

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Галунин Сергей Александрович  
Должность: проректор по учебной работе  
Дата подписания: 21.06.2023 10:26:52  
Уникальный программный ключ:  
08ef34338325bdb0ac5a47baa5472ce36cc3fc3b

Приложение к ОПОП  
«Управление и информатика в  
технических системах»



**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

27.03.04 «Управление в технических системах»

по профилю

**«Управление и информатика в технических системах»**

Санкт-Петербург

2022

## ЛИСТ СОГЛАСОВАНИЯ

Разработчики:

доцент, к.т.н. Власенко С.В.

Рабочая программа рассмотрена и одобрена на заседании кафедры АПУ  
18.01.2022, протокол № 1

Рабочая программа рассмотрена и одобрена учебно-методической комиссией  
ФКТИ, 24.02.2022, протокол № 2

Согласовано в ИС ИОТ

Начальник ОМОЛА Загороднюк О.В.

## 1 СТРУКТУРА ДИСЦИПЛИНЫ

Обеспечивающий факультет	ФКТИ
Обеспечивающая кафедра	АПУ
Общая трудоемкость (ЗЕТ)	11
Курс	1
Семестр	2, 1
<b>Виды занятий</b>	
Лекции (академ. часов)	68
Лабораторные занятия (академ. часов)	51
Практические занятия (академ. часов)	34
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
<b>Вид промежуточной аттестации</b>	
Экзамен (курс)	1
Курсовая работа (курс)	1
Экзамен (курс)	1
Курсовая работа (курс)	1

## **2 АННОТАЦИЯ ДИСЦИПЛИНЫ**

### **«ПРОГРАММИРОВАНИЕ»**

Цели дисциплины охватывают ряд тематических направлений, связанных с различными аспектами проектирования прикладного программного обеспечения – с общей методологией разработки программных продуктов, с основами проектирования программных алгоритмов и структур данных, а также с использованием универсальных языков программирования высокого уровня. В качестве базового выразительного средства в рамках настоящей дисциплины принят язык программирования C++, сочетающий в себе элементы алгоритмического и объектно-ориентированного языков программирования.

### **SUBJECT SUMMARY**

#### **«PROGRAMMING»**

The aims of discipline cover a number of thematic directions, connected with different aspects of the application software engineering – with the broad methodology of software development, with the basics of software design algorithms and data structures, as well as with the applying of universal programming languages of high level. Within this discipline the C++ programming language is embraced as a basic expressional means, combining elements of algorithmic and object-oriented programming languages.

## 3 ОБЩИЕ ПОЛОЖЕНИЯ

### 3.1 Цели и задачи дисциплины

1. Цели дисциплины -изучение различных аспектов проектирования прикладного программного обеспечения, приобретение практических умений и навыков практического применения полученных знаний на практике.

2. Задачи дисциплины: изучение и применение на практике основ алгоритмизации и прикладного программирования (с использованием языка C++) и знание методов построения программных алгоритмов и структур данных, используемых при решении с применением ЭВМ прикладных задач в различных проблемных областях; знание методологии разработки прикладного программного обеспечения, типовых способов организации программных данных и типовых подходов к построению программных алгоритмов.

3. Студенты должны приобрести:

-знание основ алгоритмизации и прикладного программирования (с использованием языка C++);

-знание методов построения программных алгоритмов и структур данных, используемых при решении с применением ЭВМ прикладных задач в различных проблемных областях;

-знание методологии разработки прикладного программного обеспечения, типовых способов организации программных данных и типовых подходов к построению программных алгоритмов.

4. Студенты должны выработать:

-умения согласованно решать задачи разработки эффективных моделей данных и алгоритмов их обработки при создании прикладного программного обеспечения;

-умение получать программные реализации полученных решений на универ-

сальном алгоритмическом языке высокого уровня.

5. Студенты должны получить:

-навыки решения на персональных ЭВМ простейших задач программной обработки данных;

-навыки использования инструментальных программных средств в процессе разработки программных продуктов.

### **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. «Организация ЭВМ и систем»

3. «Методы и средства объектно-ориентированного программирования»

4. «Технология программирования»

### 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>
ПК-6	Способен выполнять эксперименты на действующих объектах по заданным методикам и обрабатывать результаты с применением современных информационных технологий и технических средств
<i>ПК-6.2</i>	<i>Умеет обрабатывает результаты экспериментов на основе современных информационных технологий</i>
ПК-7	Способен проводить вычислительные эксперименты с использованием стандартных программных средств с целью получения математических моделей процессов и объектов автоматизации и управления
<i>ПК-7.1</i>	<i>Знает стандартные программные средства для проведения вычислительных экспериментов</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	Пр, ач	Лаб, ач	ИКР, ач	СР, ач
1	1 семестр Тема 1. Введение. Общие вопросы разработки программного обеспечения. Программирование и алгоритмизация	2	1	2		15
2	Тема 2. Общая характеристика языков С и С++	2	1	2		15
3	Тема 3. Структура С-программ	2	1	2	1	15
4	Тема 4. Простейшие программные описания	2	1	2		15
5	Тема 5. Основные управляющие конструкции языка и алгоритмические примитивы	5	3	5		15
6	Тема 6. Функции	2	1	2	1	15
7	Тема 7. Общие сведения о типизации данных в С/С++	2	1	2		15
8	Тема 8. Базовые структуры данных. Массивы. Строки	4	3	4		11
9	Тема 9. Структуры и объединения	5	2	5		10
10	Тема 10. Файлы и потоки	3	1	3	1	10
11	Тема 11. Динамические данные. Заключение	5	2	5		10
12	2 семестр Тема 1. Введение. Основные задачи курса	1	1	1		10
13	Тема 2. Основные методы и средства построения сложных динамических структур данных в программах на С/С++	3	1	1		10
14	Тема 3. Основные понятия в области списочных структур данных	3	2	2		10
15	Тема 4. Общие сведения о линейных списках	3	2	2	1	10
16	Тема 5. Основные разновидности линейных списков	7	3	3		11
17	Тема 6. Многосвязные списочные структуры	3	2	1		10
18	Тема 7. Древесные структуры данных	6	3	3	1	10
19	Тема 8. Сетевые структуры данных	3	1	2		10
20	Тема 9. Частные приложения динамических структур данных	5	2	2	1	10
	Итого, ач	68	34	51	6	237
	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	1 семестр Тема 1. Введение. Общие вопросы разработки программного обеспечения. Программирование и алгоритмизация	Предмет дисциплины и ее задачи. Содержание и форма проведения занятий. Связь с другими дисциплинами учебного плана. Основные этапы решения задач с использованием ЭВМ. Математическое, лингвистическое, информационное, алгоритмическое, программное, аппаратное обеспечение. Общая классификация программного обеспечения. Жизненный цикл программных продуктов. Этапы разработки программного обеспечения. Программы и алгоритмы. Понятие алгоритма. Методы формального описания алгоритмов. Блок-схемы алгоритмов. Основные подходы к разработке программных алгоритмов (частных целей, подъема, возврата, эвристический, рекурсивный, моделирование, метод ветвей и границ, backtracking). Основные характеристики программных алгоритмов. Взаимосвязь постановок проектных задач, алгоритмов их решения и используемых при этом структур данных.
2	Тема 2. Общая характеристика языков С и С++	Понятие языка программирования. Основные парадигмы программирования – процедурное, логическое, функциональное, объектно-ориентированное программирование. Основные классификационные признаки и характеристики языков программирования. Синтаксис и семантика языка. Система языка. Понятие алгоритмического языка программирования и наиболее распространенные представители универсальных алгоритмических языков высокого уровня. Система программирования и инструментальные средства поддержки основных этапов проектирования прикладных программных продуктов. Функциональное содержание процессов компиляции (трансляции, интерпретации) и построения загрузочных модулей, отладочных операций и тестирования. История создания С и С++. Место языка С в общей иерархии алгоритмических языков программирования. Отличительные особенности языка и системы программирования С/С++. Тенденции развития языка. Основные различия С и С++. Реализации языка для различных вычислительных платформ и операционных сред. Наиболее распр
3	Тема 3. Структура С-программ	Понятия программы, модуля, программной единицы. Общая структура программы. Функции как базовая единица структуризации С-программы. Пользовательские и библиотечные функции. Заголовочные файлы. Препроцессор С. Основные директивы препроцессора. Основные виды программных описаний в С/С++ и порядок их размещения в тексте программы.

№ п/п	Наименование темы дисциплины	Содержание
4	Тема 4. Простейшие программные описания	Алфавит языка. Ключевые слова и символы. Понятие оператора. Инструкции и декларации. Операторы, операции и выражения. Базовые типы данных и операции над ними. Приоритеты операций и образование выражений. Синтаксис описания констант и переменных. Операторы и управляющие конструкции. Простые и составные операторы. Операторные блоки и операторные скобки. Примеры простейших программ на C++.
5	Тема 5. Основные управляющие конструкции языка и алгоритмические примитивы	Основные управляющие конструкции – операторы циклов, условных и безусловных переходов, оператор выбора, операторы передачи управления continue, break, return. Простейшие средства консольного ввода-вывода. Эквивалентные представления операторов в схемах алгоритмов. Применение операторов при описании простейших типовых алгоритмических конструкций (обработка последовательностей, рекуррентные вычисления, различные реализации циклических процедур, вложенные циклические конструкции).
6	Тема 6. Функции	Декларация, реализация и вызов функции. Формальные и фактические параметры. Механизмы передачи параметров по значению и по адресу. Глобальные и локальные переменные. Область видимости и время жизни объектов. Классы памяти (auto, register, static, extern). Понятие рекурсии. Структуризация программ и вычислительных процессов на базе функций. Применение функций в простейших расчетных алгоритмах.
7	Тема 7. Общие сведения о типизации данных в C/C++	Единая концепция образования типов данных в C/C++. Комплекс базовых типов данных языка. Скалярные типы. Целочисленные, вещественные и символьные данные. Модификаторы типов. Предварительные сведения об указателях и ссылках. Физическая реализация базовых типов. Явные и неявные преобразования типов. Совместимость типов. Базовые и пользовательские типы данных. Порядок образования типов в C и C++. Базовые способы агрегирования данных (массивы, структуры, объединения, классы) и их использование при образовании типов. Образование типов и оператор typedef. Типы-функции. Перечисления enum.
8	Тема 8. Базовые структуры данных. Массивы. Строки	Понятие массива. Индексирование. Синтаксис описания массивов. Обращение к элементам массива. Инициализация массивов. Массивы и указатели. Арифметика указателей и индексирование. Двумерные и многомерные массивы. Строки и операции над ними. Основные библиотечные функции для работы со строчными данными. Ввод-вывод массивов и строк. Применение массивов в простейших алгоритмах обработки данных. Начальные сведения о задачах и алгоритмах поиска элементов в массиве и сортировки массивов.

№ п/п	Наименование темы дисциплины	Содержание
9	Тема 9. Структуры и объединения	Понятия структуры struct и объединения union. Декларация типов-структур и объединений. Структуры и объединения в общей схеме типобразования в С и С++. Механизмы доступа к полям. Указатели на структуры и объединения. Инициализация структур и объединений. Использование структур и объединений при конструировании неоднородных моделей данных.
10	Тема 10. Файлы и потоки	Описание и внутреннее представление файлов. Логическая и физическая организация файлов. Текстовые и бинарные файлы. Базовые операции над файлами. Режимы доступа. Позиционирование в файле. Буферизация. Библиотечные функции работы с файлами. Понятие потока. Стандартные потоки в С++. Функции для работы с потоками.
11	Тема 11. Динамические данные. Заключение	Распределение памяти под С-программу. Стек, «куча» (heap), размещение кода и статических данных. Управление ресурсами С-программы. Модели памяти. Статические и динамические данные. Указатели и операции над ними. Базовые операции над динамическими объектами. Механизмы выделения, перераспределения и освобождения динамической памяти. Функции, поддерживающие основные операции с динамической памятью. Операции new и delete. Указатели в составе агрегатов данных. Агрегированные динамические данные. Основные тенденции и направления развития методов, языков и средств поддержки программирования. Планируемые направления дальнейшего обучения программированию в рамках других учебных дисциплин.
12	2 семестр Тема 1. Введение. Основные задачи курса	Тематический профиль дисциплины и ее место в общей структуре учебных курсов. Содержание и форма проведения занятий. Предметные связи задач построения структур программных данных и алгоритмов их обработки. Общая характеристика сложных динамических структур данных и способов их использования при построении программных алгоритмов. Место задач разработки структур данных в едином процессе проектирования прикладного программного обеспечения.
13	Тема 2. Основные методы и средства построения сложных динамических структур данных в программах на С/С++	Единая концепция типизации данных в С/С++ и ее приложение к задачам формирования сложных динамических структур данных. Выразительные средства языка С/С++, на базе которых производится построение сложных статических и динамических структур данных. Указатели, массивы и структуры struct. Указатели как элементы структурированных типов данных. Рекурсивное определение типов данных. Базовые операции языка С/С++ для работы со сложно структурированными типами данных.

№ п/п	Наименование темы дисциплины	Содержание
14	Тема 3. Основные понятия в области списочных структур данных	Понятие списка. Элементы (узлы) списка и связи между ними. Основные классификационные признаки списков. Способы представления списочных структур данных. Основные классы задач обработки данных, решаемые на базе применения списочных структур. Общие элементы спецификации списочных структур на языке C/C++.
15	Тема 4. Общие сведения о линейных списках	Понятие линейного списка. Динамическая реализация линейных списков. Доступ к элементам списка и типовые процедуры обработки списков. Основные операции на линейных списках.
16	Тема 5. Основные разновидности линейных списков	Основные виды линейных списков. Однонаправленные (Л1), двунаправленные (Л2) и циклические списки. Стеки, очереди, деки: назначение, общая организация, базовые процедуры обработки. Типовые операции над основными видами списков: добавление и удаление элементов, вставка и исключение внутренних узлов, поиск данных. Примеры реализации типовых операций над списками в программах на C/C++.
17	Тема 6. Многосвязные списочные структуры	Структурные (топологические) и информационные классы связей в списках. Многосвязные списочные структуры. Мультисписки. Списки и индексные структуры. Примеры использования многосвязных списочных структур данных.
18	Тема 7. Древесные структуры данных	Понятие древесных структур данных. Классификационные признаки деревьев. Типовые классы задач, решаемых на базе применения древесных структур данных. Наиболее часто используемые разновидности деревьев. Деревья и задачи поиска и сортировки. Рекурсивные процедуры обработки деревьев. Бинарные деревья. Балансировка древесных структур данных. AVL-деревья. Представление многосвязных древесных структур. В-деревья. Совместное использование древесных и списочных структур данных. Обработка деревьев «по уровням». Примеры решения задач с древесными структурами данных.
19	Тема 8. Сетевые структуры данных	Понятие сетевой структуры данных. Разновидности сетей. Основные способы представления сетевых структур данных. Реализации сетей на базе динамических структур данных. Примеры построения и обработки сетевых моделей.
20	Тема 9. Частные приложения динамических структур данных	Использование сложных динамических структур в частных задачах обработки данных. Динамические реализации часто применяемых типовых абстрактных структур данных – индексных структур, ассоциативных массивов, множеств, хэш-таблиц. Обобщающие выводы по результатам изучения методов обработки сложных структур данных и их реализаций на языке C/C++.

## 4.2 Перечень лабораторных работ

Наименование лабораторной работы	Количество ауд. часов
1. Получение навыков работы со средой разработки программ на языке C++. Реализация примера простейшей программы	2
2. Программирование ветвлений (if-else) и циклические конструкции while и do-while	2
3. Циклические конструкции for. Простейшие рекуррентные вычисления	2
4. Задачи с вложенными циклическими конструкциями	4
5. Массивы и простейшие задачи их обработки (поиск, фильтрация, перестановки, сортировка)	4
6. Двумерные массивы и примеры задач их обработки	4
7. Структуры и объединения и их использование в программах	4
8. Формирование программных алгоритмов с использованием пользовательских функций.	4
9. Работа с файловыми данными.	4
10. Динамические данные и основные операции над ними	4
11. Базовые операции C/C++ при работе с динамическими структурами данных	2
12. Типовые операции над основными видами линейных списков. Работа с указателями в качестве межузловых связей	2
13. Типовые операции над основными видами линейных списков. Стек и очередь	2
14. Типовые операции над основными видами линейных списков. Дек и циклический список	2
15. Типовые операции над основными видами линейных списков для случая двунаправленных списков	2
16. Формирование мультисписков	2
17. Рекурсия при обработке динамических списочных и древесных структур данных	2
18. Работа с бинарными деревьями	3
Итого	51

## 4.3 Перечень практических занятий

Наименование практических занятий	Количество ауд. часов
1. Получение навыков работы со средой разработки программ на языке C++. Реализация примера простейшей программы	1
2. Программирование ветвлений (if-else) и простейших арифметических и логических вычислений	1
3. Циклические конструкции и простейшие рекуррентные вычисления	1
4. Задачи с вложенными циклическими конструкциями	2
5. Обработка данных на базе одномерных массивов (простейшие задачи поиска, фильтрации, перестановки, сортировки)	2

<b>Наименование практических занятий</b>	<b>Количество ауд. часов</b>
6. Обработка строчных данных и текстовых файлов.	2
7. Работа с двумерными массивами	2
8. Обработка неоднородных данных на базе использования структур и файлов	2
9. Формирование программных алгоритмов с использованием пользовательских функций	2
10. Работа с динамическими массивами	2
11. Обработка индексной структуры данных на базе динамического массива указателей на структуры	2
12. Обработка динамического массива структур	2
13. Построение и обработка линейного списка вида «стек»	2
14. Построение и обработка линейного списка вида «очередь»	2
15. Обработка последовательностей данных на базе линейных односвязных списков	2
16. Работа с двусвязным линейным списком	2
17. Работа с бинарным деревом поиска	2
18. Обработка бинарного дерева «по уровням»	3
Итого	34

#### **4.4 Курсовое проектирование**

Цель работы (проекта): 1 семестр: реализация прикладной разработки, включающей этапы постановки задачи, проектирования алгоритмов и структур данных, программирования, отладки и тестирования программы.

2 семестр: реализация прикладной разработки, включающей этапы постановки задачи, проектирования алгоритмов и структур данных, программирования, отладки и тестирования программы.

Содержание работы (проекта): 1 семестр:

Назначение проектируемой программы – поддержка интерактивной работы с динамическими числовыми матрицами и реализация заданного набора операций над ними.

Общая часть задания предусматривает реализацию иерархического пользовательского меню, средств поддержки консольного и файлового ввода/вывода матриц, средств управления динамической памятью, необходимой для хранения обрабатываемых матриц, средств редактирования заданных элементов матриц.

Персональные задания предусматривают реализацию частных операций (структурных, вычислительных, поисковых) над матрицами

2 семестр:

Общая тематика курсовой работы – формирование и обработка мультисписочной динамической структуры данных «Студенческий поток».

Общая часть задания:

Студент потока характеризуется следующими данными:

- ФИО (до 50 символов);
- номер группы;
- набор из пяти оценок за последнюю сессию (без указания предметов);
- размер стипендии.

Студенческий поток имеет заранее неопределенное количество групп.

Необходимо:

1) разработать и реализовать динамические структуры данных и алгоритмы их обработки, позволяющие поддерживать выполнение следующих функций:

- консольный ввод/вывод данных о студентах и группах потока;
- файловый ввод/вывод данных о студентах и группах потока;
- редактирование данных о студентах и группах потока, включающее операции добавления/удаления групп и студентов;

2) разработать и реализовать алгоритмы обработки структуры данных «Студенческий поток», предусмотренные персональным заданием.

Общие требования к программам:

1) Программа должна поддерживать систему меню, пункты которой соответствуют выполнению функций, предусмотренных общей и персональной частями задания.

2) Предлагаемые структуры данных должны учитывать изначальную неопределенность возможного количества групп и студентов в группах, а также обеспечивать максимальную скорость процессов обработки данных, предусмотренных заданием.

3) Тексты программ должны содержать комментарии, объясняющие назначение основных функций, типов и объектов данных, функциональных блоков и т.п.

4) Представляемые тексты программ должны обеспечивать возможность их компиляции и построения в среде MS Visual Studio.

Пояснительная записка к курсовой работе должна содержать следующие разделы:

1. Введение.
2. Постановка задачи.
3. Структура программы.
4. Описание модулей программы.
5. Листинг программы.
6. Приложения.

Требования к оформлению пояснительной записки:

1. Оформляется в виде документа MS Word, шрифт -Times New Roman, 12пт.
2. Объем пояснительной записки: 20-50 страниц.
3. В пояснительной записке должны быть ссылки на источники из списка используемых источников -минимальное число источников -2, максимальное -15.
4. Рисунки должны иметь подпись снизу в формате: "Рисунок №-название рисунка", выравнивание -посередине.
5. Заголовки таблиц указываются сверху в формате: "Таблица № -название таблицы", выравнивание -по правому краю.

Работа сдается преподавателю в печатном виде.

Темы:

№ п/п	Название темы	Перевод темы
1	Разработка программы «Интерактивная работа с динамическими числовыми матрицами	Interactive work with dynamic numeric matrixes
2	Динамические списки и их обработка	Dynamic lists and their processing



#### **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 деятельности, которые указывают путь решения проблемы.

<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)	10
Самостоятельное изучение разделов дисциплины	10
Выполнение домашних заданий, домашних контрольных работ	0
Подготовка к лабораторным работам, к практическим и семинарским занятиям	20
Подготовка к контрольным работам, коллоквиумам	0
Выполнение расчетно-графических работ	0
Выполнение курсового проекта или курсовой работы	97
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>237</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 библи.
Основная литература		
1	Калмычков, Виталий Анатольевич. Представление и обработка математических данных на языке С++ [Текст] : учеб. пособие / В.А. Калмычков, Л.А.Чугунов, 2010. -62 с.	неогр.
2	Ивановский, Сергей Алексеевич. Разработка корректных программ [Текст] : Практикум по программированию / С.А.Ивановский, В.А.Калмычков, А.А.Лисс, 2001. -71 с.	неогр.
3	Ивановский, Сергей Алексеевич. Разработка корректных программ [Текст] : Учеб. пособие / С.А.Ивановский, 2003. -99 с.	неогр.
4	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
5	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
6	Эккель, Брюс. Философия С++. Введение в стандартный С++ [Текст] : монография / Б. Эккель; [Пер. с англ. Е. Матвеева], 2004. -571 с	60
7	Эккель, Брюс. Философия С++ [Текст] : практ. программирование / Б. Эккель, Ч. Эллисон, 2004. -608 с	70
8	Саттер, Герб. Стандарты программирования на С++ [Текст] : 101 правило и рекомендация / Г. Саттер, А. Александреску, 2005. -223 с.	50
Дополнительная литература		
1	Павловская, Татьяна Александровна. С/С++. Программирование на языке высокого уровня [Текст] : Учеб. для вузов по направлению "Информатика и вычисл. техника" / Т.А. Павловская, 2003. -460 с., табл	155
2	Программирование на языках С и С++ в среде Microsoft Visual С++ [Текст] : метод. указания к лаб. работам по дисциплинам "Информатика", "Программирование" / Санкт-Петербургский государственный электротехнический университет им. В.И. Ульянова (Ленина) "ЛЭТИ", 2006. -31 с.	138
3	Объектно-ориентированное программирование [Текст] : метод. указания к лаб. работам / Санкт-Петербургский государственный электротехнический университет им. В.И. Ульянова (Ленина) "ЛЭТИ", 2007. -32 с.	128
4	Ветчинкин, Александр Сергеевич. Программирование и основы алгоритмизации [Текст] : учеб. пособие / А.С. Ветчинкин, А.Л. Стариченков, 2006. -88 с.	125
5	Водовозов, Валерий Михайлович. Объектно-ориентированное программирование на С++ [Текст] : учеб. пособие / В.М. Водовозов, Ф.В. Чмиленко, 2007. -127 с	118

№ п/п	Название, библиографическое описание	К-во экз. в библ.
6	Курсовое программирование на С++ [Текст] : метод. указания к курсовой работе / Санкт-Петербургский государственный электротехнический университет им. В.И. Ульянова (Ленина) ”ЛЭТИ”, 2006. -31 с.	108
7	Павловская, Татьяна Александровна. С/С++. Структурное программирование [Текст] : практикум / Т.А. Павловская , Ю.А. Щупак, 2004. -238 с.	96

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

№ п/п	Электронный адрес
1	Бьерн Страуструп. Язык программирования С++ <a href="http://lib.ru/CPPHB/cpptut.txt">http://lib.ru/CPPHB/cpptut.txt</a>
2	Онлайн справочник программиста на С и С++ <a href="http://www.c-cpp.ru/">http://www.c-cpp.ru/</a>
3	С++ reference <a href="https://en.cppreference.com/w/">https://en.cppreference.com/w/</a>
4	Learn C and C++ Programming <a href="https://www.cprogramming.com/">https://www.cprogramming.com/</a>

## 5.3 Адрес сайта курса

Адрес сайта курса: <https://vec.etu.ru/moodle/course/view.php?id=7445>

## 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	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 теорем
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

Для допуска к экзамену необходимо выполнить и защитить все практические и лабораторные работы, курсовую работу. Обязательно посещение не менее 80% лекций. Экзамен проводится по билетам. В каждом билете два теоретических вопроса. Время на подготовку ограничено.

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

### Вопросы к экзамену

№ п/п	Описание
1	Основные этапы решения задач с использованием ЭВМ.
2	Отличительные особенности языка и системы программирования C/C++.
3	Основные виды программных описаний в C/C++ и порядок их размещения в тексте программы.
4	Операторы и управляющие конструкции. Простые и составные операторы. Операторные блоки и операторные скобки.
5	Основные управляющие конструкции – операторы циклов, условных и безусловных переходов, оператор выбора, операторы передачи управления continue, break, return.
6	Декларация, реализация и вызов функции. Формальные и фактические параметры. Механизмы передачи параметров по значению и по адресу.
7	Классы памяти (auto, register, static, extern).
8	Единая концепция образования типов данных в C/C++. Комплекс базовых типов данных языка. Скалярные типы. Целочисленные, вещественные и символьные данные. Модификаторы типов.
9	Понятие массива. Индексирование. Синтаксис описания массивов. Обращение к элементам массива. Инициализация массивов. Массивы и указатели.
10	Понятия структуры struct и объединения union. Декларация типов-структур и объединений.
11	Стандартные потоки в C++. Функции для работы с потоками.
12	Решение задач с древесными структурами данных.
13	Организация работы с файлами в языках C и C++.
14	Статические переменные
15	Строки как структура/класс на основе массива символов, использование динамической памяти.
16	Определение класса, использование класса, вызов методов класса
17	Библиотека обработка символьных строк string, функции для работы с данными типа string.
18	Простой вид списков – однонаправленный линейный, использование указателей.
19	Верификация программ.

**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. Единая концепция типизации данных в C/C++ и ее приложение к задачам формирования сложных динамических структур данных.
2. Примеры решения задач с древесными структурами данных.

УТВЕРЖДАЮ

Заведующий кафедрой

М.Ю. Шестопалов

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 2. Общая характеристика языков С и С++ Тема 3. Структура С-программ Тема 4. Простейшие программные описания	
2		
3		
4		
5		Коллоквиум
6	Тема 5. Основные управляющие конструкции языка и алгоритмические примитивы Тема 6. Функции Тема 7. Общие сведения о типизации данных в С/С++	
7		
8		
9		
10		Коллоквиум
11	Тема 8. Базовые структуры данных. Массивы. Строки Тема 9. Структуры и объединения Тема 10. Файлы и потоки	
12		
13		
14		
15		
16		
17		Защита КР / КП
18	Тема 2. Основные методы и средства построения сложных динамических структур данных в программах на С/С++ Тема 3. Основные понятия в области списочных структур данных Тема 4. Общие сведения о линейных списках	
19		
20		
21		
22		
23	Коллоквиум	
24	Тема 5. Основные разновидности линейных списков Тема 6. Многосвязные списочные структуры Тема 7. Древесные структуры данных	
25		
26		
27		
28		
29		Коллоквиум
30	Тема 6. Многосвязные списочные структуры	
31		
32		
33		
34		Защита КР / КП

### 6.4 Методика текущего контроля

#### на лекционных занятиях

Текущий контроль включает в себя контроль посещаемости (не менее **80**



% занятий), по результатам которого студент получает допуск на экзамен.

### **на лабораторных занятиях**

- Порядок выполнения лабораторных работ, подготовки отчетов и их защиты

В процессе обучения по дисциплине «Программирование» студент обязан выполнить 18 лабораторных работ. Под выполнением лабораторных работ подразумевается подготовка к работе, ее проведение, подготовка отчета и его защита. После каждой лабораторной работы предусматривается ее защита. Выполнение лабораторных работ студентами осуществляется индивидуально (*в бригадах до 2 - 3 человек*). Оформление отчета студентами осуществляется индивидуально (*в количестве одного отчета на бригаду*) в соответствии с принятыми в СПбГЭТУ правилами оформления студенческих работ. Отчет оформляется после ее выполнения и представляется преподавателю на проверку. После проверки отчет либо возвращается (при наличии замечаний) на доработку, либо подписывается к защите.

Лабораторные работы защищаются студентами индивидуально. Каждый студент получает вопрос по теоретической части, или по программе, после чего ему предоставляется время для подготовки ответа. При обсуждении ответа преподаватель может задать несколько уточняющих вопросов. В случае если студент демонстрирует достаточное знание вопроса, работа считается защищенной.

На защите лабораторной работы студент должен показать: понимание алгоритмов программирования и знание особенностей их применения, понимание и умение объяснять особенности применяемых методов, возможные области их применения и т.д., умение давать качественную и количественную оценку полученных результатов, демонстрировать навыки и умения, приобретенные при выполнении лабораторной работы.

Текущий контроль включает в себя выполнение, сдачу в срок отчетов и их защиту по всем лабораторным работам, по результатам которой 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 семинарских и практических занятий целесообразно привлечение студентов к как можно более активному участию в дискуссиях, решении задач, обсуждениях и т. д. При этом активность студентов также может учитываться преподавателем, как один из способов текущего контроля на практических занятиях.

#### **самостоятельной работы студентов**

Контроль самостоятельной работы студентов осуществляется на лекционных, лабораторных 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 записки на курсовой проект (работу) выполняется в соответствии с требованиями к студенческим работам принятым в СПбГЭТУ.

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

Курсовая работа оценивается по пятибальной шкале:

Оценка "отлично" выставляется за курсовую работу, которая содержит

грамотно изложенный материал, с соответствующими выводами и обоснованными предложениями.

Оценка "хорошо" выставляется за грамотно выполненную во всех отношениях курсовую работу при наличии небольших недочетов в ее содержании или оформлении.

Оценка "удовлетворительно" выставляется за курсовую работу, которая удовлетворяет всем предъявляемым требованиям, в ней просматривается непоследовательность изложения материала, представлены необоснованные выводы и предложения.

Оценка "неудовлетворительно" выставляется за курсовую работу, которая не содержит анализа и практического исследования деятельности объекта, выводы и предложения носят декларативный характер.

## 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, маркерная доска	
Лабораторные работы	Лаборатория	Количество посадочных мест с ПК – в соответствии с контингентом, рабочее место преподавателя, доска	1) MS Windows 7 и выше; 2) MS Visual Studio 2013 и выше
Практические занятия	Аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, маркерная доска, ноутбук	1) MS Windows 7 и выше; 2) MS Visual Studio 2013 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

## **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>