

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

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

13.03.02 «Электроэнергетика и электротехника»

по профилю

«Возобновляемая энергетика»

Санкт-Петербург

2022

ЛИСТ СОГЛАСОВАНИЯ

Разработчики:

заведующий кафедрой, д.т.н., доцент Белов М.П.

Рабочая программа рассмотрена и одобрена на заседании кафедры РАПС
26.05.2021, протокол № 5

Рабочая программа рассмотрена и одобрена учебно-методической комиссией
ФЭА, 16.06.2021, протокол № 6

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
Виды занятий	
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
Вид промежуточной аттестации	
Дифф. зачет (курс)	2
Курсовая работа (курс)	2

2 АННОТАЦИЯ ДИСЦИПЛИНЫ

«ПРОГРАММИРОВАНИЕ»

Дисциплина обеспечивает изучение и освоение базовых понятий, методов и приемов программирования на языке программирования C++. При освоении дисциплины изучаются следующие основные положения: стандартные типы данных и набор операций для них, организация ввода/вывода и файлы, основные управляющие конструкции (ветвление, циклы) и их реализация на C++, массивы и указатели, функции, структуры и классы, представление и обработка линейных списков, понятие символьной строки и текста на основе массива и списка, модульная структура программы.

SUBJECT SUMMARY

«PROGRAMMING»

The discipline ensures study and mastery of fundamental concepts, methods and approaches of programming in the programming language C++. With the mastery of discipline the following basic statements are studied: the standard data types and the operations set for them, the input/output organization and files, basic control construction (branching, cycles) and their realization in C++, arrays and pointers, functions, structures and classes, idea and processing of linear lists, the concept of symbolic line and text on the basis of array and list, the modular structure of program.

3 ОБЩИЕ ПОЛОЖЕНИЯ

3.1 Цели и задачи дисциплины

1. Цель дисциплины – получение теоретических знаний основ программирования и базовых конструкций языка программирования C++ и формирование практических навыков их использования в профессиональной деятельности.

2. Задачи дисциплины:

Изучение теоретических и практических основ программирования и базовых конструкций языка программирования C++.

Освоение основных синтаксических конструкций языка программирования C++.

Получение начальных сведений о способах представления в программе данных разного вида (использование базовых типов и разработка пользовательских типов со структурным наполнением).

3. Знания теоретических основ, методов хранения и алгоритмов обработки данных разного вида (числовых, текстовых) на основе их модельных представлений с использованием массивов и списков.

4. Умения работы с инструментальными средствами разработки программного кода.

5. Навыки по разработке программного кода на C++ для решения практических задач из разных предметных областей.

3.2 Место дисциплины в структуре ОПОП

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

и обеспечивает изучение последующих дисциплин:

1. «Компьютерная математика»

2. «Алгоритмы и структуры данных»
3. «Организация ЭВМ и систем»
4. «Объектно-ориентированное программирование»
5. «Компьютерная графика»
6. «Сети ЭВМ»

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
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	Введение	1				1
2	Общие сведения о языке программирования С++	2	1	1		2
3	Стандартные (встроенные) типы данных, операторы, выражения	3	2	1		2
4	Организация консольного ввода/вывода	1	1	1		2
5	Основные управляющие конструкции в языке С++	2	4	1		2
6	Массивы и указатели	2	4	1		2
7	Организация ввода/вывода при работе с файлами	2	1	1	0	2
8	Функции	2	2	2	0	2
9	Разработка программ при работе с массивами	4	2	2	0	2
10	Структуры и их реализация в языке С++	2	2	1		6
11	Модульная структура программы	2	1	1		2
12	Общее введение в понятие класса и его реализация в языке С++	2	2	1		5
13	Представление и обработка текстов с использованием структурированных типов данных	3	4	1		10
14	Динамические структуры данных	2	4	2	1	6
15	Представление и обработка текстов с использованием линейного списка	2	4	1	2	2
16	Жизненный цикл и этапы разработки программ. Тестирование программ	1				4
17	Заключение	1				4
	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	Введение	Структура и содержание дисциплины, ее задачи. Содержание и форма проведения занятий. Формы доступа к материалам и заданиям дисциплины.
2	Общие сведения о языке программирования С++	Краткие сведения о языках программирования С/С++, история создания, стандарты, среды программирования. Общая характеристика синтаксиса языка С++. Структура и основные элементы программы, использованием внешних библиотек, препроцессорные директивы. Основные виды данных, способов хранения информации в программе и действий (управляющих конструкций) по обработке. Использование сред программирования.
3	Стандартные (встроенные) типы данных, операторы, выражения	Простые (встроенные) типы данных – основные понятия: переменные, константы, множество значений, инициализация, набор операций (арифметические, логические, поразрядные) и оператор присваивания, принципы формирования выражений (старшинство операций), битовое представление и побитовые операторы. Особенности представления и операций для встроенных типов данных – типы для представления целых значений, символьный тип, вещественные типы с плавающей точкой. Особенности реализаций операций для разных типов. Объявления и описания переменных, инициализация переменных, квалификатор const, директива #define, оператор typedef, перечисления enum. Выражения, правила их построения и вычисления. Преобразования типов. Специфические операторы языка С++ (инкремент и декремент, составной оператор присваивания, тернарный «условный» оператор и т.п.).
4	Организация консольного ввода/вывода	Стандартные потоки ввода и вывода языка С++. Ввод и вывод встроенных типов. Задание формата значений при вводе и выводе. Библиотека ввода/вывода для языка С и особенности задания формата представления значений. Проектирование пользовательского интерфейса.

№ п/п	Наименование темы дисциплины	Содержание
5	Основные управляющие конструкции в языке C++	Основные виды исполнения программного кода: последовательный, ветвление, повтор, обращение к предопределенным действиям (функциям). Синтаксис и семантика управляющих конструкций языка C++. Организация ветвления: конструкция if-else (полная и неполная), вариативный выбор (switch). Организация циклов: цикл с предусловием, цикл с постусловием, цикл с «известным числом повторений», особенности их синтаксиса и варианты использования (использование перехода и выхода – continue, break). Последовательность конструкций и оформление блоков. Вложенность управляющих конструкций, особенности реализации.
6	Массивы и указатели	Массивы. Индексирование. Объявление массивов. Одномерные, двумерные и многомерные массивы. Способы инициализации массивов. Ввод/вывод массивов. Строки и тексты как массивы символов. Проход по элементам массива. Массивы и указатели, доступ к элементам массива через указатель. Динамическое выделение памяти – встроенные возможности языка C++ и библиотека управления динамической памятью языка C. Использование указателей при работе с динамическими массивами.
7	Организация ввода/вывода при работе с файлами	Организация работы с файлами в языках C и C++. Поток и файлы. Типовые действия с файлами: вывод данных в файл, чтение данных из файла, прямой доступ к данным в файле. Форматирование ввода и вывода.
8	Функции	Объявление функций и их использование, синтаксические особенности оформления кода программы. Передача параметров по значению, по указателю и по ссылке. Вызов функции (аргументы функции) и возврат значения. Передача массивов в качестве аргументов. Типы возвращаемых значений. Области видимости (области действия). Глобальные и локальные имена. Внешние переменные. Автоматические переменные. Статические переменные. Использование функций при работе с массивами. Указатель на функцию. Использование функции как параметра другой функции; пример применения.
9	Разработка программ при работе с массивами	Примеры типовых действий с фрагментами массивов: задачи разделения и слияния массивов, перестановка сегментов массива (циклический сдвиг) и т.п. Линейный и бинарный поиск в массиве. Оптимальность алгоритма бинарного поиска. Функции бинарного поиска. Простые алгоритмы сортировки (выбором, вставками, обменов). 2 семестр

№ п/п	Наименование темы дисциплины	Содержание
10	Структуры и их реализация в языке C++	Определение структуры (struct) в языке C++. Обращение к полям структуры. Массивы структур. Примеры использования. Объединение (union) в языке C++. Использование полей, содержащихся в объединении. Рекомендации по применению.
11	Модульная структура программы	Структурирование программ. Представление программы в виде набора функций. Многофайловая структура программы. Прототипы функций и стиль их описания.
12	Общее введение в понятие класса и его реализация в языке C++	Общие сведения о синтаксисе и семантике классов, определяемых пользователем, в языке C++. Базовые концепции классов. Классы и объекты. Инкапсуляция данных и методов. Определение класса, использование класса, вызов методов класса. Объекты и способы доступа к ним. Конструкторы и деструкторы. Примеры использования.
13	Представление и обработка текстов с использованием структурированных типов данных	Строки как структура/класс на основе массива символов, использование динамической памяти. Определение, основные операции над строками. Представление строк (с явной длиной, с символом-ограничителем). Реализация операций над строками (поиск в строке, преобразования строки). Представление текста в виде структуры/класса на основе набора строк, использование динамической памяти. Определение, основные операции над текстом. Реализация операций над текстом (поиск в тексте, преобразования в тексте – по отдельным строкам и в целом). Библиотека обработка символьных строк string, функции для работы с данными типа string.

№ п/п	Наименование темы дисциплины	Содержание
14	Динамические структуры данных	<p>Линейный список с произвольным доступом. Представление в динамической памяти. Основные операции над линейным списком и их реализация на языке С++ с использованием структур/классов. Разновидности линейных списков: однонаправленные (Л1-списки), двунаправленные (Л2-списки), циклические списки. Реализация на языке С++. Примеры решения задач с линейными списками. 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 С++. Простой вид списков – однонаправленный линейный, использование указателей. Базовые концепции однонаправленного линейного списка: элемент списка (описание с использованием структуры/класса), указатель на следующий элемент. Базовые операции с однонаправленным линейным списком: создание пустого списка, удаление всего списка. Основные операции с однонаправленным линейным списком: вставка элемента (варианты), удаление элемента (варианты). Ссылочная реализация ограниченного Л1-списка на базе вектора (массива). Представление и реализация на языке С++. Сложно-структурированные списки: иерархические списки.</p>
15	Представление и обработка текстов с использованием линейного списка	<p>Хранение и обработка текстовой информации с использованием списков. Тексты, разделенные на слова и строки. Основные операции для работы с набором слов и строк. Представление и реализация линейного списка на языке С++ для хранения и обработки текстовой информации.</p>
16	Жизненный цикл и этапы разработки программ. Тестирование программ	<p>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 программ. Тестирование и отладка программ. Обеспечивающие мероприятия: документирование, подготовка и планирование тестирований и отладки.</p>
17	Заключение	<p>Модификация стандарта языка программирования С++, основные направления и особенности. Переход к сложно-структурированным форматам представления данных.</p>

4.2 Перечень лабораторных работ

Наименование лабораторной работы	Количество ауд. часов
1. Реализации линейной программы на языке С++ с организацией форматированного ввода-вывода.	1
2. Реализация программы с условными конструкциями.	1
3. Реализация программы с циклами для вычисления последовательности значений.	1
4. Использование файлов для ввода-вывода данных.	2
5. Реализация программы для обработки последовательности значений с использованием динамического массива.	2
6. Программирование функций пользователя.	2
7. Представление и обработка символьной информации в строке.	2
8. Представление и обработка символьной информации в тексте.	2
9. Реализация представления и базовых операций для линейного списка с произвольным доступом.	2
10. Реализация совместной обработки нескольких линейных списков с произвольным доступом.	2
Итого	17

4.3 Перечень практических занятий

Наименование практических занятий	Количество ауд. часов
1. Организация пользовательского интерфейса. Форматированный ввод-вывод. Особенности реализации линейной программы на языке С++. Типы с плавающей точкой, символьный тип. Основные операции.	2
2. Программирование ветвлений. Типы с плавающей точкой. Основные операции.	1
3. Программирование циклов. Рекуррентное вычисление суммы ряда, разбор примера. Целочисленные типы. Основные операции.	1
4. Одномерные массивы. Использование файлов для ввода/вывода.	2
5. Одномерные массивы. Адаптация программы на основе использования динамической памяти.	2
6. Двумерные массивы. Использование файлов для ввода/вывода.	4
7. Реализация функций. Передача параметров.	4
8. Методы обработки данных в массивах – поиск и сортировка.	1
9. Использование структур и классов.	2
10. Модульная организация программы.	1
11. Основные задачи по обработке символьной информации. Представление и обработка строк.	2
12. Основные задачи по обработке символьной информации. Представление и обработка текста.	2
13. Однонаправленные линейные списки: представление и основные операции обработки.	5

Наименование практических занятий	Количество ауд. часов
14. Использование сложно-структурированных списков для представления и обработки данных.	5
Итого	34

4.4 Курсовое проектирование

Цель работы (проекта): 3 семестр: полное решение содержательной задачи на основе представления данных с помощью массивов и разбиения программы на функции..

Содержание работы (проекта): 1 семестр: курсовая работа предполагает разработку студентами способа хранения данных на основе выбранной ими структуры массивов (с возможным использованием динамической памяти). Задание выдается индивидуально каждому студенту. При выполнении курсовой работы студенты должны выбрать способ хранения данных в массиве, провести декомпозицию общей задачи с разбиением программы на функции; определить формат хранения информации в файлах, разработать пользовательский интерфейс. Помимо указанного, каждый студент должен разработать алгоритм решения задачи в соответствии с ее содержательной формулировкой.

2 семестр: курсовая работа предполагает разработку студентами способа хранения данных на основе выбранной ими структуры линейных списков. Задание выдается индивидуально каждому студенту. При выполнении курсовой работы студенты должны выбрать способ хранения данных в динамической памяти, провести декомпозицию общей задачи с разбиением программы на функции; определить формат хранения информации в файлах, разработать пользовательский интерфейс. Помимо указанного, каждый студент должен разработать алгоритм решения задачи в соответствии с ее содержательной формулировкой.

Требования по оформлению курсовой работы.

Критерии оценивания курсовой работы..

Примерные темы:

№ п/п	Название темы	Перевод темы
1	Обработка математических данных	Mathematical data processing
2	Редактирование текстовой информации	Text information editing

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.

Текущая СРС	Примерная трудоемкость, ач
Работа с лекционным материалом, с учебной литературой	10
Опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	10
Самостоятельное изучение разделов дисциплины	15
Выполнение домашних заданий, домашних контрольных работ	0
Подготовка к лабораторным работам, к практическим и семинарским занятиям	10
Подготовка к контрольным работам, коллоквиумам	0
Выполнение расчетно-графических работ	0
Выполнение курсового проекта или курсовой работы	0
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	11
ИТОГО СРС	56

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.	111
2	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
3	Дейтел, Харви. Как программировать на С++ [Текст] : [Учеб.] / Х.М.Дейтел,П.Д.Дейтел; Пер. с англ. под ред. В.Г.Кузьменко, 2001. -1151 с.	32
4	Калмычков, Виталий Анатольевич. Использование структурированных данных при программировании на языке С++ [Электронный ресурс] : учеб. пособие / В. А. Калмычков, 2011. -1 эл. опт. диск (CD-ROM)	неогр.
Дополнительная литература		
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	Круглински, Дэвид Дж. Программирование на Microsoft Visual С++ 6.0 для профессионалов [Текст] : [учебное пособие] / Д.Дж. Круглински; С.Уингоу ,Дж. Шеферд; Пер. с англ, 2001. -819 с.	26
3	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
4	Топп, Уильям. Структуры данных в С++ [Текст] : монография / У. Топп; У.Форд; Пер. с англ. под ред. В.Кузьменко, 2000. -815 с.	25
5	Алексеев, Андрей Юрьевич. Динамические структуры данных [Текст] : учеб. пособие / А.Ю. Алексеев, С.А. Ивановский, Д.В. Куликов, 2004. -83 с.	151

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

№ п/п	Электронный адрес
1	Бьерн Страуструп. Язык программирования С++ http://lib.ru/СРРНВ/cpptut.txt

№ п/п	Электронный адрес
2	Бьерн Страуструп. Программирование: Принципы и практика использования С++ (на англ.) = Programming --Principles and Practice Using С++ http://www.stroustrup.com/Programming/
3	Электронный ресурс дисциплины на сайте вуза http://eplace.eltech.ru/lotus/myquickr/progr-off

6 Критерии оценивания и оценочные материалы

6.1 Критерии оценивания

Для дисциплины «Программирование» формой промежуточной аттестации является экзамен. Оценивание качества освоения дисциплины производится с использованием рейтинговой системы.

Экзамен

Оценка	Количество баллов	Описание
Неудовлетворительно	0 – 49	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	50 – 65	теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практически навыки и умения работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки
Хорошо	66 – 82	теоретическое содержание курса освоено полностью, без пробелов, некоторые практически 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	83 – 100	теоретическое содержание курса освоено полностью, без пробелов, необходимые практически навыки и умения сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено количеством баллов, близким к максимальному

Особенности допуска

Текущий контроль включает в себя выполнение разработки программного кода, сдачу в срок отчетов и их защиту по всем практическим заданиям и лабораторным работам, по результатам которых студент получает допуск к сдаче курсовой работы.

Текущий контроль при выполнении курсовой работы осуществляется в соответствии с заданием на курсовую работу. Текущий контроль включает в себя проверку выполнения разработки программного кода, предъявление программы на компьютере, сдачу в срок пояснительной записки и ее защиту, по результатам которой студент получает допуск на экзамен.

В каждом семестре итоговая оценка за экзамен формируется на основе совокупности баллов, набранных студентом по итогам текущей аттестации в форме выполнения лабораторных работ и промежуточной аттестации в форме курсовой работы (по результатам контрольных точек №1.1, №1.2 и итоговой контрольной точки 1 семестра; по результатам контрольных точек №2.1, №2.2 и итоговой контрольной точки 2 семестра) и итогам промежуточной аттестации в форме экзамена. Максимальная совокупность баллов – 100.

Максимальное количество баллов по результатам 3 контрольных точек – 15. На каждой из 3 контрольных точек выставляемая при аттестации оценка формирует соответствующий балл:

«отлично» – 5 баллов (максимальный), «хорошо» – 4 балла, «удовлетворительно» – 3 балла, «неудовлетворительно» – 2 балла, «неаттестация» – 1 балл, «не изучено» – 0 баллов (минимальный).

Максимальное количество баллов по результатам экзамена – 85. Баллы набираются на экзамене пропорционально отдельным заданиям.

Итоговая оценка промежуточной аттестации в форме экзамена формируется на основе совокупности баллов за все виды аттестации в соответствии со следующими диапазонами баллов:

83 балла и более – «отлично», 66 баллов и менее 83 баллов – «хорошо», 50 баллов и менее 66 баллов – «удовлетворительно», более 15 баллов и менее 50 баллов – «неудовлетворительно», более 0 баллов и не более 15 баллов – «неаттестация», 0 баллов – «не изучено».

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

Примерные вопросы к экзамену

№ п/п	Описание
1	Понятие алгоритма, свойства, виды, формы его записи. Примеры.
2	Переменная основных типов языка C++: диапазоны, особенности представления в памяти, операции.
3	Константы основных типов языка C++. Перечисление (enum).
4	Операторы языка C++ и их классификация. Таблица приоритета выполнения операторов.
5	Функция main. Составной оператор для задания линейной последовательности действий.
6	Ветвление -инструкция if и ее виды, инструкция switch. Примеры.
7	Циклические алгоритмы: инструкция while, do-while, for и организация цикла, инструкции continue и break. Примеры. Итерационные формулы.
8	Производные типы языка C++: ссылки, указатели.
9	Организация ввода и вывода данных в программе -стандартные потоки cin и cout.
10	Организация работы с файлами -файловые потоки, основные действия.
11	Манипуляторы и функции, используемые при потоковом вводе и выводе данных. Примеры.
12	Производные типы языка C++: массивы (одномерные), представление в памяти, инициализация, обращение к элементам. Примеры.
13	Производные типы языка C++: массивы (многомерные), представление в памяти, инициализация, обращение к элементам. Примеры.
14	Использование указателей при работе с массивами, обращение к элементам массива.
15	Двухмерные массивы -использование указателей при их обработке: указатель на вектор из N компонент, вектор указателей на строки.
16	Варианты представления результата решения задачи при использовании массивов: тот же или другой массив, индексный массив.
17	Функции: определение, список аргументов (формальных параметров), вызов, описание, возвращаемое значение, инструкция return, варианты передачи одномерных и многомерных массивов в качестве параметров. Примеры.
18	Структура текста программы в файле. Локальные и глобальные имена. Области видимости имен.

19	Структура (разделы) текста программы в файле. Определение переменных в инструкциях. Виды памяти.
20	Технологические этапы получения исполняемой программы. Понятия: препроцессор, транслятор. Директивы препроцессора и их обработка.
21	Алгоритмы поиска в массиве: линейный поиск, двоичный (бинарный) поиск.
22	Алгоритмы сортировки данных в массиве: с помощью вектора индексов, выбором (min и/или max), обменом (пузырьковая) с модификациями (контроль наличия перестановок, с плавающим изменением границы перестановок, изменение направления), сортировка вставкой с поиском (линейным, двоичным).
23	Работа с динамической памятью – операторы new и delete.
24	Пользовательские типы: структура (особенности реализации), класс (начальные сведения об особенностях реализации), объединение (назначение и использование).
25	Представления текстовой строки с маркером: пользовательский тип с однотипными данными, ввод и вывод, включая работу с файлами, вставка и удаление символа.
26	Представления текстовой строки с длиной: пользовательский тип с однотипными данными, ввод и вывод, включая работу с файлами, вставка и удаление символа.
27	Библиотека string.h: группы функций для обработки текстовой строки.
28	Представление текста как массива текстовых строк: строки с маркером, строки с длиной, преобразованию текста в пределах строки, преобразованию текста (вставка и удаление строк), динамическое выделение памяти.
29	Многофайловая организация программы: понятие проекта, структурный состав файлов, заголовочные файлы, области видимости, тип памяти extern.
30	Набор пользовательских типов для представления композиционного объекта.
31	Линейный список с произвольным доступом.
32	Однонаправленный список (Л1-список): реализация на языке C++, основные операции по модификации (вставка, удаление элемента).
33	Двунаправленный список (Л2-список): реализация на языке C++, основные операции по модификации (вставка, удаление элемента).
34	Циклический список. Реализация на языке C++.
35	Иерархические списки. Реализация на языке C++.
36	Хранение и обработка текстовой информации с использованием списков.

Вариант экзаменационного теста

1 семестр.

1. Что будет выведено на экран при выполнении следующей программы (в ответе разделителем является 1 пробел) ?

```
void main()
{ int x=1, y=1, z;
  if ( z=y<0 ) x=3; else
```

```

        if (y==0) x=5; else x=7;
        cout<< x << ' ' << z <<"\n";
    }

```

2. Заданы константа, массив и функция:

```

const N=10; int Y[N];

unsigned Control(int M[N])
{
    unsigned I=0;
    while((I<N)&&(M[I]>0))
        I++;
    return (I<=N);
}

```

Если в данном массиве все элементы положительные, то при обращении к функции Control(Y) какое будет возвращено значение?

3. Что будет выведено на экран при выполнении следующей программы (в ответе разделителем является 1 пробел) ?

```

void main()
{
    int x, y;
    x=y=0;
    while( y<10 ) x += ++y;
    cout<< x << ' ' << y <<"\n";
}

```

4. Что будет выведено на экран после символа ”=” при выполнении следующей программы ?

```

void main()

```

```

{ int x = 1;

  cout<<"( x | x )=" << ( x | x )<<"\n";

}

```

5. Что будет выведено на экран при выполнении следующей программы (в ответе разделителем является 1 пробел) ? Если значение является неопределенным, то вместо числа напишите символ @.

```

int a[ ] = { 0, 1, 2, 3, 4 };

main()

{int i, *p;

  for(p=&a[0],i=1;i<=5;i++ )

    cout<<p[i]<<' ';

}

```

6. Для правильного построчного вывода элементов двухмерного массива, описанного как float A[10][20], в реализации функции

```

void outArray(int realI, int realJ, float *p)

{int i, j;

  for(i=0; i<realI; i++)

    {cout<<' ' <<i<<"- string: ";

      for(j=0; j<realJ; j++) cout<<" " <<_____ ;

      cout<<"\n";

    }

  cout<<"\n";

}

```

при обращении к ней в виде


```
outArray(10, 20, &A[0][0])
```

что требуется вместо пропущенного фрагмента записать?

7. Что обеспечивает текст программы:

```
#include <fstream.h>

void main()

{ifstream infile;

ofstream outfile;

infile.open("filename.txt");

outfile.open("filename.dat");

}
```

8. Для указателя `float *float_point2=new float`

что означает выполнение присваивания `*float_point2=3.14159 ?`

2 семестр.

1. Что позволяет выполнить функция:

```
void swap(char *s1, char *s2)

{char tmp;

tmp=*s1; *s1=*s2; *s2=tmp;

}
```

2. Элемент линейного однонаправленного списка определен как:

```
struct elem {

elem *next;

};
```

Указатель `elem* first` указывает на первый элемент этого списка.

После выполнения фрагмента программы:

```
elem *temp;
```

```
temp = first;
```

```
while (temp!=NULL) temp = temp->next;
```

что позволяет узнать значение указателя temp?

3. Что будет выведено на экран при выполнении следующей программы:

```
void main()
```

```
{char input2[]="SSSWILTECH1\1\11W\1WALLMP1";
```

```
int i; char c;
```

```
for ( i=2; (c=input2[i])!='\0'; i++ )
```

```
    {switch(c) {
```

```
        case 'a': cout<<'i'; continue;
```

```
        case '1': break;
```

```
        case 1: while ( (c=input2[++i]) !='\1' && c!='\0');
```

```
        case 9: cout<<'S';
```

```
        case 'E': case 'L': continue;
```

```
        default: cout<<c; continue; }
```

```
        cout<<' ';
```

```
    }
```

```
    cout<<'n';
```

```
}
```

4. Элемент списка описан как:

```
struct elem
```

```
{float inf; elem *next;};
```

С помощью указателя `elem* firFIFO` обеспечивается доступ к первому элементу этого списка (для пустого списка `firFIFO==NULL`). В программе реализован следующий фрагмент:

```
elem *p=firFIFO;
while(firFIFO)
{p=firFIFO;
  firFIFO=firFIFO-> next;
  delete p;
}
```

Что обеспечивает этот фрагмент?

5. В программе определен пользовательский тип данных:

```
struct date {
  int month;
  int day;
  int year;
};
```

В тексте функции

```
void main()
{date *date_point; date_point = new date;
  date_point month = 10;
  date_point day = 18;
  date_point year = 1938;
  delete date_point;
```

```
}
```

имеются ли синтаксические ошибки в 3 последних операторах присваивания?

6. Элемент списка описан как:

```
struct elem  
{float inf; elem *next;};
```

С помощью указателя `elem* firFIFO` обеспечивается доступ к первому элементу этого списка (для пустого списка `firFIFO==NULL`). В программе реализован следующий фрагмент:

```
elem *p=firFIFO; float i=0;  
if(firFIFO)  
{i=firFIFO->inf;  
firFIFO=firFIFO-> next;  
delete p;  
}
```

Что обеспечивает этот фрагмент?

7. Элемент списка описан как:

```
struct elem  
{unsigned int inf; elem *next;};
```

С помощью указателя `elem* firFIFO` обеспечивается доступ к первому элементу этого списка (для пустого списка `firFIFO==NULL`). В программе реализован следующий фрагмент:

```
elem *p=firFIFO;  
if(firFIFO)
```

```

{while(p->next) p=p->next;

p->next=new elem;

p->next->next=NULL;

p->next->inf=10;

}

else

{firFIFO=new elem;

firFIFO->inf=77;

firFIFO->next=NULL;

}

```

Что обеспечивает этот фрагмент?

Образцы задач (заданий) для контрольных (проверочных) работ

1. Какое значение будет присвоено переменной z после выполнения следующего фрагмента программы:

```

unsigned int x, y, z; // отводится по 2 байта

x = 0x0faa;

y = 0x55b0;

z = (x << 8) + (y >> 8);

```

Результат запишите как шестнадцатеричное значение языка C++.

2. Что будет выведено на экран при выполнении следующей программы (без учета символа перехода к новой строке) (в ответе разделителем является 1 пробел) ?

```

void main()

{ int x=1, y=1, z;

```

```
if (x=z=y); x=3;

cout<< x << ' ' << z <<"\n";

}
```

3. Выполнение каких действий обеспечивает инструкция return?

4. Реализовать шейкерную сортировку по возрастанию с контролем наличия обмена элементов (необходимые идентификаторы описаны):

```
void main()

{const int L=15; // максимальное число элементов массива

float x[L] = { 12, -3, 25, 8, -23, 14, 0 }; // зарезервированный объем памяти
ПОД МАССИВ

unsigned n = 7; // реальное число используемых элементов массива

unsigned i;

unsigned left = -1;

unsigned right = n-1;

float tmp;

.....

}
```

6.3 График текущего контроля успеваемости

Неделя	Темы занятий	Вид контроля
1	Введение	
2	Общие сведения о языке программирования С++	
3	Организация консольного ввода/вывода	
4	Основные управляющие конструкции в языке С++	
5		
6		
7		Отчет по лаб. работе
8	Массивы и указатели	
9	Организация ввода/вывода при работе с файлами	
10	Функции	
11		Отчет по лаб. работе
12	Жизненный цикл и этапы разработки программ. Тестирование программ	
13		
14		
15		
16		Защита КР / КП

6.4 Методика текущего контроля

1. Методика текущего контроля на лекционных занятиях.

Текущий контроль включает в себя контроль посещаемости (не менее 70% занятий на основе системы контроля вуза), а также усвоения лекционного материала в форме контрольной работы, по результатам которого студент получает оценку.

2. Методика текущего контроля выполнения индивидуальных домашних заданий.

Текущий контроль включает в себя контроль полноты и правильности подготовки к лабораторным занятиям. В случае частичной неправильности подготовки задание считается не выполненным, а студент не допускается к сдаче индивидуального задания и должен заново выполнить подготовку по индивидуальному заданию. При наличии выполнения подготовки индивидуальных заданий студент получает допуск к их сдаче.

3. Методика текущего контроля итогов практических и лабораторных занятий.

Порядок рассмотрения практических заданий и выполнения лабораторных работ, подготовки отчетов и их защиты. В процессе обучения по дисциплине «Программирование» студент обязан выполнить в 1 семестре практические индивидуальные задания при подготовке к 5 лабораторным работам и во 2 семестре практические индивидуальные задания при подготовке к 4 лабораторным работам. Под выполнением лабораторных работ подразумевается подготовка на практических занятиях, написание программного кода в соответствии с индивидуальным заданием, сдача программ на компьютере, подготовка отчетов по результатам практических занятий и выполнения лабораторных работ, их защита с контролем по контрольным точкам.

Выполнение заданий студентами осуществляется индивидуально. Оформление отчета студентами осуществляется индивидуально в соответствии с принятыми в СПбГЭТУ правилами оформления студенческих работ. Отчет оформляется в процессе подготовки практических индивидуальных заданий и после написания программного кода, представляется преподавателю на проверку. После проверки отчет либо возвращается (при наличии замечаний) на доработку, либо допускается к защите.

Отчеты защищаются студентами индивидуально. Каждый студент получает вопрос по практической форме реализации программного кода, 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, отчет считается защищенным.

На защите отчета студент должен показать: понимание синтаксиса и семантики используемых программном коде синтаксических единиц языка программирования, методики реализации решения индивидуального задания и зна-

ние особенностей применения синтаксических единиц, понимание и умение объяснять особенности применяемых методов, возможные области их применения и т.д., умение давать оценку реализованному программному коду и объяснять реакции программы на разные обрабатываемые данные, приобретенные навыки и умения.

Текущий контроль включает в себя выполнение разработки программного кода, сдачу в срок отчетов и их защиту по всем практическим заданиям и лабораторным работам, по результатам которых студент получает допуск к сдаче курсовой работы.

В случае систематического пропуска занятий и/или полного отсутствия выполнения индивидуальных заданий без защищенных лабораторных работ по результатам контрольных точек №1.2 и №2.2 студент считается явно неуспевающим и должен быть рекомендован на рассмотрение к отчислению.

4. Методика текущего контроля самостоятельной работы студентов.

Контроль самостоятельной работы студентов осуществляется на лекционных, лабораторных и практических занятиях студентов и по итогам выполнения индивидуальных домашних заданий по методикам, описанным в п.п. 1-3.

5. Методика текущего контроля выполнения курсовой работы.

Текущий контроль при выполнении курсовой работы осуществляется в соответствии с заданием на курсовую работу. Текущий контроль включает в себя проверку выполнения разработки программного кода, предъявление программы на компьютере, сдачу в срок пояснительной записки и ее защиту, по результатам которой студент получает допуск на экзамен.

Оформление пояснительной записки на курсовую работу выполняется в соответствии с требованиями к студенческим работам, принятым в СПбГЭТУ.

Защита курсовой работы осуществляется в соответствии с требованиями

«Положения о промежуточной аттестации».

График текущего контроля успеваемости

Неделя	Тема занятия	Вид контроля
1	Реализация линейной программы с организацией форматированного ввода-вывода	Выдача ИЛЗ №1.1
2	Представление результатов ИЛЗ №1.1	
3	Реализация программы с условными конструкциями	Выдача ИЛЗ №1.2
4	Представление результатов ИЛЗ №1.2	
5	Реализация программы с циклами. Использование файлов для вывода данных	Выдача ИЛЗ №1.3
6	Представление результатов ИЛЗ №1.3	
7	Реализация программы с использованием динамического массива. Использование файлов для ввода-вывода данных	Выдача ИЛЗ №1.4 Контр. точка №1.1
8	Представление результатов ИЛЗ №1.4	
9	Программирование функций пользователя. Использование файлов для ввода-вывода данных	Выдача ИЛЗ №1.5
10	Представление результатов ИЛЗ №1.5	
11	Обработка математических данных. Особенности реализации задания на курсовую работу	Выдача ИКЗ №1
12	Представление результатов ИКЗ №1	
13	Представление результатов ИКЗ №1	Контр. точка №1.2
14	Представление результатов ИКЗ №1	
15	Представление результатов ИКЗ №1	
16	Представление результатов ИКЗ №1	Итоговая контр. точка
17		Зачетная неделя

Неделя	Тема занятия	Вид контроля
1	Представление и обработка символьной информации в строке	Выдача ИЛЗ №2.1
2	Представление результатов ИЛЗ №2.1	Выдача ИКЗ №2
3	Представление и обработка символьной информации в тексте	Выдача ИЛЗ №2.2
4	Представление результатов ИЛЗ №2.2	
5	Реализация представления и базовых операций для линейного списка с произвольным доступом	Выдача ИЛЗ №2.3
6	Представление результатов ИЛЗ №2.3	Контр. точка №2.1
7	Реализация совместной обработки нескольких линейных списков с произвольным доступом	Выдача ИЛЗ №2.4
8	Представление результатов ИЛЗ №2.4	
9	Редактирование и обработка текстовой информации. Особенности задания на курсовую работу	Уточнение ИКЗ №2
10	Представление результатов ИКЗ №2	
11	Представление результатов ИКЗ №2	Контр. точка №2.2
12	Представление результатов ИКЗ №2	
13	Представление результатов ИКЗ №2	
14	Представление результатов ИКЗ №2	
15	Представление результатов ИКЗ №2	
16	Представление результатов ИКЗ №2	Итоговая контр. точка
17		Зачетная неделя

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	Аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, проектор, экран, учебная доска, персональный компьютер (не менее, чем 1 компьютер на 2 обучающихся)	1) Windows 7 и выше; 2) Microsoft Office 2007 и выше 3) Microsoft Visual Studio 2008 и выше
Практические занятия	Аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, проектор, экран, учебная доска, персональный компьютер (не менее, чем 1 компьютер на 2 обучающихся)	1) Windows 7 и выше; 2) Microsoft Office 2007 и выше 3) Microsoft Visual Studio 2008 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

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