

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Галунин Сергей Александрович
Должность: проректор по учебной работе
Дата подписания: 07.09.2023 11:23:34
Уникальный программный ключ:
08ef34338325bdb0ac5a47baa5472ce36cc3fc3b

Приложение к ОПОП
«Организация и программирова-
ние интеллектуальных систем»



СПбГЭТУ «ЛЭТИ»
ПЕРВЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ

МИНОБРНАУКИ РОССИИ

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

РАБОЧАЯ ПРОГРАММА

ДИСЦИПЛИНЫ

«ПРОГРАММИРОВАНИЕ»

для подготовки бакалавров

по направлению

09.03.01 «Информатика и вычислительная техника»

по профилю

«Организация и программирование интеллектуальных систем»

Санкт-Петербург

2023

ЛИСТ СОГЛАСОВАНИЯ

Разработчики:

доцент, к.ф.-м.н., доцент Хахаев И.А.

Рабочая программа рассмотрена и одобрена на заседании кафедры ВТ
05.05.2023, протокол № 4

Рабочая программа рассмотрена и одобрена учебно-методической комиссией
ФКТИ, 18.05.2023, протокол № 4

Согласовано в ИС ИОТ

Начальник ОМОЛА Загороднюк О.В.

1 СТРУКТУРА ДИСЦИПЛИНЫ

Обеспечивающий факультет	ФКТИ
Обеспечивающая кафедра	ВТ
Общая трудоемкость (ЗЕТ)	11
Курс	1
Семестр	2, 1
Виды занятий	
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
Вид промежуточной аттестации	
Экзамен (курс)	1
Курсовая работа (курс)	1
Экзамен (курс)	1
Курсовая работа (курс)	1

2 АННОТАЦИЯ ДИСЦИПЛИНЫ

«ПРОГРАММИРОВАНИЕ»

Дисциплина «Программирование» нацелена на изучение и освоение базовых понятий, методов и приемов программирования на языке Си и является базовой в программистском образовании студентов, обучающихся по направлению подготовки 09.03.01-«Информатика и вычислительная техника». Курс содержит лекционные занятия, лабораторные работы и курсовые работы, при выполнении которых студенты овладевают практическими навыками разработки программного обеспечения на языке Си.

SUBJECT SUMMARY

«PROGRAMMING»

Discipline "Programming" is aimed at the study and mastering of basic concepts, methods and techniques of programming in C language. It is the basic programming in the education of students under the direction of training N 09.03.01- "Information Science and Engineering". The course contains lectures, laboratory work and coursework, where students acquire practical skills in software development on C language.

3 ОБЩИЕ ПОЛОЖЕНИЯ

3.1 Цели и задачи дисциплины

1. Целями освоения дисциплины является изучение способов постановки и описания задач для решения на ЭВМ, выбора оптимальных способов их решения, приобретение знаний основных методов и приемов разработки структурированных алгоритмов и программ, в том числе с типами данных, определяемых пользователем, в том числе при использовании рекомендуемых правил оформления кода.

2. Задачи дисциплины:

-самостоятельно определять круг задач в рамках поставленной цели, осуществлять постановку и спецификацию задач для решения на ЭВМ;

-применять методы структурной (функциональной) декомпозиции задач для разделения программы на части и ее реализации как совокупности функций, выбирать подходящие структурированные (в том числе динамические) данные, их представлять и реализовывать;

-анализировать и сопоставлять альтернативные представления и реализации структурированных данных с точки зрения производительности программ и затрат памяти.

3. Знание базовых конструкций программирования и стандартных типов данных, основных процедурных элементов языка Си (стандартные типы данных; операторы языка, реализующие основные управляющие структуры; функции), технологии работы на ЭВМ, правил и умений диалоговой работы при программировании типовых задач в интегрированной среде, способов тестирования и отладки разработанных программ, знание языка программирования Си в необходимом для решения типовых задач объеме.

4. Умения:

- разрабатывать алгоритмы и программы, пригодные для практического применения (алгоритмизации, кодирования, тестирования и отладки программ, работающих со структурированными данными, при программировании типовых задач);
- самостоятельно составлять, тестировать, отлаживать и документировать программы на языке Си на базовом уровне в парадигме процедурного программирования;
- пользоваться интегрированной средой программирования для конструирования, тестирования и отладки программ;
- формулировать функциональные требования к ключевым фрагментам программ, анализировать код с точки зрения выполнения этих требований.

5. Навыки:

- записи и документирования алгоритмов и программ, в том числе при использовании рекомендуемых правил оформления кода;
- чтения, анализа, понимания и модификации программ; освоения методик использования программных средств для решения практических задач;
- применения современных информационных компьютерных технологий;
- технологических приемов диалоговой работы на ЭВМ в интегрированной среде программирования.

3.2 Место дисциплины в структуре ОПОП

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

и обеспечивает изучение последующих дисциплин:

1. «Алгоритмы и структуры данных»
2. «Объектно-ориентированное программирование»
3. «Организация ЭВМ и систем»

4. «Операционные системы»
5. «Основы машинного обучения»
6. «Теория автоматов»
7. «Введение в разработку интеллектуальных систем»
8. «Компьютерная графика»
9. «Производственная практика (технологическая (проектно-технологическая) практика)»
10. «Организация процессов и программирование в среде Linux»
11. «Основы компьютерного зрения»
12. «Производственная практика (научно-исследовательская работа)»
13. «Цифровая обработка сигналов»
14. «Микропроцессорные системы»
15. «Параллельные алгоритмы и системы»
16. «Производственная практика (преддипломная практика)»

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-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
<i>УК-2.2</i>	<i>Определяет круг задач в рамках выбранных видов профессиональной деятельности, планирует собственную деятельность исходя из имеющихся ресурсов, решает поставленные задачи, использует нормативно-правовую документацию профессиональной сферы</i>
ОПК-8	Способен разрабатывать алгоритмы и программы, пригодные для практического применения;
<i>ОПК-8.2</i>	<i>Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ</i>
<i>ОПК-8.3</i>	<i>Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач</i>
ОПК-9	Способен осваивать методики использования программных средств для решения практических задач.
<i>ОПК-9.1</i>	<i>Знает методики использования программных средств для решения практических задач</i>
<i>ОПК-9.2</i>	<i>Умеет использовать программные средства для решения практических задач</i>
<i>ОПК-9.3</i>	<i>Имеет навыки использования программных средств для решения практических задач</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	2				2
2	Этапы и проблемы решения задач с использованием компьютера	4				12
3	Общие сведения о языке программирования Си и системах программирования	2			2	16
4	Стандартные (базовые) типы данных, операторы, выражения	2	2			16
5	Основные управляющие структуры программирования и управляющие инструкции в языке Си	2	2	3		13
6	Приемы программирования циклов	2	4	6		8
7	Функции и структура программы	4	2	3		21
8	Организация ввода/вывода и работа с файлами	2	1		2	17
9	Размещение данных в памяти. Явное управление памятью.	2	1	3		6
10	Массивы и указатели	4	2	6		22
11	Разработка программ при работе с массивами и указателями	2	1	3		14
12	Обработка текстов с использованием структурированных типов данных	4	2	6		14
13	Тестирование и отладка программ	2		4		8
14	Структуры как базовый тип данных и их реализация в языке программирования	4	2	4		12
15	Динамические структуры данных (структуры, указатели и рекурсивные типы данных)	6	2	2		14
16	Линейный список как абстрактный тип данных (АТД)	6	2	4		18
17	Разновидности списков	6	4	4		18
18	Битовые операции	4	3	3		6
19	Рекурсивные и вариационные функции	4	2		2	
20	Жизненный цикл и этапы разработки программ. Тестирование программ	2	2			
21	Заключение	2				
	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	Введение	Цель и задачи дисциплины. Краткое содержание, место и роль в формировании компетенций специалиста в области программирования. Форма проведения занятий. Связь с другими дисциплинами учебного плана направления (специальности).
2	Этапы и проблемы решения задач с использованием компьютера	Решение задач на компьютере. Уровни вычислительных систем: архитектура компьютера, операционные системы, методы и технологии программирования, прикладные технологии, информационные системы. Виды программного обеспечения (ПО): программа, программный комплекс, программный продукт (изделие), программная система. Основные показатели качества ПО. Алгоритмы. Свойства алгоритмов. Структурное программирование. Жизненный цикл ПО: основные этапы и процессы, их соотношение с содержанием и видами занятий учебной дисциплины.
3	Общие сведения о языке программирования Си и системах программирования	Языки программирования и системы программирования. История создания языка Си. Краткая характеристика языка Си. Структура и основные элементы программы. Классификация действий и данных. Пример программы на языке Си. Система программирования. Трансляция программ и выполнение программы. Стандарты Си. Системы программирования на Си для различных вычислительных платформ и операционных сред.
4	Стандартные (базовые) типы данных, операторы, выражения	Простые стандартные типы данных (схема рассмотрения: множество значений, переменные, константы, набор операций, выражения, битовое представление, побитовые операторы, оператор присваивания). Целочисленные типы, особенности представления и вычислений. Символьный тип. Особенности реализаций. Способы использования символьного типа в программах. Особенности построения логических выражений. Сокращенное (неполное) вычисление логических выражений. Вещественные типы с плавающей точкой. Особенности машинного представления вещественных чисел. Свойства машинной арифметики. Машинное ϵ . Выражения. Преобразования типов. Приведение типов. Специальные операторы в языке Си (инкремент и декремент, составной оператор присваивания, тернарный условный оператор, оператор запятой и т.п.). Объявления и описания переменных, инициализация переменных, квалификатор <code>const</code> , директива <code>#define</code> . Ввод и вывод значений стандартных типов с использованием стандартных потоков и файлов.

№ п/п	Наименование темы дисциплины	Содержание
5	Основные управляющие структуры программирования и управляющие инструкции в языке Си	Основные управляющие структуры программирования: последовательность, выбор (ветвление), итерации (циклы). Семантика управляющих структур и инструкции языка Си. Операторы и блоки. Выбор вариантов: оператор (инструкция) if, расширение оператора if, множественный выбор. Итерации: цикл с предусловием, цикл с постусловием, цикл со счетчиком. Другие управляющие операторы. Примеры преобразования управляющих конструкций в соответствии с рекомендациями структурного программирования.
6	Приемы программирования циклов	Итерация как базисная вычислительная схема (рекуррентные вычисления). Рекуррентные вычисления с целочисленными типами. Рекуррентные вычисления с вещественными типами. Программирование циклов в языке Си. Вложенные циклы. Циклы со сложным условием продолжения. Пред-и постутверждения, инвариант и вариант цикла. Примеры.
7	Функции и структура программы	Создание и использование функций. Вызов функции (аргументы функции) и возврат значения. Передача параметров по значению и по ссылке. Области видимости (области действия). Глобальные и локальные переменные. Области видимости и классы памяти. Внешние переменные. Автоматические переменные. Статические переменные. Представление программы в виде набора функций. Многофайловая структура программы.
8	Организация ввода/вывода и работа с файлами	Файловые системы языков С и С++. Потоки и файлы. Последовательность, как модель потока. Стандартные потоки ввода и вывода. Файловые потоки. Типовые действия с файлами: генерация, чтение, копирование. Форматирование ввода и вывода. Схема однопроходных алгоритмов обработки входных потоков (файлов) вычисление функций на последовательностях. Примеры.
9	Размещение данных в памяти. Явное управление памятью.	Размещение данных в памяти: статическое, автоматическое и динамическое выделение памяти. Примеры. Управление памятью во время выполнения программы в языке С. Примеры.
10	Массивы и указатели	Массивы. Индексирование. Объявление массивов. Двумерные и многомерные массивы. Ввод-вывод массивов. Строки и тексты как массивы символов. Массивы и указатели. Правила сложных объявлений. Использование функций при работе с массивами. Указатель на функцию. Использование функции как параметра другой функции; пример применения -итерационные методы решения нелинейных уравнений.

№ п/п	Наименование темы дисциплины	Содержание
11	Разработка программ при работе с массивами и указателями	Представление массивов при записи предусловия, постусловия и инварианта цикла. Примеры: задачи разделения и слияния массивов, перестановка сегментов массива (циклический сдвиг) и т.п. Линейный и бинарный поиск в массиве. Оптимальность алгоритма бинарного поиска. Оптимизация программы бинарного поиска. Простые алгоритмы сортировки (выбором, вставками, обменами). Работа с двумерными и многомерными массивами.
12	Обработка текстов с использованием структурированных типов данных	Строки. Определение, операции над строками. Представление строк в языке Си (с явной длиной, с символом-ограничителем). Реализация типовых операций над строками (анализ и редактирование строк). Функции для работы с строковыми данными. Обработка текстов. Тексты, разделенные на слова. Представление и обработка текстов, разделенных на страницы. Модульная структура программы как средство абстракции данных. Технологическое обеспечение многофайловой организации программы.
13	Тестирование и отладка программ	Причины появления ошибок в программе. Тесты. Рекомендации по выбору тестов. Методы «ручного» тестирования. Тестирование и отладка программ на компьютере.
14	Структуры как базовый тип данных и их реализация в языке программирования	Тип структуры struct в языке Си. Определение структуры. Доступ к полям структуры. Вложенные структуры. Битовые поля в структурах. Массивы структур. Структуры и указатели. Примеры использования. Объединения union. Определение. Доступ к полям объединения. Перечислимый тип данных enum. Рекомендации по применению.
15	Динамические структуры данных (структуры, указатели и рекурсивные типы данных)	Структуры и указатели при работе с динамической памятью. Определяемые рекурсивные типы данных. Цепное представление последовательностей. Линейный список с произвольным доступом. Представление в связанной памяти. Программирование основных операций.
16	Линейный список как абстрактный тип данных (АТД)	Способы реализации списка как АТД в языке Си с использованием структур, указателей и рекурсивных типов данных (в парадигме процедурного программирования). Функциональная спецификация Л1-списка. Типовые способы представления и реализации линейных списков: непрерывная реализация на базе массива, ссылочная реализация в динамической памяти.
17	Разновидности списков	Разновидности линейных списков: однонаправленные (Л1-списки), двунаправленные (Л2-списки), циклические списки. Очередь, стек. Реализация на языке Си. Примеры решения задач с линейными списками.

№ п/п	Наименование темы дисциплины	Содержание
18	Битовые операции	Битовые операции. Особенности их выполнения. Примеры функций, моделирующих команды языка Ассемблера. Рекомендации по использованию битовых операций для работы с машинным словом.
19	Рекурсивные и вариационные функции	Понятие рекурсии, рекурсивные алгоритмы. Вызовы рекурсивных функций, примеры. Рекурсия на списке. Функции с переменным числом параметров, варианты организации. Обработка списка параметров. Вариационные функции в стандартной библиотеке.
20	Жизненный цикл и этапы разработки программ. Тестирование программ	Жизненный цикл программы. Анализ требований, проектирование, реализация, сопровождение. Постановка задачи. Формализация постановки задачи. Спецификация задачи (программы). Анализ и описание диалога. Сценарий диалога. Пример спецификации диалоговой программы. Особенности и этапы проектирования. Верификация и аттестация программ. Испытание (тестирование и отладка) программ. Сквозные мероприятия: документирование, подготовка и планирование испытаний. Тестирование программ, использующих динамическую память, структуры и указатели.
21	Заключение	Актуальность языка Си, место Си среди других языков программирования. Задачи, эффективно решаемые с помощью языка Си. Попытки совершенствования и их результаты.

4.2 Перечень лабораторных работ

Наименование лабораторной работы	Количество ауд. часов
1. Условные операторы на языке Си.	3
2. Циклические вычисления на языке Си	6
3. Программирование задач с одномерными массивами.	6
4. Программирование задач с двумерными массивами.	6
5. Указатели и использование динамической памяти.	4
6. Применение функций.	3
7. Обработка строк и файлов.	6
8. Структуры. Массивы структур.	3
9. Указатели на структуры и функции.	3
10. Линейные односвязные списки.	3
11. Линейные двусвязные списки.	3
12. Кольцевые списки.	3
13. Битовые поля в структурах.	2
Итого	51

4.3 Перечень практических занятий

Наименование практических занятий	Количество ауд. часов
1. Основные сведения об интегрированной среде разработки. Требования и порядок выполнения лабораторных работ. Целочисленные типы, типы с плавающей точкой, булевский (логический) тип, символьный тип. Разбор готовых программ.	2
2. Программирование ветвлений (if-else). Простейший ввод-вывод. Разбор примера к лабораторной работе 1.	2
3. Программирование циклов while и for, рекуррентное вычисление суммы ряда, вещественные типы. Разбор примера к лабораторной работе 2.	4
4. Одномерные массивы. Разбор примера к лабораторной работе 3.	2
5. Двумерные массивы. Разбор примера к лабораторной работе 4.	2
6. Указатели и использование динамической памяти. Разбор примера к лабораторной работе 5.	3
7. Функции. Обработка строк.	2
8. Структуры. Массивы структур.	2
9. Указатели на структуры и функции.	2
10. Динамические структуры данных. Линейный список как АД. Типовые действия с линейным однонаправленным списком (Л1-список) и функции для их реализации. Типовые действия с линейным двунаправленным списком (Л2-список).	4
11. Типовые действия с циклическим списком. Стек и очередь на списках.	4
12. Списки и файлы. Вывод в файл информационных полей из элементов списка. Чтение из файла.	2
13. Использование битовых операций для работы с машинным словом.	3
Итого	34

4.4 Курсовое проектирование

Цель работы (проекта): Полное решение содержательной задачи (содержательная и формальная постановка задачи, спецификация, включая описание диалога, выбор метода решения и структур данных, разработка алгоритма, программная реализация, тестирование и отладка, документирование).

Содержание работы (проекта): Семестр 1: обработка текста и текстовой информации, представленной в виде динамических массивов символов (строк), получаемых как из стандартного потока ввода, так и из текстового файла.

Семестр 2: создание и сопровождение электронной картотеки для различных предметных областей, операции с множествами. Используются преимущественно реализации перечисленных структурированных объектов обработки в виде различного рода линейных списков.

Курсовая работа в части таблиц, рисунков и оглавления оформляется в соответствии с ГОСТ 7.32-2017 "Отчет о научно-исследовательской работе. Структура и правила оформления".

Курсовая работа состоит из следующих разделов:

- Введение
- Постановка задачи и описание решения
- Структура вызовов функций
- Описание функций
- Описание переменных (по функциям)
- Схемы алгоритмов
- Контрольные примеры
- Текст программы
- Примеры работы программы (в соответствии с контрольными примерами)
- Заключение

В курсовой работе должно быть использовано не менее 2-х и не более 5-и источников, минимальный объем курсовой работы составляет 25 страниц, максимальный -80 страниц.

Курсовая работа оформляется в редакторе документов в формате MS Word или OpenDocument, шрифт для текста (включая текст программы) должен иметь размер 12-14 пунктов, для надписей в схемах алгоритмов -не менее 10 пунктов. Для основного текста используется шрифт с засечками. для надписей в схемах алгоритмов -шрифт без засечек, для текста программы -моноширинный.

Курсовая работа сдается преподавателю в печатном виде с личной подписью студента на титульном листе...

Темы:

№ п/п	Название темы	Перевод темы
1	Обработка текстовой информации	Processing of text data
2	Разработка электронной картотеки	Development of electronic card file

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, а также прорабатывать примеры решения задач. При этом на основе изучения рекомендованной литературы це-

лесообразно 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	34
Опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	0
Самостоятельное изучение разделов дисциплины	0
Выполнение домашних заданий, домашних контрольных работ	0
Подготовка к лабораторным работам, к практическим и семинарским занятиям	51
Подготовка к контрольным работам, коллоквиумам	0
Выполнение расчетно-графических работ	0
Выполнение курсового проекта или курсовой работы	82
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
ИТОГО СРС	237

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	Павловская, Татьяна Александровна. С/С++. Программирование на языке высокого уровня [Текст] : Учеб. для вузов по направлению "Информатика и вычислительная техника" / Т. А. Павловская, 2001. -460 с.	10
2	Разумовский, Геннадий Васильевич. Основы программирования на языке СИ [Текст] : учеб.-метод. пособие / Г. В. Разумовский, И. А. Хахаев, 2019. -31, [1] с.	55
3	Разумовский, Геннадий Васильевич. Обработка структур и списков на языке СИ [Текст] : учеб.-метод. пособие / Г. В. Разумовский, И. А. Хахаев, 2020. -46, [2] с.	60
Дополнительная литература		
1	Подбельский, Вадим Валериевич. Программирование на языке Си [Текст] : [учеб. пособие для вузов по направлениям: "Прикладная математика и информатика", "Информатика и вычисл. техника", специальностям "Прикладная математика", "Вычисл. машины, комплексы, системы и сети управления"] / В.В.Подбельский, С.С.Фомин, 1999. -600 с.	13
2	Керниган, Брайн В. Язык программирования Си [Текст] : монография / Б.В. Керниган; Пер. с англ. В.С.Штаркмана; Под ред. и с предисл. В.С.Штаркмана, 1992. -271 с.	45

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

№ п/п	Электронный адрес
1	Язык программирования Chttp://www.intuit.ru/studies/courses/97/97/info
2	Курс Лекций по Языку Сиhttps://learn.info/c/

5.3 Адрес сайта курса

Адрес сайта курса: <https://vec.etu.ru/moodle/course/view.php?id=14177>

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 – 59	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	60 -74	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	75 -89	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	90 -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

Условие допуска к курсовой работе: выполнение и защита не менее 3 лабораторных работ.

Условие допуска к экзамену: выполнение лабораторных работ и защита курсовой работы.

Экзамен проводится в виде практической и теоретической части. Практическая часть включает в себя решение задачи в интегрированной среде разработки. Теоретическая часть включает в себя электронный тест. В обоих семестрах условия допуска и порядок проведения экзамена совпадают.

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

Вопросы к экзамену

№ п/п	Описание
1	Понятие алгоритма
2	Массивы указателей
3	Варианты функций (классификация)
4	Особенности строк в Си
5	Группы функций для работы с файлами (потоками)
6	Размещение структур в памяти
7	Динамический массив указателей на функции
8	Операции с линейным односвязным списком
9	Особенности кольцевых списков
10	Применение битовых операций при обработке символьных данных
11	Последовательные (линейные) вычисления
12	Циклические вычисления — виды
13	Бесконечный цикл и досрочный выход из цикла
14	Эквивалентность for() и while()
15	Одномерные массивы в Си
16	Операции с указателями (адресная арифметика)
17	Реализация структурного (модульного) подхода к разработке программ
18	Варианты функций (классификация)
19	Оператор return
20	Указатель как параметр функции
21	Функции на схемах алгоритмов
22	Пример функции: обработка введенных символов

23	Инициализация и вывод строк
24	Функции для работы со строками (базовый набор)
25	Строки в динамической памяти
26	Группы функций для работы с файлами (потоками)
27	Файлы: чтение и запись символов
28	Файлы: форматный ввод и вывод
29	Позиционирование в файле
30	Блочный ввод и вывод
31	Определение размера массива строк
32	Операции со структурами
33	Размещение структур в памяти
34	Структуры и файлы
35	Динамические массивы в полях структуры
36	Структура как параметр функции
37	Перечисляемый тип (enum)
38	Динамические массивы структур
39	Указатели на структуры
40	Динамический массив указателей на структуры
41	Динамический массив указателей на функции
42	Динамические информационные структуры
43	Линейный однонаправленный (односвязный) список
44	Типы информационных полей элемента списка
45	Операции с линейным односвязным списком
46	Добавление последнего элемента списка
47	Поиск элемента по номеру (позиции)
48	Подсчет количества элементов списка
49	Двусвязный (двунаправленный) список
50	Вставка элемента L2-списка
51	Особенности кольцевых списков
52	Очистка стека, поиск и удаление элементов
53	Реализация функции просмотра элемента в начале очереди
54	Применение битовых операций при обработке символьных данных
55	Допустимость неизвестного набора параметров функций в Си

Вариант экзаменационного теста

Выберите описание свойства **результативности** алгоритма.

Выберите один ответ:

- Любые входы и выходы должны быть описаны в виде структур или типов данных. Любые интуитивно понятные функции должны быть формально описаны в терминах элементов алгоритма.

- Алгоритм для исполнителя должен включать только те команды, которые ему (исполнителю) доступны, которые входят в его систему команд.
- Алгоритм должен представлять процесс решения задачи как последовательное выполнение некоторых простых шагов. При этом для выполнения каждого шага алгоритма требуется конечный отрезок времени.
- Интуитивно алгоритм предполагает реализацию функций и только их, так что заданному набору входных данных отвечает только один результат.
- При корректно заданных исходных данных алгоритм должен завершать работу и выдавать результат за конечное число шагов.
- Должно обеспечиваться определение и распознавание начального (правильные входные данные) и конечного (правильные выходные данные) события, связанного с правильным или неправильным результатом работы алгоритма.
- Алгоритм должен быть применим к разным наборам исходных данных.

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
2	Основные управляющие структуры программирования и управляющие инструкции в языке Си	
3		
4		Отчет по лаб. работе
5	Основные управляющие структуры программирования и управляющие инструкции в языке Си	
6		Отчет по лаб. работе
7	Массивы и указатели	
8		Отчет по лаб. работе
9	Разработка программ при работе с массивами и указателями	
10		Отчет по лаб. работе
11	Основные управляющие структуры программирования и управляющие инструкции в языке Си Размещение данных в памяти. Явное управление памятью. Функции и структура программы	
12		Отчет по лаб. работе
13		
14		Отчет по лаб. работе
15	Обработка текстов с использованием структурированных типов данных	
16		Отчет по лаб. работе
17	Тестирование и отладка программ	Защита КР / КП
18	Структуры как базовый тип данных и их реализация в языке программирования	
19		
20		Отчет по лаб. работе
21	Динамические структуры данных (структуры, указатели и рекурсивные типы данных)	
22		
23		Отчет по лаб. работе
24	Линейный список как абстрактный тип данных (АТД)	
25		
26		Отчет по лаб. работе
27	Разновидности списков	
28		Отчет по лаб. работе
29	Разновидности списков	
30		
31		Отчет по лаб. работе
32	Битовые операции	
33		Отчет по лаб. работе
34	Жизненный цикл и этапы разработки программ. Тестирование программ	Защита КР / КП

6.4 Методика текущего контроля

Лекционные занятия

Текущий контроль усвоения материала лекционных занятий проводится

в форме тестирования в середине семестра по пройденному материалу (электронное тестирование в среде Moodle).

Лабораторные работы

При выполнении лабораторной работы студент должен выполнить постановку задачи в соответствии с вариантом задания, а также спроектировать программу. В постановке задачи должны быть указаны все условия и ограничения, при которых должна работать программа, приведены все соотношения и формулы. При проектировании программы создается схема алгоритма по ГОСТ 19.701-90 "Схемы алгоритмов программ, данных и систем" с учетом ограничений, связанных с эргономикой схем алгоритмов. Также студент должен помимо алгоритма и собственно программы разработать контрольные примеры, позволяющие определить правильность работы программы.

При проверке лабораторной работы учитываются следующие факторы:

- однозначность и полнота постановки задачи;
- соответствие постановки задачи варианту задания;
- соответствие схемы алгоритма поставленной задаче, требованиям ГОСТ 19.701-90 "Схемы алгоритмов программ, данных и систем" и эргономики схем;
- соответствие контрольных примеров решаемой задаче, полнота набора контрольных примеров;
- оформление кода в соответствии с одним из общепринятых стилей (linux kernel style, GNU coding style);
- оформление отчета по лабораторной работе в соответствии с шаблоном СПбГЭТУ «ЛЭТИ»;
- соответствие выводов по лабораторной работе цели работы и результатам выполнения задания;
- наличие/отсутствие предупреждений (предупреждения означают, что в программе есть неиспользуемые переменные или неверно описаны ти-

пы/форматы данных).

Лабораторная работа не принимается в следующих случаях

- постановка задачи отсутствует или не соответствует варианту задания;
- схема алгоритма не соответствует заданию и/или требованиям ГОСТ 19.701-90 "Схемы алгоритмов программ, данных и систем";
- отчет оформлен не по шаблону, отсутствуют обязательные элементы отчета;
- контрольные примеры не соответствуют поставленной задаче или не позволяют проверить все варианты работы программы;
- выводы по лабораторной работе не соответствуют цели работы и результатам выполнения задания;
- программа не компилируется;
- при запуске программы с допустимым набором данных имеются ошибки времени выполнения (аварийное завершение программы).

Практические занятия

В ходе практических занятий студенты решают задачи с автоматической проверкой в среде Moodle. Каждое практическое занятие состоит из 5 задач, которые могут быть решены в течение трёх недель. Для каждого практического занятия дается две попытки решения случайного набора задач по теме занятия (задачи в разных попытках не совпадают). Максимальный балл за каждую задачу - 1, за практическое занятие - 5. Из двух попыток засчитывается попытка с лучшим результатом. Баллы за практическое занятия учитываются в общем балле студента по итогам семестра.

Курсовая работа

При выполнении курсовой работы студент должен выполнить постановку задачи в соответствии с вариантом задания, спроектировать программу, а также обеспечить минимальный пользовательский интерфейс и защиту от ошибочно-

го ввода. В постановке задачи должны быть указаны все условия и ограничения, при которых должна работать программа, приведены все соотношения и формулы. При проектировании программы создается схема алгоритма по ГОСТ 19.701-90 "Схемы алгоритмов программ, данных и систем" с учетом ограничений, связанных с эргономикой схем алгоритмов. Также студент должен помимо алгоритма и собственно программы разработать контрольные примеры, позволяющие определить правильность работы программы.

При проверке курсовой работы учитываются следующие факторы

- однозначность и полнота постановки задачи
- соответствие постановки задачи варианту задания
- соответствие схемы алгоритма поставленной задаче, требованиям ГОСТ 19.701-90 "Схемы алгоритмов программ, данных и систем" и эргономики схем (структурированные схемы)
- соответствие контрольных примеров решаемой задаче, полнота набора контрольных примеров
- оформление кода в соответствии с одним из общепринятых стилей (linux kernel style, GNU coding style)
- оформление отчета по курсовой работе в соответствии с шаблоном СПбГ-ЭТУ «ЛЭТИ»
- наличие и полнота описания структур данных, функций и их переменных
- соответствие выводов по курсовой работе цели работы и результатам выполнения задания
- понятность интерфейса программы и наличие защиты от ошибочного ввода (в том числе в виде сообщений в интерфейсе программы)
- наличие/отсутствие предупреждений при компиляции программы (предупреждения означают, что в программе есть неиспользуемые переменные или неверно описаны типы/форматы данных).

Курсовая работа оценивается на «неудовлетворительно» в следующих случаях

- постановка задачи отсутствует или не соответствует варианту задания
- схема алгоритма не соответствует заданию и/или требованиям ГОСТ 19.701-90 "Схемы алгоритмов программ, данных и систем"
- отчет оформлен не по шаблону, отсутствуют обязательные элементы отчета
- контрольные примеры не соответствуют поставленной задаче или не позволяют проверить все варианты работы программы
- выводы по курсовой работе не соответствуют цели работы и результатам выполнения задания
- программа не компилируется
- при запуске программы с допустимым набором данных имеются ошибки времени выполнения (аварийное завершение программы).

Курсовая работа оценивается на «удовлетворительно», если контрольные примеры и схемы алгоритмов соответствуют поставленной задаче, ошибок при компиляции не возникает, ошибок времени выполнения не возникает и отчет оформлен с соответствии с шаблоном.

Курсовая работа оценивается на «отлично», если выполнены все требования к курсовой работе и обеспечена защита от ошибочного ввода не только в виде сообщений в интерфейсе программы, но и на уровне кода.

Курсовая работа оценивается на «хорошо» при наличии одного из следующих недостатков

- схемы алгоритмов не соответствуют требованиям эргономики (не структурированы)
- интерфейс программы не очевиден (не проработан)
- при компиляции программы имеются предупреждения.

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, наличие проектора и экрана, а также ноутбука (ПК) для преподавателя	ОС «АЛЫТ Образование 8.2.» и выше; компилятор Си стандартов C89, C99, C11; Среда разработки Code::Blocks IDE
Лабораторные работы	Лаборатория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, наличие ПК на рабочих местах	ОС Windows 7 и выше или ОС «АЛЫТ Образование 8.2.» и выше; компилятор Си стандартов C89, C99, C11; Среда разработки Code::Blocks IDE
Практические занятия	Аудитория	Количество посадочных мест – в соответствии с контингентом, рабочее место преподавателя, наличие ПК на рабочих местах	ОС Windows 7 и выше или ОС «АЛЫТ Образование 8.2.» и выше; компилятор Си стандартов C89, C99, C11; Среда разработки Code::Blocks IDE
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 или MacOS / Linux 2) Microsoft Office 2007 и выше или LibreOffice

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