает в себя контроль посещаемости (не менее **80** % занятий), по результатам которого студент получает допуск на экзамен.

В ходе проведения практических занятий целесообразно привлечение студентов к как можно более активному участию в дискуссиях, решении задач, обсуждениях и т. д. При этом активность студентов также учитывается преподавателем, как один из способов текущего контроля на практических занятиях.

## **самостоятельной работы студентов**

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

## **при выполнении курсового проекта (работы)**

Текущий контроль при выполнении курсового проекта (работы) осуществляется в соответствии с методическими указаниями по курсовому проектированию и заданием на курсовой проект (работу).

Оформление пояснительной записи на курсовой проект (работу) выполняется в соответствии с требованиями к студенческим работам принятым в СПбГЭТУ.

Захист курсового проекта (работы) осуществляется в соответствии с требованиями «Положения о промежуточной аттестации».

## **Оценки за курсовую работу:**

”неудовлетворительно” - ставится при отсутствии работы, сданной в срок, либо при наличии плагиата чужого отчета

”удовлетворительно” - ставится при наличии грубых ошибок в подборе инструментов разработки, плохо написанном и неструктурированном коде, при минимальном количестве пояснений и описания работы.

”хорошо” - ставится при наличии несущественных ошибок, но при этом

видно, что студент понимает основные принципы разработки, которые преподаются в ходе курса

”отлично” - ставится при наличии полного хода рассуждений по этапам выполнения работ, при ”чистом” коде и практическом отсутствии ошибок, связанных с описанием предметной области и самой разработки в целом.

## **7 Описание информационных технологий и материально-технической базы**

<b>Тип занятий</b>	<b>Тип помещения</b>	<b>Требования к помещению</b>	<b>Требования к программному обеспечению</b>
Лекция	Лекционная аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, маркерная доска	
Практические занятия	Аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, ПК, проектор, экран, маркерная доска	1) Windows XP и выше; 2) Microsoft Visual Studio 10 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) Windows XP и выше; 2) Microsoft Office 2007 и выше 3) Microsoft Visual Studio 10 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>