

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Галунин Сергей Александрович  
Должность: проректор по учебной работе  
Дата подписания: 03.09.2022 14:46:27  
Уникальный программный ключ:  
08ef34338325bdb0ac5a47baa5472ce36cc3fc3b



**СПбГЭТУ «ЛЭТИ»**  
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

Приложение к ОПОП  
«Возобновляемая энергетика»

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 И ОСНОВЫ АЛГОРИТМИЗАЦИИ»**

для подготовки бакалавров

по направлению

13.03.02 «Электроэнергетика и электротехника»

по профилю

**«Возобновляемая энергетика»**

Санкт-Петербург

2022

## ЛИСТ СОГЛАСОВАНИЯ

Разработчики:

доцент, к.т.н. Шевченко А.В.

Рабочая программа рассмотрена и одобрена на заседании кафедры РАПС  
19.09.2020, протокол № 2

Рабочая программа рассмотрена и одобрена учебно-методической комиссией  
ФЭА, 20.09.2020, протокол № 2

Согласовано в ИС ИОТ

Начальник ОМОЛА Загороднюк О.В.

## 1 СТРУКТУРА ДИСЦИПЛИНЫ

Обеспечивающий факультет	ФЭА
Обеспечивающая кафедра	РАПС
Общая трудоемкость (ЗЕТ)	4
Курс	2
Семестр	3
<b>Виды занятий</b>	
Лекции (академ. часов)	34
Лабораторные занятия (академ. часов)	17
Практические занятия (академ. часов)	34
Иная контактная работа (академ. часов)	3
Все контактные часы (академ. часов)	88
Самостоятельная работа, включая часы на контроль (академ. часов)	56
Всего (академ. часов)	144
<b>Вид промежуточной аттестации</b>	
Дифф. зачет (курс)	2
Курсовая работа (курс)	2

## **2 АННОТАЦИЯ ДИСЦИПЛИНЫ**

### **«ПРОГРАММИРОВАНИЕ И ОСНОВЫ АЛГОРИТМИЗАЦИИ»**

В рамках дисциплины студенты продолжают знакомиться с технологиями программирования – со структурным подходом к программированию, с объектно-ориентированным программированием, методами проектирования пользовательских интерфейсов.

Вопросы разработки программ раскрываются на примере языка программирования высокого уровня C++. Рассматриваются вопросы инкапсуляции данных и методов их обработки, наследования, полиморфизма.

Изучаются типовые алгоритмы сортировки массивов, поиска в массивах, строках и файлах, работы с динамическими структурами данных, решения задач комбинаторной оптимизации.

На лабораторных занятиях осваиваются навыки создания программ в среде Borland C++ Builder. В ходе курсовой работы студентами делается индивидуальный проект интерактивного приложения, включающего развитый интерфейс пользователя, объектно-ориентированные технологии программирования и алгоритмы обработки данных.

### **SUBJECT SUMMARY**

#### **«PROGRAMMING AND ALGORITHMIC FOUNDATIONS»**

Within the discipline of students continue to get acquainted with the technology of programming -a structural approach to programming, object-oriented programming, design techniques of user interfaces.

Programming issues are revealed by the example of high-level language C++. There are the questions of encapsulating data and methods of their processing, inheritance, polymorphisme.

The typical algorithms for sorting arrays, searching into arrays, strings and files, work

with dynamic data structures, problem solving combinatoric optimisation.

On laboratory and practical classes learn skills building programs in the environment Borland C++ Builder. During the course work students do an individual project of an interactive application, including development of user interface, object-oriented programming techniques and data processing algorithms.

## **3 ОБЩИЕ ПОЛОЖЕНИЯ**

### **3.1 Цели и задачи дисциплины**

1. Получение знаний в области информационных технологий, в том числе в области создания программного обеспечения для современной компьютерной техники.
2. Получение навыков программирования на языке высокого уровня C++, знаний технологий проектирования, разработки и отладки программ, понимания и применения типовых алгоритмов.
3. Знание технологии структурного и объектно-ориентированного программирования, принципов разработки пользовательского интерфейса, типовых алгоритмы сортировки, поиска и оптимизации.
4. Умение разрабатывать структуры данных, выбирать подходящие алгоритмы, разрабатывать программы на языке высокого уровня C++.
5. Владеть навыками создания интерактивных приложений в среде Borland C++ Builder.

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

Дисциплина изучается на основе ранее освоенных дисциплин учебного плана:

1. «Информатика»
2. «Информационные технологии»

и обеспечивает изучение последующих дисциплин:

1. «Микропроцессорные средства управления»
2. «Робототехника»

### 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>
УК-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
<i>УК-1.1</i>	<i>Выполняет поиск необходимой информации, её критический анализ и обобщает результаты анализа для решения поставленной задачи</i>
<i>УК-1.2</i>	<i>Применяет методы системного подхода и критического анализа проблемных ситуаций; разрабатывает стратегию действий, принимает конкретные решения для ее реализации</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	Структурное программирование	2	2		1	4
2	Проектирование пользовательского интерфейса	4	4	7	1	8
3	Объектно-ориентированное программирование	8	8	10	1	16
4	Алгоритмы и структуры данных	20	20			28
	Итого, ач	34	34	17	3	56
	Из них ач на контроль	0	0	0	0	0
	Общая трудоемкость освоения, ач/зе	144/4				

#### 4.1.2 Содержание

№ п/п	Наименование темы дисциплины	Содержание
1	Структурное программирование	<p>Предпосылки структурного программирования. Проблемы неструктурированных программ. Структурный подход к программированию.</p> <p>Нисходящая разработка. Модульность. Заглушки. Свойства модуля. Модульность в среде Borland C++ Builder.</p> <p>Структурное программирование. Базовые конструкции. Вложенность конструкций. Принцип модульности. Псевдокод. Пошаговая детализация. Использование псевдокода. Реализация программы на C++.</p> <p>Принципы сквозного структурного контроля. Верификация программного обеспечения. Тестирование. Проверка кода (peer review).</p>



№ п/п	Наименование темы дисциплины	Содержание
2	Проектирование пользовательского интерфейса	<p>Роль пользовательского интерфейса. Эволюция пользовательских интерфейсов. Способы организации человеко-машинного диалога. Современные пользовательские интерфейсы. Уровни пользовательского интерфейса. Внешний уровень интерфейса – метафора. Внешний уровень интерфейса – стиль.</p> <p>Концептуальный уровень пользовательского интерфейса. Элементы для представления данных. Декоративные элементы. Управляющие элементы. Композиционные элементы.</p> <p>Геометрия элементов пользовательского интерфейса. Реакция элементов интерфейса на изменение размера. Выравнивание элементов. Использование выравнивания при компоновке экранов.</p> <p>Свойства элементов интерфейса. Видимость. Чувствительность. Реализация событийного принципа. Сценарии.</p>
3	Объектно-ориентированное программирование	<p>Принципы объектно-ориентированного программирования. Языки объектно-ориентированного программирования. Сравнение C и C++. Эквивалентные понятия C и C++.</p> <p>Инкапсуляция. Понятие класса в C++. Доступ к элементам класса. Определение методов класса. Конструкторы и деструкторы, их свойства. Перегрузка конструктора. Конструктор копирования. Создание и разрушение объектов класса. Статические данные класса. Статические методы класса. Указатель на текущий объект класса.</p> <p>Абстракция. Классификация объектов. Иерархия абстрактного и конкретного. Понятие наследования. Простое и множественное наследования. Критерии «хорошей» иерархии классов.</p> <p>Наследование в C++. Ограничение доступа при наследовании. Наследование и конструкторы. Наследование и деструкторы.</p> <p>Полиморфизм. Переопределение методов в C++. Раннее связывание. Позднее связывание. Чисто виртуальные методы. Абстрактные классы. Конфликты при множественном наследовании. Виртуальные базовые классы.</p> <p>Перегрузка функций в C++. Перегрузка методов в</p>

№ п/п	Наименование темы дисциплины	Содержание
4	Алгоритмы и структуры данных	<p>Понятие алгоритма. Свойства алгоритма. Эффективность алгоритмов. Трансвычислительные задачи. Представления алгоритмов. Блок-схемы алгоритмов. Элементы блок-схем. Классы алгоритмов.</p> <p>Сортировка массивов. Прямые алгоритмы сортировки. Сортировка прямым включением. Сортировка прямым выбором. Сортировка прямым обменом. Шейкерная сортировка. Сортировка массивов. Улучшенные методы сортировки. Улучшенный метод сортировки Шелла. Метод быстрой сортировки Хоара.</p> <p>Алгоритмы поиска в массивах. Линейный поиск. Двоичный поиск. Интерполяционный поиск.</p> <p>Алгоритмы поиска в строках. Прямой поиск в строке. Алгоритм Кнута, Морриса и Пратта. Алгоритм Боуера и Мура.</p> <p>Поиск в файлах базы данных. Индексно-последовательный метод доступа (ISAM) . Доступ по вычисляемому ключу (HASH).</p> <p>Динамические структуры данных. Линейные списки. Обход списка. Включение в список. Исключение из списка. Очереди. Стеки. Деревья. Обход дерева. Графы. Обход графа.</p> <p>Задачи на связность. Алгоритм Уоршалла.</p>

## 4.2 Перечень лабораторных работ

Наименование лабораторной работы	Количество ауд. часов
1. Классы, инкапсуляция	4
2. Наследование, полиморфизм	4
3. Алгоритмы сортировки	2
4. Алгоритмы поиска	2
5. Задачи комбинаторной оптимизации	5
Итого	17

## 4.3 Перечень практических занятий

Наименование практических занятий	Количество ауд. часов
1. Структурное программирование	2
2. Нисходящее проектирование	4
3. Пошаговая детализация	4
4. Проектирование пользовательского интерфейса	8
5. Разработка программного кода	12
6. Отладка программного кода	4
Итого	34

#### **4.4 Курсовое проектирование**

Цель работы (проекта): Овладение навыками создания программного обеспечения в среде Embarcadero C++ Builder..

Содержание работы (проекта): Разработка и отладка приложения на языке С в среде Embarcadero C++ Builder..

Примерные темы:

<b>№ п/п</b>	<b>Название темы</b>	<b>Перевод темы</b>
1	Анализ данных радиопередач	
2	Учет товаров на складе	
3	Картотека чертежей	
4	План испытаний станков	

#### **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 ин-

формационными 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	16
Опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	4
Самостоятельное изучение разделов дисциплины	6
Выполнение домашних заданий, домашних контрольных работ	4
Подготовка к лабораторным работам, к практическим и семинарским занятиям	0
Подготовка к контрольным работам, коллоквиумам	0
Выполнение расчетно-графических работ	0
Выполнение курсового проекта или курсовой работы	20
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	6
<b>ИТОГО СРС</b>	<b>56</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	Шевченко, Алексей Владимирович. Программирование и основы алгоритмизации [Текст] : учеб. пособие / А. В. Шевченко, 2018. -143 с.	100
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	Вирт, Никлаус. Алгоритмы и структуры данных [Текст] : монография / Н. Вирт; Пер. с англ. Д.Б.Подшивалова, 1989. -360 с.	191

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

№ п/п	Электронный адрес
1	Сайт компании Embarcadero <a href="http://www.embarcadero.com">http://www.embarcadero.com</a>

## 6 Критерии оценивания и оценочные материалы

### 6.1 Критерии оценивания

Для дисциплины «Программирование и основы алгоритмизации» формой промежуточной аттестации является дифф. зачет. Оценивание качества освоения дисциплины производится с использованием рейтинговой системы.

#### Дифференцированный зачет

Оценка	Количество баллов	Описание
Неудовлетворительно	0 – 51	теоретическое содержание курса не освоено, необходимые практически навыки и умения не сформированы, выполненные учебные задания содержат грубые ошибки, дополнительная самостоятельная работа над курсом не приведет к существенному повышению качества выполнения учебных заданий
Удовлетворительно	52 – 67	теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки и умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки
Хорошо	68 – 84	теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки и умения сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые виды заданий выполнены с ошибками
Отлично	85 – 100	теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки и умения сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено количеством баллов, близким к максимальному

## Особенности допуска

Методические материалы; описание шкал оценивания; типовые контрольные задания или иные материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении к рабочей программе дисциплины. Конкретные формы и процедуры текущего контроля знаний и промежуточной аттестации, а также методические указания для обучающихся по самостоятельной работе при освоении дисциплины доводятся до сведения обучающихся в течение первых недель обучения.

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

### Примерные вопросы к дифф.зачету

№ п/п	Описание
1	Предпосылки структурного программирования. Проблемы неструктурированных программ. Структурный подход к программированию.
2	Нисходящая разработка. Модульность. Заглушки.
3	Свойства модуля. Модульность в среде Borland C++ Builder.
4	Структурное программирование. Базовые конструкции. Вложенность конструкций.
5	Принцип модульности. Псевдокод. Пошаговая детализация.
6	Использование псевдокода. Реализация программы на C++.
7	Принципы сквозного структурного контроля. Верификация программного обеспечения. Тестирование. Проверка кода (peer review).
8	Роль пользовательского интерфейса. Эволюция пользовательских интерфейсов. Способы организации человеко-машинного диалога.
9	Современные пользовательские интерфейсы. Уровни пользовательского интерфейса. Внешний уровень интерфейса – метафора. Внешний уровень интерфейса – стиль.
10	Концептуальный уровень пользовательского интерфейса. Элементы для представления данных. Декоративные элементы. Управляющие элементы. Композиционные элементы.

### 6.3 График текущего контроля успеваемости

Неделя	Темы занятий	Вид контроля
6	Структурное программирование	Отчет по лаб. работе
11	Проектирование пользовательского интерфейса	Отчет по лаб. работе
15	Объектно-ориентированное программирование	Отчет по лаб. работе

### 6.4 Методика текущего контроля

В процессе обучения по дисциплине «**Программирование и основы алгоритмизации**» студент обязан выполнить все лабораторные работы, предусмотренные РП дисциплины. Под выполнением лабораторных работ подразумевается подготовка к работе, проведение экспериментальных исследований, подготовка отчета и его защита на коллоквиуме. После каждой 2-ой лабораторной работы предусматривается проведение коллоквиумов, на которых осуществляется защита лабораторных работ. Выполнение лабораторных работ студентами осуществляется индивидуально или при большом количестве студентов в группе – бригадами из двух человек. Оформление отчета студентами осуществляется индивидуально в соответствии с принятыми в СПбГЭТУ правилами оформления студенческих работ. Отчет оформляется после выполнения экспериментальных исследований 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 процедуре проведения экспериментальных исследований, или по последующей обработке результатов, 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 ме-



тодики исследования и знание особенностей её применения, понимание и умение объяснять особенности применяемых методов, возможные области их применения и т. д., умение давать качественную и количественную оценку полученных экспериментальных результатов и прогнозировать реакции исследуемого объекта на различные воздействия, навыки и умения, приобретенные при выполнении лабораторной работы.

## 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, экран, проектор	
Лабораторные работы	Лаборатория	Количество посадочных мест – в соответствии с контингентом,,	
Практические занятия	Аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, экран, проектор	
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>