

ВЫСШАЯ ИНЖИНИРИНГОВАЯ ШКОЛА

ОДОБРЕНО УМС ВИШ

Протокол № 132/15-12-22

от 15.12.2022 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

**ЯЗЫКИ ПРОГРАММИРОВАНИЯ. ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАМНО-  
ИНФОРМАЦИОННЫХ СИСТЕМ.**

Направление подготовки  
(специальность)

[1] 09.04.02 Информационные системы и  
технологии

Семестр	Трудоемкость, кред.	Общий объем курса, час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	В форме практической подготовки/ В СРС, час.	КСР, час.	Форма(ы) контроля, экс./зач./КР/КП
1	6	216	0	56	0	124	0	Э КР
2	4	144	0	48	0	60	0	Э КР
3	5	180	0	64	0	80	0	Э КР
Итого	15	540	0	168	0	264	0	

## АННОТАЦИЯ

Дисциплина реализует последовательно, на протяжении 3 семестров и направлена на получение студентами систематических навыков работы с различными языками программирования. В данной программе реализованы три семестровых блока:

- Изучение языка Python;
- Изучение языка C#;
- Изучение языка Java.

Дисциплина содействует формированию у студентов профессиональных компетенций, необходимых для решения задач, относящихся к определенному виду профессиональной деятельности.

Первый блок программы (первый семестр) на формирование у студентов алгоритмического и структурного мышления, а также понимание базовых возможностей языка Python. Алгоритмы — это основа программирования, определяющая, каким образом программное обеспечение будет использовать структуры данных. Студенты познакомятся с базовыми аспектами построения алгоритмов, основными понятиями и определениями, структурами данных, затем перейдут к основным методам построения алгоритмов, неразрешимости и методам решения неразрешимых задач, и, наконец, изучат проектирование алгоритмов. Алгоритмический анализ состоит из двух фундаментальных компонентов: выделения математически чистого ядра задачи и выявления методов проектирования подходящего алгоритма на основании структуры задачи. Чем лучше студент владеет полным арсеналом возможных методов проектирования, тем быстрее он начинает распознавать «чистые» формулировки, лежащие в основе запутанных задач реального мира. Второй блок программы (второй семестр) направлен на изучение основ C#. Третий блок программы (третий семестр) посвящен приобретению студентами навыков программирования на языке Java, включая синтаксис языка, основные понятия языка «метод», исключение, «процесс», «поток» и других, и работу с данными в языке Java.

### 1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Целями освоения учебной дисциплины является изучение основ языков программирования Python, C# и Java, в том числе:

- формирование у студентов профессиональных компетенций, связанных с использованием теоретических знаний в области структур данных и теории алгоритмов, пониманием концепции абстрактных типов данных и подходов к их реализации;
- способность студентами оценивать влияние выбора структуры данных и/или алгоритмов на производительность (быстродействие/эффективность) программ;
- овладение инструментарием объектно-ориентированного программирования;
- развитие умений, основанных на полученных теоретических знаниях, позволяющих на творческом и репродуктивном уровне предлагать и применять эффективные подходы к решению (алгоритмизации) поставленных задач с использованием данных простой и сложной структуры.

### 2. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Дисциплина является неотъемлемой составной частью профессиональной подготовки магистров. Вместе с другими дисциплинами общенаучного и профессионального циклов дисциплин изучение данной дисциплины призвано формировать специалиста, и в частности, выработать у него такие качества, как:

- строгость в суждениях,
- творческое мышление,
- организованность и работоспособность,
- дисциплинированность,
- самостоятельность и ответственность.

Дисциплина относится к числу дисциплин профессионального цикла.

Для успешного усвоения данной дисциплины необходимо, чтобы студент владел знаниями, умениями и навыками в объеме основ информатики, проектирования программных систем, сформированными в процессе освоения программы подготовки бакалавров или программ подготовки специалистов по родственным направлениям высшего профессионального образования.

Знания, полученные при изучении дисциплины, являются базовыми для профессиональных дисциплин, входящих в вариативную часть профессионального цикла учебного плана подготовки магистров по направлению 09.04.02 «Информационные системы и технологии».

### 3. ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Универсальные и(или) общепрофессиональные компетенции:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ОПК-2 [1] – Способен разрабатывать оригинальные алгоритмы и программные средства, в том числе с использованием современных интеллектуальных технологий для решения профессиональных задач	З-ОПК-2 [1] – Знать: методы современных интеллектуальных технологий для разработки оригинальных алгоритмов и программных средств. У-ОПК-2 [1] – Уметь: обосновывать выбор современных интеллектуальных технологий и программной среды при разработке оригинальных программных средств В-ОПК-2 [1] – Владеть:современными интеллектуальными технологиями разработки алгоритмов и программных средств для решения профессиональных задач.
ОПК-5 [1] – Способен разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем	З-ОПК-5 [1] – Знать: современные технологии разработки программного и аппаратного обеспечения информационных и автоматизированных систем. У-ОПК-5 [1] – Уметь: проектировать, разрабатывать и модернизировать программное и аппаратное обеспечение информационных и автоматизированных систем. В-ОПК-5 [1] – Владеть: технологиями и навыками разработки и модернизации программного и аппаратного обеспечение информационных и автоматизированных систем

<p>ОПК-8 [1] – Способен осуществлять эффективное управление разработкой программных средств и проектов.</p>	<p>З-ОПК-8 [1] – Знать: методы управления процессом разработки программных средств и проектов на всех стадиях жизненного цикла  У-ОПК-8 [1] – Уметь: оценивать экономическую эффективность и качество разрабатываемых программных средств и проектов, управлять их надежностью и информационной безопасностью.  В-ОПК-8 [1] – Владеть: современными инструментальными средствами управления разработкой программных средств и проектов.</p>
<p>УКЦ-2 [1] – Способен к самообучению, самоактуализации и саморазвитию с использованием различных цифровых технологий в условиях их непрерывного совершенствования</p>	<p>З-УКЦ-2 [1] – Знать основные цифровые платформы, технологии и интернет ресурсы используемые при онлайн обучении  У-УКЦ-2 [1] – Уметь использовать различные цифровые технологии для организации обучения  В-УКЦ-2 [1] – Владеть навыками самообучения, самоактуализации и саморазвития с использованием различных цифровых технологий</p>

Профессиональные компетенции в соответствии с задачами и объектами (областями знаний) профессиональной деятельности:

<b>Задача профессиональной деятельности (ЗПД)</b>	<b>Объект или область знания</b>	<b>Код и наименование профессиональной компетенции;  <b>Основание (профессиональный стандарт-ПС, анализ опыта)</b></b>	<b>Код и наименование индикатора достижения профессиональной компетенции</b>
научно-исследовательский			
<p>Разработка методов и методик научных исследований в сферах информационных и цифровых технологий сложных инженерных объектов.</p>	<p>Методы и методики научных исследований в сфере цифровых технологий, планы и программы НИР в сфере цифровых технологий</p>	<p>ПК-6 [1] - Способен к восприятию и использованию новейших достижений в области информационных систем и технологий</p> <p><i>Основание:</i>  Профессиональный стандарт: 06.003, 06.015</p>	<p>З-ПК-6[1] - Знать: новейшие достижения в области информационных систем и технологий, информационных сетей нового поколения, общественных сервисов информационной безопасности, технологии распределенных реестров. ;  У-ПК-6[1] - Уметь: эффективно воспринимать и использовать новейшие достижения</p>

			в области ИСТ в профессиональной деятельности.; В-ПК-6[1] - Владеть: навыками адаптации новейших достижений в области ИСТ к использованию в профессиональной деятельности.
производственно-технологический			
Осуществление эксплуатации программно-аппаратных комплексов, обеспечение соответствия программных и аппаратных компонент решаемым производственно-технологическим задачам, эффективное внедрение цифровых продуктов и АПК в производство, нормативную поддержку и документационное обеспечение процессов эксплуатации аппаратно-программных систем для эффективного решения производственно-технологических задач.	Процессы эксплуатации информационных систем, нормативно-техническая документация.	ПК-8 [1] - Способен разрабатывать нормативную и техническую документацию на аппаратные средства и программное обеспечение, осуществлять анализ ИТ-продуктов на соответствие задачам пользователей  <i>Основание:</i> Профессиональный стандарт: 06.015, 06.016	З-ПК-8[1] - Знать: существующие нормативные документы и стандарты на аппаратные средства и программное обеспечение в области профессиональной деятельности. ; У-ПК-8[1] - Уметь: разрабатывать нормативную и техническую документацию на программное обеспечение и аппаратные средства в соответствии с принятыми стандартами и осуществлять анализ ИТ-продуктов на соответствие задачам пользователей.; В-ПК-8[1] - Владеть: навыками разработки нормативной и технической документацию на аппаратные средства и программное обеспечение.

#### 4. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Разделы учебной дисциплины, их объем, сроки изучения и формы контроля:

№ п.п	Наименование раздела учебной дисциплины	Недели	Лекции/ Практи. (семинары )/ Лабораторные работы, час.	Обязат. текущий контроль (форма*, неделя)	Максимальный балл за раздел**	Аттестация раздела (форма*, неделя)	Индикаторы освоения компетенции
	<i>1 Семестр</i>						
1	Первый раздел	1-8	0/32/0		25	КИ-8	З-ОПК-2, 3-ОПК-5, У-ОПК-5
2	Второй раздел	9-16	0/24/0		25	КИ-16	У-ПК-6, В-ПК-6, 3-ПК-8
	<i>Итого за 1 Семестр</i>		0/56/0		50		
	<b>Контрольные мероприятия за 1 Семестр</b>				50	Э, КР	3-ПК-8, У-ПК-8, 3-ПК-8, В-ПК-8
	<i>2 Семестр</i>						
1	Первый раздел	1-8	0/24/0		25	КИ-8	З-ОПК-8, У-ОПК-8, У-ПК-8, В-ПК-8
2	Второй раздел	9-15	0/24/0		25	КИ-15	У-ОПК-2, У-ОПК-5, В-ПК-6,

							У-ПК-8
	<i>Итого за 2 Семестр</i>		0/48/0		50		
	<b>Контрольные мероприятия за 2 Семестр</b>				50	Э, КР	У-ОПК-2, У-ОПК-5, В-ПК-6, В-ПК-8
	<i>3 Семестр</i>						
1	Первый раздел	1-8	0/32/0		25	КИ-8	3-ОПК-2, 3-ОПК-5, В-ОПК-5
2	Второй раздел	9-16	0/32/0		25	КИ-16	В-ОПК-8, У-ПК-6, В-ПК-6, 3-УКЦ-2, У-УКЦ-2, В-УКЦ-2
	<i>Итого за 3 Семестр</i>		0/64/0		50		
	<b>Контрольные мероприятия за 3 Семестр</b>				50	Э, КР	3-ПК-6, У-ПК-6, В-ОПК-2, В-ПК-6

\* – сокращенное наименование формы контроля

\*\* – сумма максимальных баллов должна быть равна 100 за семестр, включая зачет и (или) экзамен

Сокращение наименований форм текущего контроля и аттестации разделов:

Обозначение	Полное наименование
КИ	Контроль по итогам
Э	Экзамен
КР	Курсовая работа

## КАЛЕНДАРНЫЙ ПЛАН

Недели	Темы занятий / Содержание	Лек., час.	Пр./сем., час.	Лаб., час.
	<i>1 Семестр</i>	0	56	0
<b>1-8</b>	<b>Первый раздел</b>	0	32	0
1 - 2	<b>Основы программирования на языке Python.</b> Введение. Принцип работы программ. Процесс разработки программы, отладка. Возможные ошибки в программе (синтаксические ошибки, ошибки во время исполнения). Использование интерпретатора языка Python: интерактивный режим, интерпретация программного кода в файле. Основные понятия и синтаксические конструкции. Переменная, значение, присваивание, оценка выражения. Операторы. Арифметические операторы, логические операторы, операции над строками, порядок выполнения операций. Встроенные математические функции. Функции для работы со строками. Типы данных. Операторы для реализации нелинейных алгоритмов. Условный оператор, операторы цикла while и for. Прерывание цикла. Оформление блоков кода в Python и других языках. Функции, аргументы, возвращаемые значения. Модули и библиотеки в Python. Чтение и запись файлов. Работа со случайными числами и вероятностными распределениями. Структуры данных. Массивы, адресация элементов массива, функции для работы с массивами, срезы массивов. Строка как массив. Ассоциативные массивы (словари), способы их задания в Python и функции для работы с ними.	Всего аудиторных часов		
		0	8	0
		Онлайн		
		0	0	0
3 - 4	<b>Структурное программирование.</b> Описание простых функций с параметрами. Декомпозиция задачи. Структурное программирование. Проектирование «сверху-вниз». Стек вызовов. Полиморфизм в Python. Duck typing. Значения параметров по умолчанию. Именованные параметры функций. Метод грубой силы. Тест простоты числа. Разложение числа на множители. Структурное программирование на примере постройки дома. Риелтор. Покупатель. Архитектор. Прораб. Тестировщик. Разнорабочие.	Всего аудиторных часов		
		0	8	0
		Онлайн		
		0	0	0

5 - 6	<b>Список как массив данных.</b> Список как массив данных. Линейный поиск в массиве. Копирование массива. Копирование задом-наперёд. Алгоритм обращения массива. Алгоритм циклического сдвига в массиве. Поиск максимального/минимального элемента в массиве. Добавление элемента в конец и в начало массива. Удаление элемента из конца и начала массива. Ссылочная модель данных в Python. Изменяемость списка. Решето Эратосфена. Массивы. Списки. Алгоритмическая сложность. Временная сложность. Асимптотики. Конструкции.	Всего аудиторных часов		
		0	8	0
		Онлайн		
		0	0	0
7 - 8	Методы append(), pop() и функция len() для списка. Списковые включения. Мастер-класс по TDD. Сортировка вставками. Сортировка выбором. Сортировка методом пузырька. Сортировка подсчётом. Стек. Интернированные объекты	Всего аудиторных часов		
		0	8	0
		Онлайн		
		0	0	0
9-16	<b>Второй раздел</b>	0	24	0
9 - 10	<b>Двумерное динамическое программирование.</b> Двумерное динамическое программирование. Наибольшая общая подпоследовательность. Наибольшая возрастающая подпоследовательность. Расстояние Левенштейна. Проверка равенства строк. Наивный поиск подстроки в строке. Алгоритм Кнута-Морриса-Пратта. Фракталы. Суммы ряда. Шахматные задачи. Графовые задачи. Ханойские башни.	Всего аудиторных часов		
		0	8	0
		Онлайн		
		0	0	0
11 - 12	<b>Рекурсия.</b> Рекурсия. Сказка "Репка" и изготовление матрёшки. Прямой и обратный ход рекурсии. Фрактальный квадрат в квадрате(теория). Факториал числа. Алгоритм Евклида. Быстрое возведение в степень. Реализация сортировок. Матрёшка. Фрактальный квадрат в квадрате	Всего аудиторных часов		
		0	0	0
		Онлайн		
		0	0	0
13 - 14	<b>Генерация комбинаторных объектов. Алгоритмы сортировки.</b> Генерация комбинаторных объектов. Рекурсивная генерация всех чисел длины M. Генерация всех перестановок (рекурсивная). Быстрые сортировки: Тони Хоара и слиянием (теория без реализации). Слияние двух упорядоченных массивов. Реализация быстрых сортировок. Сортировка слиянием (реализация).	Всего аудиторных часов		
		0	8	0
		Онлайн		
		0	0	0
15 - 16	<b>Алгоритмы сортировки (продолжение)</b> Устойчивость сортировок. Проверка упорядоченности массива за $O(N)$ . Вычисление чисел Фибоначчи и проблема перевычислений. Одномерное динамическое программирование на примере чисел Фибоначчи. Задачи о кузнечике (количество траекторий, траектория наименьшей стоимости). Двумерные массивы (списки списков). Оператор is.	Всего аудиторных часов		
		0	8	0
		Онлайн		
		0	0	0
2 Семестр		0	48	0

1-8	<b>Первый раздел</b>	0	24	0
1	<b>Обзор синтаксиса C#</b> Обзор способов разработки приложений с использованием C# Типы данных, операторы и выражения Программирование языковых конструкций на C# Обработка исключений	Всего аудиторных часов		
		0	4	0
		Онлайн		
		0	0	0
2 - 3	<b>Объектно-ориентированное программирование в C#</b> Понятие класса Создание и вызов методов Статические члены Создание перегруженных методов Использование опциональных параметров и параметров, возвращающих значение Конструкторы Свойства Наследование Интерфейсы	Всего аудиторных часов		
		0	4	0
		Онлайн		
		0	0	0
4	<b>Продвинутое аспекты ООП (объектно-ориентированного программирования)</b> Реализация структур и перечислений Создание универсальных (generic) типов Делегаты и лямбда-выражения Основы LINQ Обработка событий	Всего аудиторных часов		
		0	4	0
		Онлайн		
		0	0	0
5	<b>Чтение и запись локальных данных</b> Чтение и запись файлов Сериализации и десериализации данных Реализация ввода/вывода с помощью потоков	Всего аудиторных часов		
		0	4	0
		Онлайн		
		0	0	0
6 - 7	<b>Доступ к базам данных</b> Основы реляционной модели данных Низкоуровневые классы Создание и использование моделей данных (Entity Data Models) Запрос данных с помощью LINQ Обновление данных с помощью LINQ Современные подходы NO SQL	Всего аудиторных часов		
		0	4	0
		Онлайн		
		0	0	0
8	<b>Основы Web приложений</b> Основы HTML и JavaScript Подход MVC Подход Web API	Всего аудиторных часов		
		0	4	0
		Онлайн		
		0	0	0
9-15	<b>Второй раздел</b>	0	24	0
9	<b>Многопоточность</b> Реализация многопоточности Асинхронное выполнение операций Синхронизация одновременного доступа к данным	Всего аудиторных часов		
		0	4	0
		Онлайн		
		0	0	0
10	<b>Управление памятью и другими ресурсами</b> Сборщик мусора	Всего аудиторных часов		
		0	4	0

	Интерфейс IDisposable Взаимодействие с неуправляемым кодом	Онлайн	0	0	0
11	<b>Рефлексия</b> Проверка метаданных объекта Создание и использование пользовательских атрибутов Генерация управляемого кода	Всего аудиторных часов	0	4	0
		Онлайн	0	0	0
12 - 13	<b>Алгоритмические задачи</b> Производительность алгоритмов Классические алгоритмы Подходы к решению алгоритмических задач	Всего аудиторных часов	0	8	0
		Онлайн	0	0	0
14 - 15	<b>Инструменты и принципы командной работы</b> <b>название пункта</b> Введение в Agile и Scrum Git, TFS и их интеграция со средой разработки	Всего аудиторных часов	0	4	0
		Онлайн	0	0	0
	<i>3 Семестр</i>		0	64	0
<b>1-8</b>	<b>Первый раздел</b>		0	32	0
1 - 4	<b>Принципы ООП (объектно-ориентированного программирования)</b> Принципы ООП. Понятие “метод”. Понятие “исключение”. Обрабатываемые и не обрабатываемые исключения. Конструкция используется в Java для обработки исключений. Логические операции и операторы. Понятию “коллекция”. Сериализация.	Всего аудиторных часов	0	16	0
		Онлайн	0	0	0
5 - 8	<b>Виды потоков ввода/вывода.</b> Преобразование байтовых потоки в символьные и обратно. Понятию "процесс" и "поток". Понятию “метод”. Поле/атрибут класса. Таблица истинности. Иерархии исключений. Пользовательские исключение. Понятию “пул строк”. Синхронизация потоков. Взаимодействия программ, процессов и потоков. Примитивные типы. Примитивные типы данных, потеря данных, преобразования логического типа. Конструкторы по-умолчанию, копирование и конструктор с параметрами. Работа с элементами файловой системы. Блок. Завершение выполнения потока. Коллекции List? Синхронизация метода. Интерфейс FilenameFilter. Преобразование строки в число. Сравнение строк. Монитор при вызове нестатического и статического метода.	Всего аудиторных часов	0	16	0
		Онлайн	0	0	0
<b>9-16</b>	<b>Второй раздел</b>		0	32	0
9 - 12	<b>Создание нескольких потоков.</b> Выполнение одного кода в программе два потока. Инициализация переменных по умолчанию. Понятие “монитор” в контексте многопоточности. Монитор при выполнении участка кода метода. Модификации уровня доступа. Тернарный оператор выбора. Понятие “взаимная блокировка” в контексте многопоточности.	Всего аудиторных часов	0	16	0
		Онлайн	0	0	0

	Примеры использования модификаторов <code>volatile</code> и метода <code>yield()</code> . Обращение обратной последовательности символов строки.			
13 - 16	<b>Принудительный выброс исключений.</b> Дополнительные условия к методу, который потенциально может выбросить исключение. Классы-надстройки. Понятие "поток-демон. Принудительная остановка потока. <code>ThreadPoolExecutor</code> , его назначение.	Всего аудиторных часов		
		0	16	0
		Онлайн		
		0	0	0

Сокращенные наименования онлайн опций:

Обозначение	Полное наименование
ЭК	Электронный курс
ПМ	Полнотекстовый материал
ПЛ	Полнотекстовые лекции
ВМ	Видео-материалы
АМ	Аудио-материалы
Прз	Презентации
Т	Тесты
ЭСМ	Электронные справочные материалы
ИС	Интерактивный сайт

## 5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

С целью формирования и развития профессиональных навыков студентов в дисциплине используются активные и интерактивные формы проведения занятий. Ключевую роль имеют выполнение задач в программной оболочке соответствующего языка программирования.

В соответствии со спецификой университета в процессе преподавания дисциплины методически целесообразно в каждом разделе выделить наиболее важные темы и акцентировать на них внимание обучаемых.

## 6. ФОНД 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 представлена в следующей таблице:

Компетенция	Индикаторы освоения	Аттестационное мероприятие (КП 1)	Аттестационное мероприятие (КП 2)	Аттестационное мероприятие (КП 3)
ОПК-2	З-ОПК-2	КИ-8		КИ-8
	У-ОПК-2		Э, КИ-15	
	В-ОПК-2			КР
ОПК-5	З-ОПК-5	КИ-8		КИ-8

	У-ОПК-5	КИ-8	Э, КИ-15	
	В-ОПК-5			КИ-8
ОПК-8	З-ОПК-8		КИ-8	
	У-ОПК-8		КИ-8	
	В-ОПК-8			КИ-16
ПК-6	З-ПК-6			Э
	У-ПК-6	КИ-16		Э, КИ-16
	В-ПК-6	КИ-16	КР, КИ-15	КР, КИ-16
ПК-8	З-ПК-8	КР, Э, КИ-16		
	У-ПК-8	Э	КИ-8, КИ-15	
	В-ПК-8	КР	КР, КИ-8	
УКЦ-2	З-УКЦ-2			КИ-16
	У-УКЦ-2			КИ-16
	В-УКЦ-2			КИ-16

### 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 и промежуточного контроля.

Итоговая оценка выставляется в соответствии со следующей шкалой:

Сумма баллов	Оценка по 4-ех балльной шкале	Оценка ECTS	Требования к уровню освоению учебной дисциплины
90-100	5 – <i>«отлично»</i>	A	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.
85-89	4 – <i>«хорошо»</i>	B	Оценка «хорошо» выставляется студенту, если он твёрдо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.
75-84		C	
70-74		D	
65-69	3 – <i>«удовлетворительно»</i>	E	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.
60-64			
Ниже 60	2 – <i>«неудовлетворительно»</i>	F	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.
--	--	--	---

## 7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

### ОСНОВНАЯ ЛИТЕРАТУРА:

1. 004 P55 CLR via C# . Программирование на платформе Microsoft.NET Framework 4.5 на языке C# : , Москва [и др.]: Питер, 2013
2. ЭИ В 19 Java. Объектно-ориентированное программирование. Учебное пособие. Стандарт третьего поколения : , Санкт-Петербург: Питер, 2011
3. ЭИ С 17 Python на практике : учебное пособие, Москва: ДМК Пресс, 2014
4. 510 A45 Алгоритмы : построение и анализ, Москва [и др.]: Вильямс, 2011
5. ЭИ К 88 Основы алгоритмизации и программирования на языке C# : учебное пособие для вузов, Москва: Юрайт, 2021
6. ЭИ Н 62 Прикладное программирование : учебное пособие, Санкт-Петербург: Лань, 2022
7. 004 Л 86 Программирование на Python Т.1 , Санкт-Петербург ; Москва: Символ-Плюс, 2018
8. 004 Л 86 Программирование на Python Т.2 , Санкт-Петербург ; М.: Символ-Плюс, 2018
9. ЭИ П 44 Программирование. Базовый курс C# : учебник для вузов, Москва: Юрайт, 2020
10. ЭИ Ш 35 Программирование: математическая логика : учебное пособие для спо, Москва: Юрайт, 2021
11. 004 М 15 Совершенный код. Мастер-класс : практ. рук-во по разработке ПО, Санкт-Петербург: БХВ, 2018
12. 004 М 29 Чистый код. Создание, анализ и рефакторинг : , Санкт-Петербург: Питер, 2018

### ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА:

### ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ:

Специальное программное обеспечение не требуется

### LMS И ИНТЕРНЕТ-РЕСУРСЫ:

<https://online.mephi.ru/>

## **8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

Специальное материально-техническое обеспечение не требуется

## **9. УЧЕБНО-МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ СТУДЕНТОВ**

Успешное освоение дисциплины требует от студентов посещения лекций, активной работы во время практических занятий, выполнения всех учебных заданий преподавателя, ознакомления с базовыми учебниками, основной и дополнительной литературой, а также предполагает творческое участие студента путем планомерной, повседневной работы.

Лекционный материал тесно связан с выполнением практических заданий на семинарах. Посещение лекций является обязательным.

Перед выполнением практических работ студент должен заранее изучить теоретический и учебно-методический материалы, относящиеся непосредственно к выполнению данной работы. При необходимости студент может обратиться к преподавателю за консультацией по вопросам, относящимся к выполнению данной работы.

Практические задания являются необходимым элементом данного модуля. Значимость успешного выполнения практических заданий определяется тем, что во время прохождения студенты получают необходимые практические навыки и умения работы с современным цифровым инструментарием. Основная цель практического обучения состоит в формировании и закреплении первичных теоретических знаний и профессиональных навыков. В ходе практических занятий обычно формируется теоретическая и практическая база будущей профессиональной деятельности.

## **10. УЧЕБНО-МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРЕПОДАВАТЕЛЕЙ**

Целью работы преподавателя должно быть эффективное восприятие материала слушателями.

Со стороны преподавателя должен быть установлен контакт со студентами, и они должны быть информированы о порядке прохождения курса, его особенностях, учебно-методическом обеспечении по дисциплине.

В ходе подготовки лекций, указанных в рабочей программе модуля, преподаватель разрабатывает план лекции, определяет моменты, которые слушатели должны усвоить на лекции, и освоить в ходе самостоятельной работы с литературой.

Преподаватель дает методические рекомендации обучаемым по самостоятельному изучению проблем, характеризуя пути и средства достижения поставленных перед ними задач, высказывает советы и рекомендации по изучению учебной литературы, самостоятельной и групповой практической работе.

При подготовке к практическому занятию преподаватель готовит план его проведения, знакомится с новыми публикациями по теме.

Преподаватель предоставляет учащимся обратную связь о выполненных практических заданиях, ставит перед учащимися четкие цели и представляет новый материал с той степенью

подробности изложения, чтобы материал был усвоен, но учащиеся не чувствовали себя перегруженными. Учащимся предоставляется инструкции и стратегии для выполнения практического задания. Для проверки текущего уровня понимания лекционных занятий задаются вопросы для понимания степени усвоения материала. Когда учащиеся работают индивидуально, преподаватель контролирует их деятельность.

Автор(ы):

Дунаев Максим Евгеньевич

Даньшин Вадим Владимирович

Анриенко Юрий Александрович