

**Северский технологический институт –**

филиал федерального государственного автономного образовательного учреждения высшего образования  
«Национальный исследовательский ядерный университет «МИФИ»

**(СТИ НИЯУ МИФИ)**

**Кафедра «Электроники и автоматики физических установок»**

ОДОБРЕНО  
Ученым советом СТИ НИЯУ МИФИ  
протокол № 5 от 28.06.2023

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

**ЯЗЫКИ ПРОГРАММИРОВАНИЯ**

НАПРАВЛЕНИЕ ПОДГОТОВКИ

**09.03.03 Прикладная информатика**

НАИМЕНОВАНИЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

**Цифровизация прикладных процессов и создание информационных систем**

Форма обучения: очная

Семестр	Трудоемкость, ЗЕ	Общий объем курса, час.	Лекции, час.	Практические занятия, час.	Лабораторные работы, час.	В форме практической подготовки / в интерактивной форме, час.	СРС, час.	Форма(ы) контроля (Э, З, ДифЗ, КР, КП)
3	3	108	16	0	32	0	60	Зач.
4	3	108	16	0	32	0	60	ДифЗ
Итого	6	216	32	0	64	0	120	

## Аннотация

Рабочая программа дисциплины «Языки программирования» составлена в соответствии с требованиями образовательного стандарта НИЯУ МИФИ и рабочим учебным планом по направлению подготовки (специальности) 09.03.03 «Прикладная информатика», образовательной программы «Цифровизация прикладных процессов и создание информационных систем».

В результате освоения дисциплины, у выпускника должны быть сформированы следующие результаты обучения (РО):

**1) знать:**

- 3.1 многопоточное и асинхронное программирование на C++
- 3.2 основные концепции функционального программирования на JavaScript
- 3.3 принципы объектно-ориентированного программирования на Python

**2) уметь:**

- У.1 реализовывать многопоточные и асинхронные программы на C++
- У.2 реализовывать функциональное программирование на JavaScript
- У.3 писать объектно-ориентированные приложения на Python

**3) владеть или быть в состоянии продемонстрировать:**

- В.1 навыками многопоточного и асинхронного программирования на C++
- В.2 навыками функционального программирования на JavaScript
- В.3 навыками объектно-ориентированного программирования на Python

### 1 Цели и задачи освоения учебной дисциплины

Целями освоения дисциплины «Языки программирования» являются:

ознакомление с концепциями функциональной и объектно-ориентированной парадигм программирования; освоение многопоточного и асинхронного программирования

Основными задачами дисциплины являются:

- ознакомление с многопоточным и асинхронным программированием на C++
- освоение функционального программирования на JavaScript
- освоение объектно-ориентированного программирования на Python

### 2 Место учебной дисциплины в структуре ООП ВО

Дисциплина «Языки программирования» (Б1.Б.3.5) - Общепрофессиональный модуль образовательной программы.

### 3 Формируемые компетенции и планируемые результаты обучения

Универсальные и общепрофессиональные компетенции:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной	З-ОПК-2 Знать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, используемых при решении задач профессиональной деятельности У-ОПК-2 Уметь выбирать современные информационные

<b>Код и наименование компетенции</b>	<b>Код и наименование индикатора достижения компетенции</b>
деятельности	технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности <b>В-ОПК-2</b> Владеть навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности
<b>ОПК-3</b> Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	<b>З-ОПК-3</b> Знать: принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности <b>У-ОПК-3</b> Уметь: решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности <b>В-ОПК-3</b> Владеть: навыками подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности
<b>ОПК-7</b> Способен разрабатывать алгоритмы и программы, пригодные для практического применения	<b>З-ОПК-7</b> Знать: основные языки программирования и принципы работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий <b>У-ОПК-7</b> Уметь: применять языки программирования, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ <b>В-ОПК-7</b> Владеть: навыками программирования, отладки и тестирования программно-технических комплексов

#### **4 Воспитательный потенциал учебной дисциплины**

Воспитательный потенциал дисциплины «Языки программирования» отражен в Рабочей программе воспитания в Северском технологическом институте – филиале федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский ядерный университет «МИФИ» (<http://www.ssti.ru/education.html>/Информация по образовательным программам).

#### **5 Структура и содержание учебной дисциплины**

##### **5.1 Основные разделы дисциплины, трудоемкость и виды учебной работы**

Настоящая рабочая программа составлена для формы обучения «очная» по направлению 09.03.03 «Прикладная информатика», образовательной программе «Цифровизация прикладных процессов и создание информационных систем».

Общая трудоемкость дисциплины составляет в **зачетных единицах – 6, 216 час.**, обучение по дисциплине проходит в **семестре 3, 4.**

Дисциплина (модуль) содержит **разделы:**

– **раздел 1** – «Многопоточное и асинхронное программирование на C++. Функциональное программирование на JavaScript»

– **раздел 2** – «Объектно-ориентированное программирование на Python»

Трудоемкость, формы и график контроля по разделам дисциплины представлены в таблице 1.

Таблица 1 – Трудоемкость, формы и график контроля отдельных разделов дисциплины

№	Наименование раздела	Виды учебной деятельности, включая самостоятельную работу студентов и трудоемкость, час				Аттестационные мероприятия		Макс. балл за раздел
		Лекции	Практ. занятия	Лабор. работы	Самост. работа	Текущий контроль (нед/форма)	Аттестация раздела (нед/форма)	
<b>3 семестр (18 недель)</b>								
1	Многопоточное и асинхронное программирование на C++. Функциональное программирование на JavaScript	16		32	60	2/ЛР1, 4/ЛР2, 6/ЛР3, 8/ЛР4, 10/ЛР5, 12/ЛР6, 14/ЛР7, 16/ЛР8	16/КР1	60
	Зачет							40
<b>Итого за 3 семестр:</b>		16		32	60			100
<b>4 семестр (18 недель)</b>								
2	Объектно-ориентированное программирование на Python	16		32	60	2/ЛР9, 4/ЛР10, 8/ЛР11, 10/ЛР12, 12/ЛР13, 16/ЛР14	16/КР2	60
	Дифференцированный зачет							40
<b>Итого за 4 семестр:</b>		16		32	60			100

В таблице 2 представлено соответствие содержания каждого раздела и результатов обучения, что позволяет оценить их вклад в достижение целей курса.

Таблица 2 – Соответствие содержания требуемым результатам обучения

Код и наименование индикатора достижения компетенции	Номера разделов	Аттестационные мероприятия
– Знать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, используемых при решении задач профессиональной деятельности (3-ОПК-2)	1, 2	ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)

<p>– Уметь выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности (<b>У-ОПК-2</b>)</p>	<p>1, 2</p>	<p>ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)</p>
<p>– Владеть навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности (<b>В-ОПК-2</b>)</p>	<p>1, 2</p>	<p>ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)</p>
<p>– Знать: принципы, методы и средства решения стандартных задач профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности (<b>З-ОПК-3</b>)</p>	<p>1, 2</p>	<p>ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)</p>
<p>– Уметь: решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности (<b>У-ОПК-3</b>)</p>	<p>1, 2</p>	<p>ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)</p>
<p>– Владеть: навыками подготовки обзоров, аннотаций, составления рефератов, научных докладов, публикаций и библиографии по научно-исследовательской работе с учетом требований информационной безопасности (<b>В-ОПК-3</b>)</p>	<p>1, 2</p>	<p>ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)</p>
<p>– Знать: основные языки программирования и принципы работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий (<b>З-ОПК-7</b>)</p>	<p>1, 2</p>	<p>ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)</p>

– Уметь: применять языки программирования, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ ( <b>У-ОПК-7</b> )	1, 2	ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)
– Владеть: навыками программирования, отладки и тестирования программно-технических комплексов ( <b>В-ОПК-7</b> )	1, 2	ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)

## 5.2 Содержание лекционного курса дисциплины

Содержание лекционного курса дисциплины представлено в таблице 4.

Таблица 3 – Содержание и трудоемкость лекционного курса по разделам в целом по дисциплине

Содержание разделов / тематика разделов	Трудоемкость разделов/тем, ауд. час
<b>Раздел 1 Многопоточное и асинхронное программирование на C++. Функциональное программирование на JavaScript</b>	
<b>1.1 Введение в многопоточность в C++.</b> Поток и процессы. Как создавать и управлять потоками в C++. Безопасность потоков. Состояние гонки.	2
<b>1.2 Синхронизация в многопоточной среде.</b> Блокировки потоков. Примитивы синхронизации в C++: mutex, condition_variable. Атомарные операции. Взаимные блокировки.	2
<b>1.3 Введение в параллельные вычисления в C++.</b> Разница между параллелизмом и многопоточностью (concurrency). Основы параллельных алгоритмов. Знакомство с параллельными вычислениями на языке Си++ (стандартная библиотека C++, библиотека OpenMP)	2
<b>1.4 Введение в асинхронное программирование в C++.</b> Разница между синхронным и асинхронным программированием. Введение в std::future и std::async в C++. Обработка асинхронных задач с помощью std::promise	2
<b>1.5 Введение в программирование на JavaScript.</b> Обзор платформы Node.js. Первичная настройка Node.js для разработки на JavaScript. Основы языка JavaScript: переменные, типы данных, массивы, объекты, циклы, условия. Отладка (debugging) JavaScript-программ.	2
<b>1.6 Продвинутый JavaScript. Возможности ES6.</b> Погружение в возможности JavaScript ES6+: let/const, шаблонные литералы, операторы spread/rest, стрелочные функции, promises, async/await. Модульный JavaScript: export и import.	2

Содержание разделов / тематика разделов	Трудоемкость разделов/тем, ауд. час
<b>1.7 Продвинутый JavaScript. Основы функционального программирования.</b> Расширенные возможности JavaScript: замыкания, обратные вызовы, event loop. Основные концепции функционального программирования: неизменяемость данных, функции как объекты, функции высшего порядка. Функциональное программирование на JavaScript: функции map, reduce, filter.	2
<b>1.8 Функциональное программирование на JavaScript.</b> Концепции функционального программирования: чистые функции, рекурсия, композиция функций, ленивые вычисления. Повторение и связывание концепций функционального программирования с JavaScript.	2
<i>Итого по разделу 1:</i>	16
<b>Раздел 2 Объектно-ориентированное программирование на Python</b>	
<b>2.1 Введение в Python.</b> Краткая история Python. Для чего применяется. Настройка окружения для работы с Python. Базовый синтаксис Python: переменные, типы данных, условия, циклы.	2
<b>2.2 Основы Python.</b> Работа с функциями в Python. Основные структуры данных в Python: списки, кортежи, словари, множества. Работа со строками в Python.	2
<b>2.3 Продвинутая работа с Python.</b> Работа с файлами в Python. Механизм исключений и обработка ошибок в Python. Конструкции try, except, finally. Работа с модулями. Директива "import". Краткий обзор основных библиотек, применяемых в разработке на Python.	2
<b>2.4 Введение в объектно-ориентированное программирование (ООП).</b> Что такое ООП. Основные понятия ООП: класс, объект, поле, метод. Классы и объекты в Python. Основные отличия от процедурного и функционального программирования. Использование диаграмм классов UML для составления документации о системе.	2
<b>2.5 Абстракция и инкапсуляция в ООП.</b> Абстракции в разработке ПО. Как ООП помогает работать с абстракциями. Инкапсуляция. Понятие интерфейса класса. Конструкторы класса. Модификаторы доступа. В чём смысл и польза сокрытия внутренней структуры классов. Отличия абстракции от инкапсуляции.	2
<b>2.6 Наследование в ООП.</b> Что такое наследование. Стандартная реализация методов в родительском классе. Переопределение методов, вызов родительской реализации метода. Абстрактные классы и методы. Преимущества и недостатки наследования. Множественное наследование. Недостатки использования множественного наследования. В каких случаях не стоит применять наследование.	2
<b>2.7 Полиморфизм и композиция в ООП.</b> Что такое полиморфизм. Связь инкапсуляции и наследования с полиморфизмом. Протоколы/интерфейсы в Python. Утиная типизация. Что такое композиция. Как композиция помогает поддерживать компактность иерархий классов. В каких случаях предпочитать композицию, а в каких - наследование.	2
<b>2.8 Pattern matching и шаблонное программирование в Python.</b> Механизм pattern matching в Python. Разница между языками программирования со статической и динамической типизацией. Статическая типизация в Python. Использование шаблонных (generic) типов в Python 3.12.	2

Содержание разделов / тематика разделов	Трудоемкость разделов/тем, ауд. час
<i>Итого по разделу 2:</i>	<i>16</i>
<b>Всего по теоретическому разделу дисциплины:</b>	<b>32</b>

### 5.3 Содержание лабораторного практикума

В таблице 4 представлено содержание и трудоемкость лабораторного практикума дисциплины.

Таблица 4 – Содержание и трудоемкость лабораторного практикума дисциплины

Перечень лабораторных работ по разделам и их содержание	Трудоемкость разделов/тем, ауд. час
<b>Раздел 1 Многопоточное и асинхронное программирование на C++. Функциональное программирование на JavaScript</b>	
<b>1.1 Введение в многопоточность в C++.</b> Реализация программы, создающей несколько потоков и выводящей в консоль сообщение от каждого из них. Демонстрация состояния гонки при работе с несколькими потоками.	4
<b>1.2 Синхронизация в многопоточной среде.</b> Разрешение состояния гонки с использованием mutex. Применение других механизмов синхронизации потоков, их влияние на производительность программы. Взаимная блокировка потоков.	4
<b>1.3 Введение в параллельные вычисления в C++.</b> Реализация последовательную и параллельной пузырьковой сортировки большого массива случайных чисел с использованием стандартной библиотеки C++ (или библиотеки OpenMP). Сравнение времени выполнения программ в каждом из сценариев.	4
<b>1.4 Введение в асинхронное программирование в C++.</b> Создание двух версий программы, выполняющей чтение нескольких больших файлов: последовательной и асинхронной с использованием конструкций <code>std::future</code> and <code>std::async</code> . Сравнение времени обработки в каждом из сценариев.	4
<b>1.5 Введение в программирование на JavaScript.</b> Создание приложения "викторина" на JavaScript с применением таких конструкций JavaScript, как условия, циклы, объекты, массивы. Викторина состоит из списка вопросов и возможных ответов на них, верный из которых только один.	4
<b>1.6 Продвинутый JavaScript. Возможности ES6.</b> Создание приложения "преобразователь текстовых файлов для JavaScript" с применением возможностей ES6: объявление переменных через <code>let/const</code> , шаблонные литералы, стрелочные функции, <code>async/await</code> .	4
<b>1.7 Основы функционального программирования.</b> Создание приложения, проводящего различные операции над массивом чисел, с применением принципов функционального программирования.	4
<b>1.8 Функциональное программирование на JavaScript.</b> Создание приложения, проводящее бинарный поиск заданного элемента рекурсивным образом, а затем применяющее к найденному элементу какую-либо операцию, с применением принципов функционального программирования.	4



Перечень лабораторных работ по разделам и их содержание	Трудоемкость разделов/тем, ауд. час
<i>Итого по разделу 1:</i>	32
<b>Раздел 2 Объектно-ориентированное программирование на Python</b>	
<b>2.1 Создание калькулятора на Python.</b> Реализация простого консольного приложения "калькулятор" с использованием механизмов условий и циклов.	4
<b>2.2 Разработка системы управления библиотекой на Python.</b> Создание программы, симулирующей систему управления библиотекой, с применением функций и основных структур данных (списков и словарей).	4
<b>2.3 Разработка приложения-телефонной книги на Python.</b> Разработка приложения "телефонная книга", данные для которой хранятся в файле, с использованием конструкций обработки ошибок. Работа с файлами.	8
<b>2.4 Проектирование и реализация класса "Банковский счёт" на Python.</b> Реализация класса "Банковский счет" с полями "номер счета", "владелец счета", "баланс". Методы работы полями: пополнить счёт, снять деньги со счёта и т.д. Пополнение счета через консоль, выведение в консоль обновленного баланса пользователя.	4
<b>2.5 Применение абстракции и инкапсуляции в банковской системе.</b> Добавление классов "Владелец" (с полями "имя", "адрес", "идентификационный номер") и "Транзакция" (с полями "дата", "сумма", "тип транзакции") в класс "Банковский счет".	4
<b>2.6 Применение наследования и полиморфизма в модели банковского счёта.</b> Преобразование класса "Банковский счет" в абстрактный. Добавление подклассов "расчетный счет", "сберегательный счет". Каждый из наследников должен иметь свою реализацию абстрактных методов родителя. Добавление абстракции "Банк", содержащей методы по управлению счетами (добавление, удаление, получение объекта-счёта, получение общей суммы на счетах). Принципы наследования и полиморфизма.	8
<i>Итого по разделу 2:</i>	32
<b>Всего по лабораторному практикуму дисциплины:</b>	<b>64</b>

#### 5.4 Тематика практических / семинарских занятий

Практические/семинарские занятия в соответствии с рабочим учебным планом не предусмотрены.

#### 5.5 Курсовое проектирование

Курсовая работа/проект в соответствии с рабочим учебным планом не предусмотрены.

### 6 Образовательные технологии

При проведении лекций используются следующие образовательные технологии: IT-методы, Обучение на основе опыта.

При проведении лабораторных работ используются следующие образовательные технологии: IT-методы, Обучение на основе опыта, Исследовательский метод.

Для организации самостоятельной работы используются следующие образовательные технологии: IT-методы, Исследовательский метод.

## 7 Аннотация фонда оценочных средств

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий текущего, рубежного и промежуточного контроля по дисциплине.

Связь между формируемыми компетенциями и формами контроля их освоения:

Компетенция	Индикаторы освоения	Аттестационные мероприятия
ОПК-2	З-ОПК-2	ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)
ОПК-2	У-ОПК-2	ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)
ОПК-2	В-ОПК-2	ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)
ОПК-3	З-ОПК-3	ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)
ОПК-3	У-ОПК-3	ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)
ОПК-3	В-ОПК-3	ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)
ОПК-7	З-ОПК-7	ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)
ОПК-7	У-ОПК-7	ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)
ОПК-7	В-ОПК-7	ЛР1, ЛР2, ЛР3, ЛР4, ЛР5, ЛР6, ЛР7, ЛР8, КР1, Зачет (3 сем.), ЛР9, ЛР10, ЛР11, ЛР12, ЛР13, ЛР14, КР2, ДЗачет (4 сем.)

**Шкалы оценки образовательных достижений.** Шкала каждого контрольного мероприятия лежит в пределах от 0 до установленного максимального балла включительно. Итоговая аттестация по дисциплине оценивается по 100-балльной шкале и представляет собой сумму баллов, заработанных студентом при выполнении заданий в рамках текущего (**60 баллов**) и промежуточного контроля (**40 баллов**). Для допуска к промежуточному контролю по дисциплине студенту в течение календарного модуля необходимо набрать не менее 60% баллов при условии сдачи **всех** дисциплинарных разделов. Раздел считается сданным, если выполнены все виды контроля и набрано по ним не менее 60 % баллов от максимального по разделу.

В соответствии с учебным планом промежуточная аттестация в конце семестра осуществляется в форме Зачета.

**Аттестация в 3 семестре:**

Вид контроля	Наименование видов контроля	Максимальная положительная оценка в баллах	Минимальная положительная оценка в баллах
<b>Текущая аттестация</b>			
ЛР1	Лабораторная работа	6	3.6
ЛР2	Лабораторная работа	6	3.6
ЛР3	Лабораторная работа	6	3.6
ЛР4	Лабораторная работа	6	3.6
ЛР5	Лабораторная работа	6	3.6
ЛР6	Лабораторная работа	6	3.6
ЛР7	Лабораторная работа	6	3.6
ЛР8	Лабораторная работа	6	3.6
КР1	Контрольная работа	12	7.2
<b>Сумма:</b>		<b>60</b>	<b>36</b>
<b>Промежуточная аттестация</b>			
Зачет		<b>40</b>	<b>24</b>
<b>Итого:</b>		<b>100</b>	<b>60</b>

#### Аттестация в 4 семестре:

Вид контроля	Наименование видов контроля	Максимальная положительная оценка в баллах	Минимальная положительная оценка в баллах
<b>Текущая аттестация</b>			
ЛР9	Лабораторная работа	6	3.6
ЛР10	Лабораторная работа	6	3.6
ЛР11	Лабораторная работа	10	6
ЛР12	Лабораторная работа	6	3.6
ЛР13	Лабораторная работа	6	3.6
ЛР14	Лабораторная работа	10	6
КР2	Контрольная работа	16	9.6
<b>Сумма:</b>		<b>60</b>	<b>36</b>
<b>Промежуточная аттестация</b>			
Дифференцированный зачет		<b>40</b>	<b>24</b>
<b>Итого:</b>		<b>100</b>	<b>60</b>

Итоговая оценка выставляется в соответствии со следующей шкалой:

Сумма баллов по дисциплине	100–90	89–85	84–75	74–70	69–65	64–60	ниже 60
Оценка (ECTS)	A	B	C	D		E	F
Оценка по 4-х бальной шкале	отлично (отл.)	хорошо (хор.)			удовлетворительно (удовл.)		неудовлетворительно (неуд.)
Зачет	Зачтено						Не зачтено

Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, использует в ответе материал монографической литературы.

Оценка «хорошо» выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.

Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно

правильные формулировки, нарушения логической последовательности в изложении программного материала.

Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

#### **Вопросы для Зачета (3 семестр):**

- 1 Отличия процесса от потока
- 2 Проблемы синхронизации потоков при работе с многопоточными приложениями на C++ и их решение
- 3 Как можно решить проблему «состояние гонки» в C++
- 4 Основные конструкции для синхронизации потоков в C++ и их отличия
- 5 Различия между параллельным и многопоточным программированием
- 6 Асинхронное программирование и его реализация на C++ с использованием `std::future` и `std::async`
- 7 Проблемы при использовании мьютексов для синхронизации потоков в многопоточном приложении на C++ и способы их избежать
- 8 Виды циклов в JavaScript
- 9 Способы задания функции в JavaScript
- 10 Асинхронность в JavaScript
- 11 Модульность в JavaScript
- 12 JavaScript event loop
- 13 Функции JavaScript `map`, `reduce`, `filter`
- 14 Основные концепции функционального программирования
- 15 Почему одним из столпов функционального программирования считается неизменяемость данных?
- 16 Что такое «чистые функции», в чём преимущества от их применения?

#### **Вопросы для Дифференцированного зачета (4 семестр):**

- 1 Язык программирования Python и его применение
- 2 Основные структуры данных в Python
- 3 Исключения в Python и их обработка
- 4 Основные понятия объектно-ориентированного программирования
- 5 Преимущества использования абстракций в разработке ПО
- 6 Инкапсуляция и модификаторы доступа в Python
- 7 Абстрактные классы и методы и их применение
- 8 Преимущества и недостатки множественного наследования
- 9 Полиморфизм и его связь с наследованием и инкапсуляцией. Какие преимущества даёт полиморфизм
- 10 Утиная типизация и её применение в Python
- 11 Отличия композиции от наследования. Когда стоит применять композицию, а когда наследование
- 12 Механизм `pattern matching`, используемый в Python
- 13 Разница между языками программирования со статической и динамической типизацией
- 14 Протоколы в Python, их отличия от механизма наследования и абстрактных классов
- 15 Методы работы с файлами в Python
- 16 Обработка ошибок в Python

## **8 Учебно-методическое и информационное обеспечение учебной дисциплины**

### **8.1 Основная литература**

Л1.1 Павловская Т. А. С/С++. Программирование на языке высокого уровня: Учебник для вузов / Т. А. Павловская - Санкт-Петербург: Питер, 2013 - 464 с.

Л1.2 Свердлов С. З. Языки программирования и методы трансляции [Электронный ресурс] / Свердлов С. З. - Санкт-Петербург: Лань, 2021 - 564 с.

Л1.3 Семкин А. О. Информационные технологии. Языки и системы программирования [Электронный ресурс]: учебное пособие / Семкин А. О., Перин А. С. - Москва: ТУСУР, 2021 - 180 с.

### **8.2 Дополнительная литература**

Л2.1 Борзунов С. В. Языки программирования. Python: решение сложных задач [Электронный ресурс]: учебное пособие для вузов / Борзунов С. В., Кургалин С. Д. - Санкт-Петербург: Лань, 2023 - 192 с.

Л2.2 Мерсов А. А. Языки программирования [Электронный ресурс] / Мерсов А. А., Русаков А. М., Филатов В. В. - Москва: РТУ МИРЭА, 2022 - 83 с.

Л2.3 Ненашев В. А. Языки программирования в моделировании и обработке информации. С++ [Электронный ресурс]: учебно-методическое пособие / Ненашев В. А., Григорьев Е. К. - Санкт-Петербург: ГУАП, 2022 - 107 с.

### **8.3 Информационно-образовательные ресурсы**

Э1 <https://stepik.org/course/149/promo> - многопоточное программирование на С++

Э2 <https://gb.ru/blog/funktionalnoe-programmirovanie/> - функциональное программирование

Э3 <https://proglib.io/p/python-oop> - ООП на Python

## **9 Материально-техническое обеспечение учебной дисциплины**

Материально-техническое обеспечение дисциплины приведено на сайте СТИ НИЯУ МИФИ <http://www.ssti.ru/objects.html>

## **10 Учебно-методические рекомендации для студентов**

Самостоятельная работа студентов – это планируемая учебная и внеаудиторная работа студентов, выполняемая по заданию преподавателя и под его методическим руководством, но без его непосредственного участия.

Целью самостоятельной работы студентов является приобретение новых знаний, систематизация и закрепление полученных теоретических знаний и практических умений студентов.

**Лекции.** Рекомендации по написанию конспекта лекций: кратко, последовательно фиксировать основные положения, выводы, формулировки, обобщения: пометать основные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь (тезаурус). Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на следующем занятии или консультации.

**Лабораторные работы.** Подготовка к лабораторной работе включает в себя работу с конспектом лекций, рекомендуемой литературой, подготовку ответов к контрольным вопросам для допуска к выполнению лабораторной работы, написание отчета.

Лабораторные занятия проводятся в специализированных лабораториях института.

Прежде чем начать занятия в данной лаборатории студент знакомится с правилами техники безопасности, о чем расписывается в журнале. В лабораториях кафедры запрещается находиться в верхней одежде. Запрещается класть на рабочий стол сумки, пакеты, шапки и другие посторонние предметы. Студент приступает к выполнению лабораторной работы только после ознакомления с описанием работы и подготовки к ней. Запрещается включать какие-либо приборы или без предварительной проверки их преподавателем или лаборантом.

Для записи результатов измерения в отчете должны быть заранее подготовлены таблицы, включающие как сами измерения, так и их погрешности.

К следующему занятию студент готовит очередную работу и предъявляет отчет о работе, выполненной на предыдущем занятии. Работа считается окончательно сданной после защиты отчета.

**Промежуточная аттестация.** Для подготовки к промежуточной аттестации студенту необходимо проработать конспекты лекционных и практических занятий, подготовить ответы к вопросам, выносимым на промежуточную аттестацию, при необходимости воспользоваться рекомендуемой литературой.

## **11 Учебно-методические рекомендации для преподавателей**

На лекционных, лабораторных занятиях студентам сообщаются новые сведения, систематизируется и обобщается накопленный запас знаний, формируются на этой основе познавательные и профессиональные интересы. Преподаватель, проводя занятия, должен стремиться увлечь студентов, активно воздействовать на их эмоции, вызвать интерес к учебному предмету, стремление постоянно пополнять знания.

Самостоятельная работа студентов по данному курсу

- Проработка лекционного материала
- Подготовка к лабораторным работам
- Подготовка к контрольным работам
- Оформление отчетов по лабораторным работам
- Подготовка к промежуточному контролю: Зачет (3 семестр)

В течение 3 семестра осуществляется контроль знаний студентов: см. раздел 5.1.

По результатам аттестационных мероприятий формируется допуск студента к итоговому контролю – Зачету по дисциплине. Студент на Зачете должен показать знание программного материала, исчерпывающе, последовательно, четко и логически стройно его излагать, уметь тесно увязывать теорию с практикой, использовать в ответе материал рекомендуемой литературы.

- Подготовка к промежуточному контролю: Дифференцированный зачет (4 семестр)

В течение 4 семестра осуществляется контроль знаний студентов: см. раздел 5.1.

По результатам аттестационных мероприятий формируется допуск студента к итоговому контролю – Дифференцированному зачету по дисциплине. Студент на Дифференцированном зачете должен показать знание программного материала, исчерпывающе, последовательно, четко и логически стройно его излагать, уметь тесно увязывать теорию с практикой, использовать в ответе материал рекомендуемой литературы.

\*\*\*

Автор(ы): Н.А. Мельникова