

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Национальный исследовательский университет «МЭИ»

Направление подготовки/специальность: 13.03.02 Электроэнергетика и электротехника

Наименование образовательной программы: Электроэнергетика и электротехника

Уровень образования: высшее образование - бакалавриат

Форма обучения: Очная

Рабочая программа дисциплины
ПРОГРАММИРОВАНИЕ И ОСНОВЫ АЛГОРИТМИЗАЦИИ

Блок:	Блок 1 «Дисциплины (модули)»
Часть образовательной программы:	Обязательная
№ дисциплины по учебному плану:	Б1.О.16
Трудоемкость в зачетных единицах:	3 семестр - 3; 4 семестр - 3; всего - 6
Часов (всего) по учебному плану:	216 часов
Лекции	3 семестр - 16 часов; 4 семестр - 16 часов; всего - 32 часа
Практические занятия	3 семестр - 16 часов; 4 семестр - 16 часов; всего - 32 часа
Лабораторные работы	3 семестр - 16 часов; 4 семестр - 16 часов; всего - 32 часа
Консультации	4 семестр - 2 часа;
Самостоятельная работа	3 семестр - 59,7 часа; 4 семестр - 57,5 часа; всего - 117,2 часов
в том числе на КП/КР	не предусмотрено учебным планом
Иная контактная работа	проводится в рамках часов аудиторных занятий
включая: Тестирование Контрольная работа Лабораторная работа Расчетно-графическая работа	
Промежуточная аттестация:	
Зачет с оценкой	3 семестр - 0,3 часа;
Экзамен	4 семестр - 0,5 часа; всего - 0,8 часа

Москва 2025

ПРОГРАММУ СОСТАВИЛ:

Преподаватель

	Подписано электронной подписью ФГБОУ ВО «НИУ «МЭИ»	
	Сведения о владельце ЦЭП МЭИ	
	Владелец	Воробьева И.А.
	Идентификатор	R86e9a563-VorobyevaIA-80eec2d

И.А. Воробьева

СОГЛАСОВАНО:

Руководитель
образовательной программы

	Подписано электронной подписью ФГБОУ ВО «НИУ «МЭИ»	
	Сведения о владельце ЦЭП МЭИ	
	Владелец	Тульский В.Н.
	Идентификатор	R292b173d-TulskyVN-7e812984

В.Н. Тульский

Заведующий выпускающей
кафедрой

	Подписано электронной подписью ФГБОУ ВО «НИУ «МЭИ»	
	Сведения о владельце ЦЭП МЭИ	
	Владелец	Тульский В.Н.
	Идентификатор	R292b173d-TulskyVN-7e812984

В.Н. Тульский

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цель освоения дисциплины: состоит в изучении теоретических, практических основ и технологий программирования, теории алгоритмов, анализа и алгоритмизации решений прикладных задач.

Задачи дисциплины

- овладение основами программирования на языках с разными концептуальными подходами;
- изучение базовых аспектов реализаций языков программирования;
- изучение методов построения и анализа алгоритмов;
- освоение способов теоретического и практического сравнения алгоритмов;
- приобретение навыков самостоятельного проектирования алгоритмов и использования прикладных библиотек.

Формируемые у обучающегося **компетенции** и запланированные **результаты обучения** по дисциплине, соотнесенные с **индикаторами достижения компетенций**:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Запланированные результаты обучения
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	ИД-1 _{УК-1} Выполняет поиск необходимой информации, ее критический анализ и обобщает результаты анализа для решения поставленной задачи	знать: - численные и алгоритмические методы решения типовых задач поиска, сортировки, фильтрации, кодирования, приближенных решений математических задач; - аналитические и практические способы анализа сложности и эффективности алгоритмов. уметь: - составлять алгоритмы решения задач по словесному описанию методов, производить сравнительный анализ алгоритмов по различным оценкам качества аналитически и на реальных тестах; - адаптировать, модифицировать и комбинировать существующие алгоритмы для решения задач поиска, сортировки, фильтрации данных, кодирования и преобразования данных.
ОПК-1 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ИД-2 _{ОПК-1} Применяет средства информационных, компьютерных и сетевых технологий для поиска, хранения, обработки, анализа и представления информации	знать: - основные принципы хранения, доступа и обработки данных в динамических структурах, базах данных, при поточном и многопоточном программировании; - встроенные и внешние библиотеки языков. уметь: - использовать возможности встроенных и внешних библиотек, а

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Запланированные результаты обучения
		также графические библиотеки для визуализации решений; - программировать собственные классы для динамических объектов на основании принятых базовых операций в заданных структурах данных с применением их в прикладных задачах.
ОПК-2 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ИД-1 _{ОПК-2} Алгоритмизирует решение задачи и реализует алгоритмы с помощью программных средств	<p>знать:</p> <ul style="list-style-type: none"> - принципы и особенности программирования в языках со статической и динамической типизацией; - понятие рекуррентных соотношений и методы рекурсии; - основные концепции программирования. <p>уметь:</p> <ul style="list-style-type: none"> - программировать решения типовых задач в языках с отличающейся семантикой; - решать задачи с применением рекуррентного и рекурсивного подхода, использовать динамические структуры данных; - применять языковые возможности поточной обработки данных для типичных задач.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ ВО

Дисциплина относится к основной профессиональной образовательной программе Электроэнергетика и электротехника (далее – ОПОП), направления подготовки 13.03.02 Электроэнергетика и электротехника, уровень образования: высшее образование - бакалавриат.

Требования к входным знаниям и умениям:

- знать представление информации в электронно-вычислительных устройствах, их структуру и основные характеристики
- знать базовые численные и алгоритмические методы решения типовых задач
- знать основы программирования на языках высокого уровня (Pascal, C, Python и др.)
- уметь модифицировать и комбинировать базовые алгоритмические подходы, для решения конкретных задач
- уметь применять методологию нисходящего проектирования в решении сложных многоуровневых задач

Результаты обучения, полученные при освоении дисциплины, необходимы при выполнении выпускной квалификационной работы.

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

3.1 Структура дисциплины

Общая трудоемкость дисциплины составляет 6 зачетных единиц, 216 часов.

№ п/п	Разделы/темы дисциплины/формы промежуточной аттестации	Всего часов на раздел	Семестр	Распределение трудоемкости раздела (в часах) по видам учебной работы										Содержание самостоятельной работы/ методические указания
				Контактная работа							СР			
				Лек	Лаб	Пр	Консультация		ИКР		ПА	Работа в семестре	Подготовка к аттестации /контроль	
КПР	ГК	ИККП	ТК											
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	Основные аспекты реализации Python и C(C++)	26	3	8	4	3	-	-	-	-	-	11	-	<p><u>Подготовка к текущему контролю:</u> Повторение материала по разделу "Основные аспекты реализации Python и C(C++)"</p> <p><u>Подготовка к текущему контролю:</u> Подготовка к тесту "Области видимости, пространства имен, время жизни переменных" направлена на подтверждение знаний материала, полученного в разделе "Основные аспекты реализации Python и C(C++)" по части темы "Сравнение реализаций Python и C(C++)". Дополнительно студенту необходимо изучить литературу и разобрать примеры, приводимые на практических занятиях</p> <p><u>Подготовка к контрольной работе:</u> Изучение материалов по разделу Основные аспекты реализации Python и C(C++) и подготовка к контрольной работе "Реализация основных алгоритмов, организация ввода и вывода в C и C++" направлена на подтверждение знаний материала, полученного по теме "Введение в язык C(C++)". Дополнительно студенту необходимо изучить литературу и разобрать примеры, приводимые на практических занятиях</p> <p><u>Изучение материалов литературных</u></p>
1.1	Введение в язык C(C++). Сравнение реализаций Python и C(C++)	26		8	4	3	-	-	-	-	-	-	11	

													<u>источников:</u> [3], с. 18-29 [6], с. 39-43, 57-59, 70-90, 95-105, 132-149, 453-460
2	Основы анализа алгоритмов	22	4	5	4	-	-	-	-	-	9	-	<u>Подготовка к лабораторной работе:</u> Выполнение и подготовка к защите лабораторной работы "Реализация стандарта NCR_R (русификация) на двух языках" направлено на отработку умений решения профессиональных задач. Выдается студентам по изученному в разделе "Основы анализа алгоритмов" материалу по теме: "Методы поиска, сортировки, кодирования и декодирования". Дополнительно студенту необходимо изучить литературу и разобрать примеры выполнения заданий, рассмотренные на практических занятиях
2.1	Введение. Методы поиска, сортировки, кодирования и декодирования	22	4	5	4	-	-	-	-	-	9	-	<u>Подготовка к лабораторной работе:</u> Выполнение и подготовка к защите лабораторных работ "Реализация алгоритма пузырьковой сортировки в двух направлениях" и "Азбука Морзе. Разработка алгоритма кодирования и декодирования на двух языках" направлено на отработку умений решения профессиональных задач. Выдается студентам по изученному в разделе "Основы анализа алгоритмов" материалу по части темы: "Методы поиска, сортировки, кодирования и декодирования". Дополнительно студенту необходимо изучить литературу и разобрать примеры выполнения заданий, рассмотренные на практических занятиях <u>Подготовка к лабораторной работе:</u> Выполнение и подготовка к защите лабораторных работ "Бинарный поиск. Поиск подстроки в строке. Предложение модификаций" и "Реализация и анализ алгоритмов сортировки выбором и вставкой"

													направлено на закрепление знаний и отработку умений решения профессиональных задач. Выдается студентам по изученному в разделе "Основы анализа алгоритмов" материалу по темам: "Введение" и части темы "Методы поиска, сортировки, кодирования и декодирования". Дополнительно студенту необходимо изучить литературу и разобрать примеры выполнения заданий, рассмотренные на практических занятиях <u>Подготовка к текущему контролю:</u> Повторение материала по разделу "Основы анализа алгоритмов" <u>Изучение материалов литературных источников:</u> [2], с. 268-272 [7], с. 45-51, 155, 383-392, 496-505
3	Исследование и проектирование алгоритмов	42	4	7	9	-	-	-	-	-	22	-	<u>Подготовка расчетных заданий:</u> Для выполнения задания "Внешние библиотеки в Python. Графика в Python. Алгоритм имитации нормальных распределений объектов на плоскости с помощью равномерных распределений и их графическое отображение" необходимо предварительно изучить тему "Проектирование алгоритмов. Файлы, внешние библиотеки и графика в Python". Задание выполняется индивидуально по общей схеме алгоритма, изложенного в практических занятиях. Индивидуализация заключается в моделировании собственных распределенных объектов, анализе тестов
3.1	Исследование алгоритмов. Внутренние библиотеки в Python	18	2	3	4	-	-	-	-	-	9	-	<u>Подготовка к текущему контролю:</u> Повторение материала по разделу "Исследование и проектирование алгоритмов" <u>Подготовка к лабораторной работе:</u> Для выполнения заданий по лабораторной работе
3.2	Проектирование алгоритмов. Внешние библиотеки и графика в Python	24	2	4	5	-	-	-	-	-	13	-	

														<p>"Усовершенствованная пузырьковая сортировка. Экспериментальное исследование сортировок. Разработка и реализация экспериментальных тестов" необходимо предварительно изучить тему и задачи выполнения лабораторной работы по изученному в разделе "Исследование и проектирование алгоритмов" материалу по теме "Исследование алгоритмов. Внутренние библиотеки в Python"</p> <p><u>Изучение материалов литературных источников:</u></p> <p>[4], с. 251-278 [5], с. 35-50, 136-144, 278-288, 294-300</p>
	Зачет с оценкой	18.0		-	-	-	-	-	-	-	0.3	-	17.7	
	Всего за семестр	108.0		16	16	16	-	-	-	-	0.3	42	17.7	
	Итого за семестр	108.0		16	16	16	-	-	-	-	0.3	59.7		
4	Взаимодействие программ с вызывающим контекстом	24	4	4	6	5	-	-	-	-	-	9	-	<p><u>Подготовка к контрольной работе:</u> Изучение материалов по теме "Рекурсия. Методы уточнения корней" раздела "Взаимодействие программ с вызывающим контекстом" и подготовка к контрольной работе "Рекуррентные и рекурсивные алгоритмы" направлена на подтверждение знаний материала. Дополнительно студенту необходимо изучить литературу и разобрать примеры, приводимые на практических занятиях</p> <p><u>Подготовка к лабораторной работе:</u> Для выполнения заданий по лабораторной работе «Динамические массивы в качестве параметров подпрограмм в С(С++)» необходимо предварительно изучить тему "Взаимодействие программ с вызывающим контекстом С(С++)" раздела "Взаимодействие программ с вызывающим контекстом" и задачи рассмотренные на практических занятиях</p> <p><u>Подготовка к лабораторной работе:</u></p>
4.1	Взаимодействие программ с вызывающим контекстом С(С++). Рекурсия	24		4	6	5	-	-	-	-	-	9	-	

														<p>Выполнение и подготовка к защите лабораторной работы "Методы приближения корней. Указатель на функцию в C(C++)" направлено на отработку знаний и умений решения профессиональных задач. Выдается студентам по изученному в разделе "Взаимодействие программ с вызывающим контекстом" материалу. Дополнительно студенту необходимо изучить материал по части темы "Рекурсия. Методы уточнения корней" и разобрать примеры выполнения заданий, рассмотренные на практических занятиях</p> <p><u>Подготовка к лабораторной работе:</u> Для выполнения заданий по лабораторной работе "Динамические массивы и рекурсивные алгоритмы в C(C++)" необходимо предварительно изучить часть темы "Рекурсия. Методы уточнения корней" раздела "Взаимодействие программ с вызывающим контекстом" и разобрать примеры выполнения заданий, рассмотренные на практических занятиях</p> <p><u>Подготовка к текущему контролю:</u> Повторение материала по разделу "Взаимодействие программ с вызывающим контекстом"</p> <p><u>Изучение материалов литературных источников:</u> [1], с. 32-33, 119-125 [6], с. 158-165, 177-186, 194-198 [7], с. 272-281 [8], с. 30-42</p>
5	Особенности динамической типизации. Способы поточной обработки данных	18	4	4	4	-	-	-	-	-	6	-	<p><u>Подготовка к лабораторной работе:</u> Для выполнения заданий по лабораторной работе "Элементы функционального программирования в Python" необходимо предварительно изучить тему и задачи выполнения лабораторной работы, а так же</p>	
5.1	Особенности	18	4	4	4	-	-	-	-	-	6	-		

	динамической типизации. Способы поточной обработки данных												изучить вопросы вариантов обработки результатов по изученному в разделе "Особенности динамической типизации. Способы поточной обработки данных" материалу <u>Подготовка к текущему контролю:</u> Повторение материала по разделу "Особенности динамической типизации. Способы поточной обработки данных" <u>Изучение материалов литературных источников:</u> [5], с. 67-91
6	Дополнительные возможности программирования в Python	30	8	6	7	-	-	-	-	-	9	-	<u>Подготовка к текущему контролю:</u> Подготовка к тесту "Базы данных. Инструменты многопоточного программирования" направлена на подтверждение знаний материала, полученного в разделе "Дополнительные возможности программирования в Python" по части темы "Двоичные файлы. Базы данных. Многопоточное программирование". Дополнительно студенту необходимо разобрать примеры, приводимые на практических занятиях
6.1	Структуры данных. Обзор методов построения и анализа алгоритмов	18	5	3	5	-	-	-	-	-	5	-	Подтверждение знаний материала, полученного в разделе "Дополнительные возможности программирования в Python" по части темы "Двоичные файлы. Базы данных. Многопоточное программирование". Дополнительно студенту необходимо разобрать примеры, приводимые на практических занятиях
6.2	Введение в объектно-ориентированное программирование (ООП) в Python	12	3	3	2	-	-	-	-	-	4	-	<u>Подготовка к текущему контролю:</u> Повторение материала по разделу "Дополнительные возможности программирования в Python" <u>Подготовка к текущему контролю:</u> Подготовка к тесту "Динамические структуры данных" направлена на подтверждение знаний материала, полученного в разделе "Дополнительные возможности программирования в Python" по части темы "Динамические структуры данных. Обзор методов построения и анализа алгоритмов". Дополнительно студенту необходимо разобрать примеры, приводимые на практических занятиях

														<p><u>Подготовка расчетных заданий:</u> Задания ориентированы на решения бригадных задач по теме "Введение в объектно-ориентированное программирование (ООП) в Python" раздела "Дополнительные возможности программирования в Python". Студентам необходимо повторить теоретический материал, разобрать пример решения аналогичной задачи, изложенной в практическом занятии, а также изучить дополнительную литературу. В качестве тем задания применяются следующие: "Реализация класса стек", "Реализация класса бинарное дерево" и др.</p> <p><u>Изучение материалов литературных источников:</u> [2], с. 90-100, 232-248 [3], с. 82-92, 180-200 [5], с. 177-180, 241-277 [6], с. 26-38, 90-93 [7], с. 122-129, 141-150, 224-228, 875-880, 1055-1064, 1180-1187, 1231-1243</p>
	Экзамен	36.0	-	-	-	-	2	-	-	0.5	-	33.5		
	Всего за семестр	108.0	16	16	16	-	2	-	-	0.5	24	33.5		
	Итого за семестр	108.0	16	16	16	2	-	-	-	0.5	57.5			
	ИТОГО	216.0	-	32	32	32	2	-	-	0.8	117.2			

Примечание: Лек – лекции; Лаб – лабораторные работы; Пр – практические занятия; КПр – аудиторные консультации по курсовым проектам/работам; ИККП – индивидуальные консультации по курсовым проектам/работам; ГК- групповые консультации по разделам дисциплины; СР – самостоятельная работа студента; ИКР – иная контактная работа; ТК – текущий контроль; ПА – промежуточная аттестация

3.2 Краткое содержание разделов

1. Основные аспекты реализаций Python и C(C++)

1.1. Введение в язык C(C++). Сравнение реализаций Python и C(C++)

Введение в язык C(C++). Типы данных, операции и приоритеты. Объявление и определение, суть понятий. Объявления констант, переменных. Инициализация, тип void, переопределение типов. Ассоциативность, семантика выполнения логических и тернарных операций. Структура программы. Ввод и вывод данных в C и C++. Спецификаторы формата и манипуляторы. Методы считывания символьной информации из потоков. Указатели, арифметика указателей, константные и void указатели. Динамическое распределение памяти в C и C++. Массивы одномерные, многомерные, статические и динамические. Ссылки, понятие ссылки на константу. Подпрограммы, взаимодействие программ с вызывающим контекстом. Сравнение реализаций Python и C(C++). Обзор парадигм программирования, поддерживаемых Python, влияние других языков на Python, применение Python. Структура программы. Область видимости и пространство имен. Операторы global, nonlocal. Способы подключения модулей в C(C++) и импортирования модулей в Python.

2. Основы анализа алгоритмов

2.1. Введение. Методы поиска, сортировки, кодирования и декодирования

Понятия сложности задачи, вычислительной сложности алгоритма и оценки сложности. Способы оценки и сравнения алгоритмов: аналитические, практические. Методы поиска: бинарный и линейный поиск в дискретных и непрерывных последовательностях. Методы сортировки: метод вставки, метод пузырька, сортировка слиянием. Понятие внутренней и внешней сортировки. Оценки сложности алгоритмов. Алгоритмы перекодирования. Специфика символьной кодировки. Алгоритм генерации случайного пароля. Проектирование алгоритма перекодировки азбуки Морзе и работа с текстовыми файлами..

3. Исследование и проектирование алгоритмов

3.1. Исследование алгоритмов. Внутренние библиотеки в Python

Исследование и сравнительный анализ алгоритмов на базе алгоритмов поиска, сортировок. Модуль time и точность временных характеристик, модуль random и генератор псевдослучайных чисел.

3.2. Проектирование алгоритмов. Внешние библиотеки и графика в Python

Алгоритм генерации нормально-распределенных величин. Подключение внешних библиотек. Проектирование алгоритма имитации распределенных объектов на ограниченном пространстве и их графическое отображение.

4. Взаимодействие программ с вызывающим контекстом

4.1. Взаимодействие программ с вызывающим контекстом C(C++). Рекурсия

Способы передачи параметров в функцию: аргумент-значение, аргумент-функция, аргумент-ссылка. Основные способы изменения переменной в функции. Передача встроенных массивов в функции. Ссылка на константный объект в параметрах функции. Указатель на функцию, указатель на функцию в качестве аргумента другой функции. Рекурсия: основные понятия и определения. Связь рекурсии и рекуррентных вычислений. Виды рекурсий: каскадная и линейная, простая и косвенная. Рекурсивные алгоритмы. Примеры реализации в C и Python (задача динамического программирования). Численные методы приближения корней: Ньютона, секущих, хорд.

5. Особенности динамической типизации. Способы поточной обработки данных

5.1. Особенности динамической типизации. Способы поточной обработки данных

Динамическая типизация данных. Подсчет ссылок и сборка мусора. Изменяемые и неизменяемые данные. Сравнение переменных в C и Python. Связь структуры в C и PyObject в Python. Интернированные объекты. Вызов по соиспользованию и взаимодействие подпрограмм с вызывающим контекстом (сравнение Python с C и C++). Элементы функционального программирования в Python. Лямбда функции. Списковые сборки, генераторы коллекций. Генераторы и создание матриц. Генераторные выражения. Итерируемые объекты, объекты-итераторы и объекты-генераторы. Функции генераторы.

6. Дополнительные возможности программирования в Python

6.1. Структуры данных. Обзор методов построения и анализа алгоритмов

Основные динамические структуры: список, очередь, стек, дек, дерево. Реализация типа список в CPython, создание собственных классов в Python. Хеш-функции, хеш-таблицы, прямая адресация. Динамическое программирование. Жадные алгоритмы. Амортизационный анализ. Графы и основные алгоритмы на графах. Поиск в глубину и в ширину. Работа с бинарными файлами. Работа с базами данных. Введение в многопоточное потоковое программирование.

6.2. Введение в объектно-ориентированное программирование (ООП) в Python

Концепция ООП. Определение класса. Объекты-классы, объекты-экземпляры. Методы экземпляров классов. Сравнение реализаций в C++ и Python.

3.3. Темы практических занятий

1. Элементы функционального программирования. Лямбда-выражения. Генераторы коллекций. Генераторные выражения. Генераторы и итераторы;
2. Передача функций в качестве параметров другой функции. Сравнение реализаций C(C++) и Python;
3. Статическая и динамическая типизация. Изменяемые и неизменяемые типы данных. Сравнение реализаций данных в C(C++) и Python;
4. Рекурсия, рекуррентные соотношения, рекурсивные алгоритмы. Реализация в C(C++) и Python;
5. Взаимодействие подпрограмм с вызывающим контекстом. Об особенностях передачи встроенных массивов в C(C++);
6. Алгоритм имитации нормальных распределений объектов на плоскости с помощью равномерных распределений точек;
7. Экспериментальные способы исследования алгоритмов на примере исследования алгоритма пузырьковой сортировки и его модификации. Освоение модулей time и random, особенности генератора псевдослучайных чисел (ГПСЧ) в Python;
8. Методы поиска и сортировки. Алгоритмы бинарного поиска, поиска подстрок в строке, алгоритмы внутренних и внешних сортировок. Оценки сложности и сравнение рассмотренных алгоритмов, варианты модификаций;
9. Динамические структуры данных. Реализация собственного класса в Python;
10. Разработка собственной базы данных (БД). БД SQLite и модуль сопряжения для БД. Формирование SQL-запросов. Использование файлов в формате CSV для наполнения БД;
11. Алгоритмы кодирования и перекодирования на примерах работы с символьной информацией. Работа с символами в C и потоками ввода и вывода в C++;
12. Особенности работы с подпрограммами в C(C++) в сравнении с Python;

13. Подключение внешних библиотек в Python на примере изучения графической библиотеки `graphics`;
14. Работа с текстовыми и бинарными файлами. Оператор `with`. Модуль `pickle`. Текстовый формат `CSV`.

3.4. Темы лабораторных работ

1. «Азбука Морзе. Разработка алгоритма кодирования и декодирования на двух языках»;
2. «Бинарный поиск. Поиск подстроки в строке. Предложение модификаций»;
3. «Реализация и анализ алгоритмов сортировки выбором и вставкой»;
4. «Реализация стандарта `NCR_R` (русификация) на двух языках»;
5. «Реализация алгоритма пузырьковой сортировки в двух направлениях»;
6. «Усовершенствованная пузырьковая сортировка. Экспериментальное исследование сортировок. Разработка и реализация экспериментальных тестов»;
7. «Динамические массивы в качестве параметров подпрограмм в `C(C++)`»;
8. «Динамические массивы и рекурсивные алгоритмы в `C(C++)`»;
9. «Элементы функционального программирования в Python»;
10. «Методы приближения корней. Указатель на функцию в `C(C++)`».

3.5 Консультации

Групповые консультации по разделам дисциплины (ГК)

1. Обсуждение материалов по кейсам раздела "Основные аспекты реализаций Python и `C(C++)`"
2. Обсуждение материалов по кейсам раздела "Основы анализа алгоритмов"
3. Обсуждение материалов по кейсам раздела "Исследование и проектирование алгоритмов"
4. Обсуждение материалов по кейсам раздела "Взаимодействие программ с вызывающим контекстом"
5. Обсуждение материалов по кейсам раздела "Особенности динамической типизации. Способы поточной обработки данных"
6. Обсуждение материалов по кейсам раздела "Дополнительные возможности программирования в Python"

3.6 Тематика курсовых проектов/курсовых работ

Курсовой проект/ работа не предусмотрены

3.7. Соответствие разделов дисциплины и формируемых в них компетенций

Запланированные результаты обучения по дисциплине (в соответствии с разделом 1)	Коды индикаторов	Номер раздела дисциплины (в соответствии с п.3.1)						Оценочное средство (тип и наименование)
		1	2	3	4	5	6	
Знать:								
аналитические и практические способы анализа сложности и эффективности алгоритмов	ИД-1 _{УК-1}			+				Лабораторная работа/Защита работы №6 "Усовершенствованная пузырьковая сортировка. Экспериментальное исследование сортировок. Разработка и реализация экспериментальных тестов"
численные и алгоритмические методы решения типовых задач поиска, сортировки, фильтрации, кодирования, приближенных решений математических задач	ИД-1 _{УК-1}		+					Лабораторная работа/Защита работ №1-2 (1: "Бинарный поиск. Поиск подстроки в строке. Предложение модификаций", 2: "Реализация и анализ алгоритмов сортировки выбором и вставкой")
встроенные и внешние библиотеки языков	ИД-2 _{ОПК-1}			+				Расчетно-графическая работа/Защита расчетного задания №1 "Внешние библиотеки в Python. Графика в Python. Алгоритм имитации нормальных распределений объектов на плоскости с помощью равномерных распределений и их графическое отображение"
основные принципы хранения, доступа и обработки данных в динамических структурах, базах данных, при поточном и многопоточном программировании	ИД-2 _{ОПК-1}						+	Тестирование/(Тест №2) Динамические структуры данных Тестирование/(Тест №3) Базы данных. Инструменты многопоточного программирования
основные концепции программирования	ИД-1 _{ОПК-2}					+		Лабораторная работа/Защита работы №9 "Элементы функционального программирования в Python"
понятие рекуррентных соотношений и методы рекурсии	ИД-1 _{ОПК-2}				+			Контрольная работа/(КР №2) Рекуррентные и рекурсивные алгоритмы
принципы и особенности программирования в языках со статической и динамической типизацией	ИД-1 _{ОПК-2}	+						Контрольная работа/(КР №1) Реализация основных алгоритмов, организация ввода и вывода в C и C++ Тестирование/(Тест №1) Области видимости, пространства имен, время жизни переменных

Уметь:							
адаптировать, модифицировать и комбинировать существующие алгоритмы для решения задач поиска, сортировки, фильтрации данных, кодирования и преобразования данных	ИД-1УК-1		+				Лабораторная работа/Защита работы №4 "Азбука Морзе. Разработка алгоритма кодирования и декодирования на двух языках" Лабораторная работа/Защита работы №5 "Реализация алгоритма пузырьковой сортировки в двух направлениях"
составлять алгоритмы решения задач по словесному описанию методов, производить сравнительный анализ алгоритмов по различным оценкам качества аналитически и на реальных тестах	ИД-1УК-1				+		Лабораторная работа/Защита работы №10 "Методы приближения корней. Указатель на функцию в C(C++)"
программировать собственные классы для динамических объектов на основании принятых базовых операций в заданных структурах данных с применением их в прикладных задачах	ИД-2ОПК-1					+	Расчетно-графическая работа/Защита расчетного задания №2 "Динамические структуры данных. Разработка собственного класса в Python. Запись и считывание объектов (работа с бинарными файлами). Реализация итерируемого объекта и генератора в классе"
использовать возможности встроенных и внешних библиотек, а также графические библиотеки для визуализации решений	ИД-2ОПК-1				+		Расчетно-графическая работа/Защита расчетного задания №1 "Внешние библиотеки в Python. Графика в Python. Алгоритм имитации нормальных распределений объектов на плоскости с помощью равномерных распределений и их графическое отображение"
применять языковые возможности поточной обработки данных для типичных задач	ИД-1ОПК-2					+	Лабораторная работа/Защита работы №9 "Элементы функционального программирования в Python"
решать задачи с применением рекуррентного и рекурсивного подхода, использовать динамические структуры данных	ИД-1ОПК-2					+	Лабораторная работа/Защита работы №7 "Динамические массивы в качестве параметров подпрограмм в C(C++)" Лабораторная работа/Защита работы №8 "Динамические массивы и рекурсивные алгоритмы в C(C++)"
программировать решения типовых задач	ИД-1ОПК-2		+				Лабораторная работа/Защита работы №3 "Реализация

в языках с отличающейся семантикой								стандарта NCR_R (русификация) на двух языках"
------------------------------------	--	--	--	--	--	--	--	---

4. КОМПЕТЕНТНОСТНО-ОРИЕНТИРОВАННЫЕ ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ ОСВОЕНИЯ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ДИСЦИПЛИНЕ)

4.1. Текущий контроль успеваемости

3 семестр

Форма реализации: Компьютерное задание

1. (КР №1) Реализация основных алгоритмов, организация ввода и вывода в С и С++ (Контрольная работа)
2. (Тест №1) Области видимости, пространства имен, время жизни переменных (Тестирование)
3. Защита работ №1-2 (1: "Бинарный поиск. Поиск подстроки в строке. Предложение модификаций", 2: "Реализация и анализ алгоритмов сортировки выбором и вставкой") (Лабораторная работа)
4. Защита работы №3 "Реализация стандарта NCR_R (русификация) на двух языках" (Лабораторная работа)
5. Защита работы №4 "Азбука Морзе. Разработка алгоритма кодирования и декодирования на двух языках" (Лабораторная работа)
6. Защита работы №5 "Реализация алгоритма пузырьковой сортировки в двух направлениях" (Лабораторная работа)
7. Защита работы №6 "Усовершенствованная пузырьковая сортировка. Экспериментальное исследование сортировок. Разработка и реализация экспериментальных тестов" (Лабораторная работа)
8. Защита расчетного задания №1 "Внешние библиотеки в Python. Графика в Python. Алгоритм имитации нормальных распределений объектов на плоскости с помощью равномерных распределений и их графическое отображение"" (Расчетно-графическая работа)

4 семестр

Форма реализации: Компьютерное задание

1. (КР №2) Рекуррентные и рекурсивные алгоритмы (Контрольная работа)
2. (Тест №2) Динамические структуры данных (Тестирование)
3. (Тест №3) Базы данных. Инструменты многопоточного программирования (Тестирование)
4. Защита работы №10 "Методы приближения корней. Указатель на функцию в С(С++)" (Лабораторная работа)
5. Защита работы №7 "Динамические массивы в качестве параметров подпрограмм в С(С++)" (Лабораторная работа)
6. Защита работы №8 "Динамические массивы и рекурсивные алгоритмы в С(С++)" (Лабораторная работа)
7. Защита работы №9 "Элементы функционального программирования в Python" (Лабораторная работа)
8. Защита расчетного задания №2 "Динамические структуры данных. Разработка собственного класса в Python. Запись и считывание объектов (работа с бинарными файлами). Реализация итерируемого объекта и генератора в классе" (Расчетно-графическая работа)

Балльно-рейтинговая структура дисциплины является приложением А.

4.2 Промежуточная аттестация по дисциплине

Зачет с оценкой (Семестр №3)

Оценка определяется в соответствии с Положением о балльно-рейтинговой системе для студентов НИУ «МЭИ» на основании семестровой и аттестационной составляющих

Экзамен (Семестр №4)

Оценка определяется в соответствии с Положением о балльно-рейтинговой системе для студентов НИУ «МЭИ» на основании семестровой и аттестационной составляющих

В диплом выставляется оценка за 4 семестр.

Примечание: Оценочные материалы по дисциплине приведены в фонде оценочных материалов ОПОП.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

5.1 Печатные и электронные издания:

1. Архипов О. Г., Батасова В. С., Гречкина П. В., Зубов В. С., Воробьева И. А., Ионова Т. В., Костина М. Б., Крюков А. А., Чибизова Н. В., Щербин В. М., Марана М. М. - "Программирование. Сборник задач", Издательство: "Лань", Санкт-Петербург, 2019 - (140 с.)
<https://e.lanbook.com/book/121485>;
2. Прикладные задачи дискретной математики и сложность алгоритмов : Учебное пособие для вузов по направлению "Прикладная математика", "Информатика и вычислительная техника" / А. Б. Фролов, Моск. энерг. ин-т (МЭИ ТУ), и др. ; Ред. В. Б. Кудрявцев. – 1997. – 312 с. – ISBN 5-7046-0063-8 : 20000.00.;
3. Ч. Северенс- "Введение в программирование на Python", (2-е изд., испр.), Издательство: "Национальный Открытый Университет «ИНТУИТ»", Москва, 2016 - (231 с.)
<https://biblioclub.ru/index.php?page=book&id=429184>;
4. Златопольский Д. М.- "Основы программирования на языке Python", Издательство: "ДМК Пресс", Москва, 2017 - (284 с.)
<https://e.lanbook.com/book/97359>;
5. Сузи Р. А.- "Язык программирования Python", (2-е изд.), Издательство: "ИНТУИТ", Москва, 2016 - (350 с.)
<https://e.lanbook.com/book/100546>;
6. Страуструп Б.- "Язык программирования C++ для профессионалов", (2-е изд.), Издательство: "ИНТУИТ", Москва, 2016 - (670 с.)
<https://e.lanbook.com/book/100542>;
7. Р. Седжвик- "Алгоритмы на C++", (2-е изд., испр.), Издательство: "Национальный Открытый Университет «ИНТУИТ»", Москва, 2016 - (1773 с.)
<https://biblioclub.ru/index.php?page=book&id=429164>;
8. Численные методы: теория и практика : учебник и практикум для академического бакалавриата по направлению подготовки "Математика. Прикладная математика" / У. Г. Пирумов, [и др.], Московский авиационный ин-т ; ред. У. Г. Пирумов. – 5-е изд., перераб. и доп. – М. : Юрайт, 2015. – 421 с. – (Бакалавр. Академический курс). – ISBN 978-5-9916-4802-8..

5.2 Лицензионное и свободно распространяемое программное обеспечение:

1. СДО "Прометей";
2. Office / Российский пакет офисных программ;
3. Windows / Операционная система семейства Linux;
4. Видеоконференции (Майнд, Сберджаз, ВК и др);
5. Notepad++;

6. Python;
7. GNU Compiler Collection.

5.3 Интернет-ресурсы, включая профессиональные базы данных и информационно-справочные системы:

1. ЭБС Лань - <https://e.lanbook.com/>
2. ЭБС "Университетская библиотека онлайн" - http://biblioclub.ru/index.php?page=main_ub_red
3. Научная электронная библиотека - <https://elibrary.ru/>
4. База данных Web of Science - <http://webofscience.com/>
5. База данных Scopus - <http://www.scopus.com>
6. Национальная электронная библиотека - <https://rusneb.ru/>
7. ЭБС "Консультант студента" - <http://www.studentlibrary.ru/>
8. Электронная библиотека МЭИ (ЭБ МЭИ) - <http://elib.mpei.ru/login.php>
9. Портал открытых данных Российской Федерации - <https://data.gov.ru>
10. База открытых данных Министерства труда и социальной защиты РФ - <https://rosmintrud.ru/opendata>
11. База открытых данных Министерства экономического развития РФ - <http://www.economy.gov.ru>
12. База открытых данных Росфинмониторинга - <http://www.fedsfm.ru/opendata>
13. Электронная открытая база данных "Polpred.com Обзор СМИ" - <https://www.polpred.com>

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Тип помещения	Номер аудитории, наименование	Оснащение
Учебные аудитории для проведения лекционных занятий и текущего контроля	Д-213, Учебная аудитория	парта со скамьей, стол преподавателя, стул, доска меловая
Учебные аудитории для проведения практических занятий, КР и КП	Д-213, Учебная аудитория	парта со скамьей, стол преподавателя, стул, доска меловая
Учебные аудитории для проведения лабораторных занятий	Ж-110, Компьютерный класс ИВЦ	стол, стул, доска меловая, компьютерная сеть с выходом в Интернет, мультимедийный проектор, экран, компьютер персональный, кондиционер
	Ж-113, Компьютерный класс ИВЦ	стол, стул, доска меловая, компьютерная сеть с выходом в Интернет, мультимедийный проектор, экран, компьютер персональный, кондиционер
Учебные аудитории для проведения промежуточной аттестации	Д-213, Учебная аудитория	парта со скамьей, стол преподавателя, стул, доска меловая
Помещения для самостоятельной работы	НТБ-303, Лекционная аудитория	стол компьютерный, стул, стол письменный, вешалка для одежды, компьютерная сеть с выходом в Интернет, компьютер персональный, принтер, кондиционер
Помещения для	Д-213, Учебная	парта со скамьей, стол преподавателя,

консультирования	аудитория	стул, доска меловая
Помещения для хранения оборудования и учебного инвентаря	М-703а/1, Кладовая каф. "ПМИИ"	стеллаж для хранения книг, тумба, экран, ноутбук, книги, учебники, пособия

БАЛЛЬНО-РЕЙТИНГОВАЯ СТРУКТУРА ДИСЦИПЛИНЫ

Программирование и основы алгоритмизации

(название дисциплины)

3 семестр

Перечень контрольных мероприятий текущего контроля успеваемости по дисциплине:

- КМ-1 (Тест №1) Области видимости, пространства имен, время жизни переменных (Тестирование)
- КМ-2 (КР №1) Реализация основных алгоритмов, организация ввода и вывода в С и С++ (Контрольная работа)
- КМ-3 Защита работ №1-2 (1: "Бинарный поиск. Поиск подстроки в строке. Предложение модификаций", 2: "Реализация и анализ алгоритмов сортировки выбором и вставкой") (Лабораторная работа)
- КМ-4 Защита работы №3 "Реализация стандарта NCR_R (русификация) на двух языках" (Лабораторная работа)
- КМ-5 Защита работы №4 "Азбука Морзе. Разработка алгоритма кодирования и декодирования на двух языках" (Лабораторная работа)
- КМ-6 Защита работы №5 "Реализация алгоритма пузырьковой сортировки в двух направлениях" (Лабораторная работа)
- КМ-7 Защита работы №6 "Усовершенствованная пузырьковая сортировка. Экспериментальное исследование сортировок. Разработка и реализация экспериментальных тестов" (Лабораторная работа)
- КМ-8 Защита расчетного задания №1 "Внешние библиотеки в Python. Графика в Python. Алгоритм имитации нормальных распределений объектов на плоскости с помощью равномерных распределений и их графическое отображение"" (Расчетно-графическая работа)

Вид промежуточной аттестации – Зачет с оценкой.

Номер раздела	Раздел дисциплины	Индекс КМ:	КМ-1	КМ-2	КМ-3	КМ-4	КМ-5	КМ-6	КМ-7	КМ-8
		Неделя КМ:	4	7	8	10	11	12	14	16
1	Основные аспекты реализаций Python и C(C++)									
1.1	Введение в язык C(C++). Сравнение реализаций Python и C(C++)		+	+						
2	Основы анализа алгоритмов									
2.1	Введение. Методы поиска, сортировки, кодирования и декодирования				+	+	+	+		
3	Исследование и проектирование алгоритмов									
3.1	Исследование алгоритмов. Внутренние библиотеки в Python								+	

3.2	Проектирование алгоритмов. Внешние библиотеки и графика в Python									+
Вес КМ, %:		5	10	10	5	15	5	25	25	

4 семестр

Перечень контрольных мероприятий текущего контроля успеваемости по дисциплине:

- КМ-9 Защита работы №7 "Динамические массивы в качестве параметров подпрограмм в С(С++)" (Лабораторная работа)
- КМ-10 Защита работы №8 "Динамические массивы и рекурсивные алгоритмы в С(С++)" (Лабораторная работа)
- КМ-11 (КР №2) Рекуррентные и рекурсивные алгоритмы (Контрольная работа)
- КМ-12 Защита работы №9 "Элементы функционального программирования в Python" (Лабораторная работа)
- КМ-13 Защита работы №10 "Методы приближения корней. Указатель на функцию в С(С++)" (Лабораторная работа)
- КМ-14 (Тест №2) Динамические структуры данных (Тестирование)
- КМ-15 (Тест №3) Базы данных. Инструменты многопоточного программирования (Тестирование)
- КМ-16 Защита расчетного задания №2 "Динамические структуры данных. Разработка собственного класса в Python. Запись и считывание объектов (работа с бинарными файлами). Реализация итерируемого объекта и генератора в классе" (Расчетно-графическая работа)

Вид промежуточной аттестации – Экзамен.

Номер раздела	Раздел дисциплины	Индекс КМ:	КМ-9	КМ-10	КМ-11	КМ-12	КМ-13	КМ-14	КМ-15	КМ-16
		Неделя КМ:	4	6	8	9	12	14	15	16
1	Взаимодействие программ с вызывающим контекстом									
1.1	Взаимодействие программ с вызывающим контекстом С(С++). Рекурсия		+	+	+		+			
2	Особенности динамической типизации. Способы поточной обработки данных									
2.1	Особенности динамической типизации. Способы поточной обработки данных					+				
3	Дополнительные возможности программирования в Python									
3.1	Структуры данных. Обзор методов построения и анализа алгоритмов							+	+	
3.2	Введение в объектно-ориентированное									+

	программирование (ООП) в Python								
	Вес КМ, %:	10	10	15	10	10	7	8	30