

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»**

**Рабочая программа дисциплины
базовая часть**

**Б. 1.6.1 Модуль "Системное и прикладное программное обеспечение".
Программирование**

**ОСНОВНАЯ ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
ПОДГОТОВКИ БАКАЛАВРА
(программа академического бакалавриата)
по направлению
09.03.01 Информатика и вычислительная техника
Направленность (профиль) «Технологии разработки программного обеспечения»**

**Квалификация выпускника: бакалавр
Форма обучения - очная**

Одобрена на заседании Ученого
совета института компьютерных наук и
технологического образования
Протокол от 23.05.2018 № 8

Санкт-Петербург

**Перечень планируемых результатов обучения по дисциплине**

Обучающийся должен обладать следующими компетенциями

Таблица 1

Код компетенции	Содержание компетенции
ОПК-2	способностью осваивать методики использования программных средств для решения практических задач
ПК-2	способностью разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования

Обучающийся должен**знать:**

- нормы, правила и способы использования программных средств для решения практических задач в типовых ситуациях;
- сложившиеся профессиональные, юридические и этические практики НОРМЫ работы с программными средствами; тенденции развития программных средств.
- основные синтаксические конструкции, основные типы данных и понятия языка Python (циклы, условные операторы, функции, модули, области видимости, основные типы аргументов функций и т.д.);
- основные модели данных языка Python, а также дискретные структуры (список, кортеж, словарь, множество), основные методы перебора и сортировки.
- основные стандарты оформления программного кода Python (PEP8);
- методику работы с файлами, основные принципы обработки данных, описание стандартных потоков ввода/вывода, методику работы с ними;
- основные принципы и техники тестирования;
- основные типы исключений и методы их обработки;
- правила применения декораторов и принципы их работы;
- основные синтаксические конструкции, необходимые для создания и работы в парадигме ООП;
- встроенные коллекции, специализированные типы данных, методы работы с ними;
- названия и функциональные возможности основных модулей и пакетов, предустановленных с Python;
- устройство итераторов, их протокол и семантику, методы работы с ними;

- основные шаблоны проектирования и ситуации для их применения;
- семантику и синтаксис, необходимые для построения графиков функций, основные принципы и языковые конструкции, основные графические модули и пакеты для работы с графикой;
- названия, предназначение и функционал основных модулей и пакетов, предназначенных для организации научных вычислений в Python;
- устройство генераторов и корутин, их протокол и семантику, методы работы с ними;
- семантику и синтаксис, необходимые для создания графических интерфейсов, основные принципы и языковые конструкции, основные графические модули и пакеты для работы с графикой;
- принципы использования виртуального окружения и методику работы с ним;
- основные понятия, синтаксические конструкции и методы для организации работы программы в режиме разделения потоков, более чем одного процесса;
- названия и функциональные возможности программных инструментов для оценки времени вычислений и его оптимизации;
- основные понятия, концепции, связанные с термином «фреймворк», а также синтаксис и семантику методов при его использовании;
- базовыми понятиями и концепциями, связанными с базами данных, а также синтаксис и семантику программных методов при работе с ними;

уметь:

- демонстрировать способности к адекватному выбору используемых методик и базовых программных средств для решения типовых практических задач;
- логически верно и грамотно строить устную и письменную речь при объяснении решения типовых задач;
- разрабатывать модели и алгоритмы и программно реализовывать их на ЭВМ;
- работать с современными системами программирования;
- производить тестирование и отладку программ.
- обоснованно использовать условные операторы, операторы циклов и модели и дискретные структуры данных и функции для работы с ними;
- применять программные средства для оформления программного кода согласно стандартам;
- грамотно оформлять функции, создаваемые пользователем, документировать и тестировать корректность их работы, использовать области видимости при создании переменных;
- применять функции для обработки строковых типов данных, для работы с потоками данных,

файлами;

- анализировать и обоснованно использовать программного обеспечения и встроенные механизмы для тестирования и отладки Python-сценариев;
- применять расширенный функционал обработки исключений;
- обоснованно использовать встроенные декораторы и создавать собственные декораторы;
- применять доступные языковые средства для создания объектно-ориентированных структур данных различного уровня сложности;
- исполнять методы для обработки данных во встроенных коллекциях и специализированных типах данных;
- использовать предустановленные с Python модули и пакеты, устанавливать новые с помощью pip;
- применять итераторы, синтаксические конструкции для работы с ними;
- обоснованно применять шаблоны проектирования;
- рационально применять методы для построения различных видов графиков функций, модулей и пакетов для работы с графикой;
- выполнять методы пакетов для научных вычислений;
- применять генераторы и корутины, синтаксические конструкции для работы с ними;
- использовать библиотеки их методы и структуры для создания графического интерфейса, его элементов;
- использовать инструментальные средства для создания и работы с виртуальным окружением;
- применять методы, модули и пакеты для организации многопоточных и многопроцессных вычислений;
- применять программные средства для профилирования программ, оптимизации вычислений;
- рационально применять методы, программные инструменты при использовании фреймворка;
- исполнять методы для подключения и работы с БД, реализовывать концепцию CRUD с помощью языковых средств;

владеть:

- приемами работы с типовыми программными средствами;
- способами решения типовых задач с использованием программных средств и пониманием важности качества итогового продукта;
- программными средствами для решения практических задач в конкретной предметной профессиональной области;
- способами самостоятельного освоения программных средств;
- современным инструментарием разработки ПО;



- навыками использования изученных структур данных и алгоритмов при решении прикладных задач;
- способами преобразования и обработки структур, данных, строк, файлов;
- основными программными средствами для написания и запуска программного кода на выполнение, а также его форматирования;
- способами грамотной передачи и извлечения аргументов функции различных типов и значений из различных областей видимости;
- способами обработки данных в потоках ввода, вывода, ошибок;
- навыками и способами тестирования реализованных решений с использованием различных библиотек, осваивать новые современные библиотеки для тестирования кода;
- языковыми средствами для создания программ, устойчивых к исключениям;
- навыками обработки вложенных исключений;
- навыками модификации работы функции с помощью декораторов, способами сохранения информации декорируемых функций;
- инструментами моделирования объектов бизнес-логики доступными языковыми средствами;
- способами обработки данных встроенных коллекций, специализированных типов данных, навыками адекватного использования соответствующих структуре языковых средств.
- навыками по созданию собственных пакетов, поиску модулей и пакетов, анализа возможностей содержимого, их установки и использования;
- навыками разработки собственных итераторов для различных структур данных;
- методикой использования актуальных моделей/паттернов при проектировании приложений (в т.ч. реализации объектно-ориентированных решений), их комбинированию;
- способностью к поиску и использованию современных графических библиотек сторонних разработчиков, решению с их помощью практических задач;
- навыками использования комбинации методов из различных пакетов для научных вычислений при решении комплексных задач;
- навыками разработки корутин (сопрограмм) и собственных генераторов для различных структур данных;
- методикой реализации кроссплатформенных интерфейсов с использованием возможностей современных графических пакетов;
- способностью к настройке сложных зависимостей различных пакетов в виртуальном окружении с учетом версий, совместимости, других особенностей;
- навыками оптимизации работы скриптов с использованием программных реализаций примитивов синхронизации, специализированных модулей и пакетов;



- навыками анализа и оптимизации скорости работы вычислительных процессов с использованием языковых программных инструментов;
- способностью использования возможностей фреймворков (например, Django) и расширений (например, requests, plywood) для создания веб-приложений, разработки API, REST-сервисов;
- методологией разработки приложений с БД.

Место дисциплины в структуре ОПОП: базовая часть

Объем (в зачетных единицах) и формы аттестации по дисциплине

Таблица 2

Дисциплина	Семестр	Всего зачетных единиц / из них на экзамен	Всего часов на теоретическое обучение	Контактная работа обучающихся с преподавателем	Самостоятельная работа	Лекционные занятия	Практические занятия	Лабораторные занятия	Формы аттестации
Б. 1.6 Модуль "Системное и прикладное программное обеспечение"		36	1296	576	720	162	54	360	
Б. 1.6.1 Программирование		21	756	324	432	108		216	
Б. 1.6.1 Программирование	2	4/1	144	54	90	18		36	экзамен
Б. 1.6.1 Программирование	3	3	108	54	54	18		36	
Б. 1.6.1 Программирование	4	3	108	54	54	18		36	
Б. 1.6.1 Программирование	5	3	108	54	54	18		36	
Б. 1.6.1 Программирование	6	4/1	144	54	90	18		36	экзамен
Б. 1.6.1 Программирование	7	4/1	144	54	90	18		36	экзамен
Б. 1.6.2 Базы данных	4	4	144	72	72	36		36	дифф. зачет
Б. 1.6.3 Инженерная и компьютерная графика		8	288	126	162		54	72	
Б. 1.6.3 Инженерная и компьютерная графика	5	4	144	54	90		18	36	
Б. 1.6.3 Инженерная и компьютерная графика	6	4/1	144	72	72		36	36	экзамен



Дисциплина	Семестр	Всего зачетных единиц / из них на экзамен	Всего часов на теоретическое обучение	Контактная работа обучающихся с преподавателем	Самостоятельная работа	Лекционные занятия	Практические занятия	Лабораторные занятия	Формы аттестации
Б. 1.6.4 Операционные системы	3	3	108	54	54	18		36	зачет

Содержание дисциплины с указанием разделов (тем) и часов по видам занятий, а также часов самостоятельной работы

Таблица 3

№ темы	Название темы с кратким содержанием	Виды занятий, часы			Самостоя тельная работа, часы	Всего часов
		Лекционн ые	Практиче ские	Лаборато рные		
Семестр 2						
1	Программирование. Язык СИ: общие сведения. Программирование. Задачи программирования. Языки программирования. Парадигмы программирования. Среды разработки. Стандарты Языка СИ. Общие сведения о языке программирования СИ: синтаксис, ключевые слова, переменные, базовые типы данных, операции, операторы. Структура программы. Консольный ввод и вывод. Стандартная библиотека. Заголовочные файлы. Препроцессор. Понятие функции. Примеры решения задач.	4		8	14	26
2	Управляющие инструкции. Условный оператор if. Вложенные условные операторы. Оператор выбора switch. Операторы цикла с предусловием while и с постусловием do- while. Оператор цикла с for. Примеры решения задач.	4		8	20	32
3	Массивы и указатели. Указатели. Операции с указателями. Использование указателей. Одномерный массивы. Двумерные	4		8	20	32



	массивы. Примеры решения задач.					
4	Функции. Создание функции. Локальные и глобальные переменные. Аргументы функции. Результат функции. Рекурсия. Указатель на функцию. Главная функция программы. Функции стандартной библиотеки. Примеры решения задач.	2		3	10	15
5	Функции для работы со строками и символами. Библиотечные функции для работы с символами. Функции и операции для работы со строками и символами. Символьные массивы. Примеры решения задач.	2		3	10	15
6	Графика. Текстовая и пиксельная графика. Библиотека graphics. Цвета. Графики. Текстовый режим и графический режим. Инициализация графики. Функции для работы с графикой. Структура графической программы. Примеры решения задач.	2		6	16	24
Семестр 3						
1	Введение в Python и основы синтаксиса. История языка, описание его возможностей, области применения. Создание скрипта на Python. Установка и настройка программного обеспечения, необходимого для работы. Переменные, типы данных (None, логические, числовые, строковые, изменяемые коллекции: списки, слайсы, множества, словари: неизменяемые типы данных: кортежи), операторы, создание комментариев, форматы данных, приведение форматов. Выражения: арифметические, отношений, логические, тернарный оператор. Условный оператор (if),	4		8	12	24



	работа с циклами (while, for in, range, slice). Zen of Python. PEP20.					
2	Понятие функции, объявление функций. Документирование функций. Аргументы функций, передача аргументов. Позиционные и именованные аргументы. Произвольное количество аргументов в функции. Упаковка и распаковка ключевых и позиционных аргументов функции. Значения, возвращаемые функцией. Присвоение значений. Расширенный синтаксис распаковки аргументов. Функции, определяемые пользователем, встроенные функции. Области видимости функций. LEGB. Global, nonlocal. Самовывзывающая функция. Рекурсия. PEP8. Функциональное программирование в Python. Функции высшего порядка. Лямбда-функция. Объявление. Использование. Функции, используемые совместно с лямбда-функциями: map, filter, zip. Генераторы списков. Вложенные генераторы. Генерация кортежей, словарей и множеств.	4		10	16	30
3	Строки. Байты. Файлы. Строки. Байты. Файлы. Особенности. Многострочные блоки текста. Конкатенация строк. Экранированные последовательности (табуляция, перенос строки). Юникод. Получение символа по коду, кода по символу. Выравнивание. Удаление символов. Разделение строки (split). Проверка на вхождение. Поиск подстроки. Замена подстроки. Предикаты. Форматирование строки. Спецификация формата. Преобразование объекта в строку. Форматирование строки: позиционные	4		8	12	24

	и ключевые аргументы. Файлы. Ввод/вывод. Методы работы с файлами. Стандартные потоки ввода/вывода.					
4	Тестирование в Python. Тестирование. Основные принципы, понятия. Типы тестов. Модульное тестирование (unit testing). Тестирование с помощью print. Тестирование с помощью оператора assert. Вывод сообщения об ошибке. Организация тестирования средствами библиотек на Python: Doctest. Unittest, пакет py.test. Директивы doctest. Тестовый случай. Основные виды тестовых утверждений unittest (assertTrue, assertFalse, assertEquals, assertIs). Автоматический поиск тестов. Подготовка контекста для тестов (fixtures). Запуск py.test из командной строки. Обзор лучших практик запуска тестов из командной строки. Возможности py.test (интроспекция, проверка на поднятие исключений, параметрические тесты, подготовка контекста, встроенные контексты). Сравнение методов тестирования. Тестирование нетривиальных случаев. Пакет hypothesis. Формулирование и проверка свойств. Типы генерируемых свойств.	6		10	14	30
Семестр 4						
1	Обработка исключений. Исключения. Типы. Иерархия. Встроенные исключения и исключения, определяемые пользователем. Базовый синтаксис для обработки исключений (try .. except), расширенный синтаксис (finally, else). Вложенные исключения. Исключения, поднимаемые при проверке утверждений или высказываний (assert). AssertionError. Поднятие исключений (raise). Менеджеры контекста. Протокол	4		8	12	24

	менеджеров контекста. Модуль contextlib.					
2	<p>Декораторы.</p> <p>Определение. Назначение. Примеры использования. Метод help. Функция-обертка. Внутренняя функция. Управление поведением функции. Трассировка функции. Декораторы с аргументами. Синтаксис создания. Примеры использования. Декораторы с опциональными аргументами. Практика использования декораторов. Контрактное программирование. Контракты @pre и @post. Цепочки декораторов.</p> <p>Модуль functools. Хранение определенного количества последних вызовов функции (lru_cache). Фиксация части позиционных и ключевых аргументов функции (partial). Обобщенные функции. Применение. Реализация обобщенной функции. Свертка (reduce).</p>	4		10	14	28
3	<p>Введение в ООП.</p> <p>Основные понятия ООП: абстракция, инкапсуляция, наследование, полиморфизм подтипов, класс, объект. Поля данных. Методы. Контроль доступа. Методы доступа. Свойства объекта, класса. Реализация ООП в Python. Определение и использование классов, типов. Атрибуты класса, атрибуты объекта. Методы и свойства, атрибуты класса, изменение свойств (getter, setter, deleter). Соккрытие атрибутов, методов. Внутренние атрибуты класса. Связанные и несвязанные методы. Наследование. Перегрузка методов и функция super(). Предикат isinstance(). Множественное наследование. Классы-примеси.</p>	6		10	14	30



	«Магические» методы. Методы, работающие с атрибутами. Операторы сравнения. Метод <code>__call__</code> . «Магические» методы для преобразования в строку.					
4	Встроенные коллекции. Встроенные коллекции: кортеж, список, словарь, множество (tuple, list, dict, set). Создание. Поиск. Слайсы. Именованные слайсы. Инверсия слайсов. Конкатенация кортежей. Именованные кортежи. Методы работы со списками. Добавление элементов. Конкатенация списков. Удаление элементов из списка. Инверсия. Сортировка. Модуль collections, его содержимое, примеры использования. Стек. Очередь. Множество. Операции с множествами (объединение, пересечение, разница, равенство, симметрическое разница). Сравнение множеств. Удаление элементов множества. Словарь. Конструирование. Обновление. Словарь с порядком по добавлению (orderdict). Счетчик (counter).	4		8	14	26
Семестр 5						
1	Модули. Пакеты. Основные типы модулей в Python. Стандартный модуль. Модули, предустановленные с Python. Модули, загражаемые извне. Обзор модулей из стандартной библиотеки: string, re, datetime, math, random, os, multiprocessing, subprocess, socket, email, json, doctest, unittest, pdb, argparse, sys. Модуль. Файл со скриптом на Python как модуль. Документирование модуля. Выполнение модуля. Импорт модуля (import). Импортирование отдельные	4		8	12	24

	<p>компоненты модуля (from ... import ...), импортирование всех компонент модуля (from ... import *). Пакеты. Понятие пакета. Обзор наиболее используемых пакетов в Python. Создание пользовательского пакета. Управление существующими пакетами. Pip. Виртуальное окружение. Публикация собственных пакетов.</p> <p>Относительный и абсолютный экспорт. Вложенные пакеты. Организация фасада (facade) для импорта модулей. Исполняемые модули и пакеты.</p> <p>Система импорта. Описание. Устранение циклического импорта. Пути поиска модулей и пакетов (sys.path). Построение sys.path. Реализация собственной системы импорта с помощью sys.path, sys.meta_path, sys.path_hooks.</p>					
2	<p>Итераторы.</p> <p>Итератор. Определение. Описание. Примеры. Алгоритм работы итератора. Область применения.</p> <p>Протокол работы итератора. Метод __iter__. Метод __next__. Итератор и итерабельный объект (iterable object). Операторы in и not in. Итераторы по умолчанию (__getitem__). Семантика упрощенного протокола итераторов. Итератор и генератор: сходства и различия. Модуль itertools, его содержимое, примеры использования</p>	4		8	12	24
3	<p>Введение в объектно-ориентированный анализ, шаблоны проектирования.</p> <p>Мотивация к использованию. Категории и их описание (поведенческие, структурные, порождающие). Примеры реализации паттернов на Python. Архитектурные шаблоны, характеристика, описание, примеры. Применение объектно-</p>	6		10	16	32



	ориентированных шаблонов, ошибки при использовании, графическое представление. Примеры применения паттернов при решении практических задач.					
4	Построение графиков на Python. Визуализация графиков функций (библиотека matplotlib). Виды 2D и 3D графиков, форматирование в Python. Экспорт графиков в виде изображений. QR-код. Генерация QR-кодов средствами Python, библиотека qrcode. Работа с векторной графикой, библиотека cairosvg. Стандартная библиотека для работы с изображениями: PIL – Python Image Library. Обзор возможностей, примеры использования.	4		10	14	28
Семестр 6						
1	Пакеты для научных вычислений (Scikit-learn, Pandas, Numpy). Обзор пакетов для научных вычислений и инженерных расчётов. Типы решаемых задач. Библиотека SciPy. Поиск минимумов и максимумов функций. Вычисление интегралов. Обработка изображений. Решение обыкновенных дифференциальных уравнений. Визуализация больших наборов данных с помощью графиков функций (библиотека matplotlib). Библиотека Scikit-learn. Обзор возможностей. Решение задач регрессии (линейная, полиномиальная). NumPy. Pandas. Операции с матрицами (сложение, вычитание, произведение, транспонирование, получение обратной матрицы).	6		8	24	38
2	Генераторы в Python. Корутины, сопрограммы. Определение. Описание. Примеры использования. Алгоритм работы генератора. Область применения.	4		8	24	36



	<p>Выбрасывание значений (yield). Интерфейс генераторов (send, throw, close).</p> <p>Сопрограммы (coroutines). Оператор yield from. Оператор return и StopIteration. Менеджеры контекста и генераторы: @contextmanager.</p> <p>Модуль itertools. Обобщение слайсов на произвольный итератор (islice). Бесконечные итераторы. Комбинаторные генераторы.</p>					
3	<p>Разработка графических интерфейсов(GUI).</p> <p>Пакеты Tkinter, PyQt. Основные элементы пакета. Окно верхнего уровня, заголовок окна. Рамка (Frame). Этикетка (Label). Поле ввода текста (Entry). Кнопка (Button). Переключатель (Radiobutton). Холст (Canvas). Обработка событий. Закрывание окна. Нажатие кнопки. Упаковщики окна (grid, place). Простейшая программа с графическим интерфейсом. Отрисовка графических примитивов. Овал. Линия, Прямоугольник. Обработка изображений (Pillow).</p>	4		8	24	36
4	<p>Виртуальное окружение (virtualenv, pipenv, pyenv).</p> <p>Мотивация к использованию. Классификация, сравнение. Обзор достоинств и недостатков решений. Стандартный пакет – virtualenv. Использование virtualenvwrapper. Установка virtualenv, создание, активация, деактивация изолированной области. Решение pipenv, установка, активация, использование, деактивация. Работа с pipfile. Менеджер версий — pyenv, основные команды. Примеры использования.</p>	4		12	18	34
Семестр 7						
1	<p>Многопоточность: потоки,</p>	6		8	24	38

	<p>процессы, очереди.</p> <p>Процесс. Поток. Характеристика. Особенности. Отличия. Демон. Мьютекс. Семафор.</p> <p>Модуль threading. Завершение работы потока (join).</p> <p>Примитивы синхронизации. Мьютекс (Lock). Рекурсивный мьютекс (RLock). Семафор (Semaphore). Связанный семафор (bounded semaphore). Интерфейс. Взаимодействие потоков друг с другом. Отправка сигналов между потоками (condition). Модуль "queue": FIFO, LIFO очередь («стэк»).</p> <p>Очередь с приоритетами (PriorityQueue). Работа с асинхронным кодом (модуль concurrent.futures). Параллелизм и конкурентность. Оценка производительности. GIL (Global interpreter lock). Многопроцессовое вычисление. Модуль multiprocessing. Параллельное вычисление независимых задач. Пакет joblib.</p>					
2	<p>Оптимизация и профилирование вычислений.</p> <p>Оценка времени вычисления (timeit). Профилирование. Модуль cProfile. Модуль line_profiler. Анализ программы на предмет оптимизации. Компиляция кода Python в машинный код (Numba). Расширение Python для компиляции в код на язык C (Cython). Режим аннотации Cython. Типизация.</p>	4		8	24	36
3	<p>Разработка с Python для веб.</p> <p>Использование фреймворка Django для построения веб-приложений. Настройка БД. Архитектура приложения на Django. Модель. Контроллер. Представление. Шаблон. Клиент-серверное взаимодействие. Создание асинхронных приложений с использованием AJAX и Python на сервере. Тестирование приложений на платформе Django.</p>	4		8	24	36



	Веб-фреймворк и библиотека для асинхронной сетевой работы Tornado. Микрофреймворк Flask. API, взаимодействие с API, использование API сторонних сервисов, создание собственного API. Понятие REST. Создание REST API, использование возможностей фреймворка Eve. Концепция MVC, её компоненты (модель, вид, контроллер)					
4	Взаимодействие с базами данных. Объектно-реляционное отображение (ORM): определение, мотивация использования, особенности применения, достоинства и недостатки. Концепция CRUD, реализация CRUD на Python, осуществление CRUD с помощью ORM. Примеры решений: Pony, Peewee. ORM Pony. Установка, подключение к БД, определение сущностей (entities), организация связей между сущностями / таблицами, создание ключей, составных ключей, отладка запросов, организация транзакций с помощью сессий. Работа с несколькими БД одновременно, управление параллелизмом. Агрегирующие функции. ORM Peewee: установка, подключение к Postgres, MySQL, SQLite, сортировка, фильтры данных, агрегирование, предварительная выборка. Взаимодействие Python-сценариев и MongoDB. Использование pymongo: подключение, реализация CRUD.	4		12	18	34
Итого:		108	-	216	432	756

Формы учебных занятий и интерактивные формы организации учебного процесса

Таблица 4

№	Интерактивные формы организации учебного процесса
---	---



темы	
Семестр 2	
1	Лекционные занятия: мультимедийные презентации с элементами взаимодействия, в том числе с использованием мобильных устройств. Лабораторные занятия: выполнение заданий в среде программирования CodeBlocks; работа в электронной информационной образовательной среде по дисциплине "Программирование", в том числе с использованием технологий электронного обучения (электронные учебные курсы в системе Moodle).
2	Лекционные занятия: мультимедийные презентации с элементами взаимодействия, в том числе с использованием мобильных устройств. Лабораторные занятия: выполнение заданий в среде программирования CodeBlocks; работа в электронной информационной образовательной среде по дисциплине "Программирование", в том числе с использованием технологий электронного обучения (электронные учебные курсы в системе Moodle).
3	Лекционные занятия: мультимедийные презентации с элементами взаимодействия, в том числе с использованием мобильных устройств. Лабораторные занятия: выполнение заданий в среде программирования CodeBlocks; работа в электронной информационной образовательной среде по дисциплине "Программирование", в том числе с использованием технологий электронного обучения (электронные учебные курсы в системе Moodle).
4	Лекционные занятия: мультимедийные презентации с элементами взаимодействия, в том числе с использованием мобильных устройств. Лабораторные занятия: выполнение заданий в среде программирования CodeBlocks; работа в электронной информационной образовательной среде по дисциплине "Программирование", в том числе с использованием технологий электронного обучения (электронные учебные курсы в системе Moodle).
5	Лекционные занятия: мультимедийные презентации с элементами взаимодействия, в том числе с использованием мобильных устройств. Лабораторные занятия: выполнение заданий в среде программирования CodeBlocks; работа в электронной информационной образовательной среде по дисциплине "Программирование", в том числе с использованием технологий электронного обучения (электронные учебные курсы в системе Moodle).
6	Лекционные занятия: мультимедийные презентации с элементами взаимодействия, в том числе с использованием мобильных устройств. Лабораторные занятия: выполнение заданий в среде программирования CodeBlocks; работа в электронной информационной образовательной среде по дисциплине "Программирование", в том числе с использованием технологий электронного обучения (электронные учебные курсы в системе Moodle).
Семестр 3	
1	Лекционные занятия: презентация с использованием видеофрагментов.



	<p>Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).</p>
2	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).</p>
3	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).</p>
4	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).</p>
Семестр 4	
1	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в</p>



	группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).
2	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).</p>
3	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).</p>
4	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).</p>
Семестр 5	
1	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).</p>
2	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного</p>



	<p>редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).</p>
3	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).</p>
4	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).</p>
Семестр 6	
1	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).</p>
2	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с</p>



	помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).
3	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).</p>
4	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).</p>
Семеср 7	
1	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).</p>
2	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода.</p> <p>Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).</p>
3	<p>Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации,</p>



	стандартов, посвященных написанию программного кода. Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).
4	Лекционные занятия: презентация с использованием видеофрагментов. Использование сети Интернет, использование инструментов совместного редактирования документов, демонстрация страниц официальной документации, стандартов, посвященных написанию программного кода. Лабораторные занятия: индивидуальная работа (сравнительный анализ понятий, сопоставление типов данных, операторов, их формальное описание), работа в группах (обмен мнениями, обсуждение, совместное редактирование информации с помощью соответствующих инструментов, в том числе редактирование кода в Git-репозиториях).

Содержание самостоятельной работы обучающихся по темам дисциплины

Содержание инвариантной самостоятельной работы обучающихся по темам

Таблица 5

№ темы	Содержание самостоятельной работы обучающихся	Количество часов
Семестр 2		
1	1.1. Знакомство с электронным учебным курсом «Программирование» и технологиями работы с ним.	4
	1.2. Выполнение заданий по теме «Простейшие типы данных. Функции ввода и вывода».	4
2	2.1. Выполнение заданий: разработка алгоритмов и компьютерных программ по теме «Управляющие инструкции»	14
3	3.1. Выполнение заданий: разработка алгоритмов и компьютерных программ по теме «Массивы и указатели»	14
4	4.1. Выполнение заданий: разработка алгоритмов и компьютерных программ по теме «Функции»	6
5	5.1. Выполнение заданий: разработка алгоритмов и компьютерных программ по теме «Функции работы со строками и символами»	6
6	6.1. Выполнение заданий: разработка алгоритмов и компьютерных программ по теме «Графика»	12
	Подготовка к экзамену	36



	Итого	60 + 36
Семестр 3		
1	<p>1.1. Установка и настройка среды для программирования на Python. Обзор IDE. Создание простейшей сценария по выводу текста на экран. Онлайн-интерпретаторы Python. Запись скринкаста с собственным комментарием и размещение его в портфолио, персональном репозитории.</p> <p>1.2. Разработка скрипта, вычисляющего сумму первых n-членов арифметической прогрессии (использование функций, условных операторов). Формирование отчета по выполнению задания и размещение его в портфолио, персональном репозитории.</p> <p>1.3. Разработка скрипта, позволяющего вычислить площадь треугольника с помощью формулы Герона. Формирование отчета по выполнению задания и размещение его в портфолио, персональном репозитории.</p> <p>1.4. Создание сценария, вычисляющего операции сложения, вычитания, умножения, деления для двух операндов. Формирование отчета по выполнению задания и размещение его в портфолио, персональном репозитории.</p>	<p>2</p> <p>2</p> <p>2</p> <p>2</p>
2	<p>2.1. Разработать скрипт с функцией, которая строит таблицу истинности для логического выражения (по вариантам) для двух и трех аргументов (используются различные наборы значений аргументов).</p> <p>2.2. Разработать программу, которая выводит на экран с помощью ASCII-графики таблицу истинности на основе переданных ей на вход аргументов (логическое выражение, аргументы, результат вычисления выражения). Формирование отчета по выполнению задания и размещение его в портфолио, персональном репозитории.</p> <p>2.3. Разработать скрипт с функцией, которая для ряда Фибоначчи, где количество элементов, $n = 22$, возвращает подмножество значений или единственное значение (по вариантам). Для нахождения элемента требуется использовать слайсы. Формирование отчета по выполнению задания и размещение его в портфолио, персональном репозитории.</p> <p>2.4. Напишите программу с функцией, в которой будет реализовано решение физической задачи (по вариантам). Например: ящик, имеющий форму куба с ребром a см без одной грани, нужно покрасить со всех сторон снаружи. Найдите площадь поверхности, которую необходимо покрасить. Ответ дайте в квадратных сантиметрах. Решение задачи оформите в виде функции <code>square(a)</code>, которая возвращает значение s. Например, при значении $a=30$, <code>square(30)</code> вернет $s = 4500$. Формирование отчета по выполнению задания и размещение его в портфолио, персональном репозитории.</p>	<p>3</p> <p>3</p> <p>2</p> <p>2</p>
3	<p>3.1. Создание аннотированного списка библиотек для работы с текстом в Python. Формирование отчета по выполнению задания и размещение его в портфолио, персональном репозитории.</p> <p>3.2. Разработка сценария с реализацией операции поиска подстроки в тексте.</p>	<p>2</p> <p>2</p> <p>2</p>

	3.3. Создание скрипта для считывания данных справочных логов из текстового файла и преобразования их в CSV-формат с последующей записью в новый файл. Формирование отчета по выполнению задания и размещение его в портфолио, персональном репозитории. 3.4. Реализовать программу шифрующую строку, задаваемую пользователем, с помощью алгоритма шифрования ROT13. Формирование отчета по выполнению задания и размещение его в портфолио, персональном репозитории.	2
4	4.1. Разработать программу для считывания данных JSON-формата из файла и вывод их в табличном виде на экран. Организовать тестирование работоспособности программы с помощью assert, print. 4.2. Дополнение программы задания 4.1 (считывание данных JSON-формата) тестами с использованием библиотеки doctest. 4.3. Дополнение программы задания 4.1,4.2 (считывание данных JSON-формата) тестами с использованием пакета py.test. 4.4. Формирование отчета по самостоятельно работе и публикация его в портфолио	3 2 3 2
	Итого	36
Семестр 4		
1	1.1. Разработать программу с реализацией функции для считывания json-данных из файла и вывод их в табличном виде на экран. Реализовать базовый синтаксис для обработки исключений (try .. except) 1.2. Дополнение программы для считывания данных проверкой утверждений или высказываний (assert). Создание отдельного блока для такой проверки (с помощью __name__) и скрипта командной строки для запуска этих проверок. 1.3. Дополнение программы для считывания данных с использованием менеджера контекстов и реализации расширенного синтаксиса для обработки исключений. 1.4. Формирование отчета по практическому заданию и публикация его в портфолио.	2 2 2 2
2	2.1 Разработать прототип программы «Калькулятор», позволяющую выполнять базовые арифметические действия и функцию обертку, сохраняющую название выполняемой операции, аргументы и результат в файл 2.2 Дополнение программы «Калькулятор» декоратором, сохраняющий действия, которые выполняются в файл-журнал. 2.3 Рефакторинг (модификация) программы с декоратором модулем functools и использование его функционала 2.4 Формирование отчета по практическому заданию и публикация его в портфолио.	2 2 2 2
3	2.1 Разработка классов и объектов «запись», «комментарий» для приложения	4



	«Блог» (использование наследования). 2.2. Создание геттеров и сеттеров для классов «запись», «комментарий» приложения «Гостевая книга». Создание функций для вывода на печать информации, хранящийся в объектах. 2.3. Формирование отчета по практическому заданию и публикация его в портфолио.	4 2
4	3.1 Создание программы по заполнению массивов случайными значениями. Сортировка значений в списке методом вставки, плавной сортировки, с помощью встроенных функций языка. 3.2 Создание программы по распределению списка с случайными значениями на два списка по определенному критерию (четность/нечетность, положительные/отрицательные числа). 3.4. Формирование отчета по практическому заданию и публикация его в портфолио.	4 4 2
	Итого	36
Семестр 5		
1	1.1 Исследовать функционал одного из модулей стандартной библиотеки (string, re, datetime, math, random, os, и т.д.) и, используя инструмент Jupyter Notebook, создать документ с описанием и примерами использования его функционала. Опубликовать его в портфолио. 1.2 Создание пользовательского пакета для приложения «Гостевая книга» с прототипами методов, позволяющих взаимодействовать с JSON-файлом (создание, удаление, переименование, чтение, запись). Формирование отчета по практическому заданию и публикация его в портфолио.	4 4
2	2.1 Разработать функцию, возвращающую элементы ряда Фибоначчи по данному максимальному значению. 2.2 Создание программы, возвращающей список чисел Фибоначчи с помощью итератора. 2.3 Формирование отчета по практическому заданию и публикация его в портфолио.	2 4 2
3	3.1 Разработать фрагмент программы, позволяющий получать данные о текущих курсах валют с сайта Центробанка РФ с использованием сервиса, который они предоставляют. Применить шаблон проектирования «Одиночка» для предотвращения отправки избыточных запросов к серверу ЦБ РФ. Оформить решение в виде корректно работающего приложения, реализовать тестирование и опубликовать его в портфолио. 3.2 На основе фрагмента программы, предложенного преподавателем, реализовать класс для получения данных с сайта Центробанка РФ с использованием сервиса, который они предоставляют. Применить шаблон проектирования «Декоратор» для реализации функционала, позволяющего преобразовывать данные о курсах валют в формат JSON. Реализовать	4 4



	сохранение (сериализацию) данных в файл формата JSON. 3.3. Создание ЭОР на тему «Обзор современных фреймворков, реализующих шаблон архитектуры системы MVC», создание сравнительной таблицы 3-5 фреймворков.	2
4	4.1 Используя свободные источники (bn.ru, avito.ru и т.д.), собрать данные о ценах на недвижимость, выставленную на продажу в разных районах города. Преобразовать данные в формат csv. Разработать скрипт для визуализации данных, используя библиотеку matplotlib. Для визуализации использовать тип “точечная диаграмма” (scatterplot). 4.2 Разработать фрагмент программы с использованием библиотеки pyqrcode, позволяющей создавать изображение QR-кода на основе переданной в программу текстовой строки. 4.3 Реализовать модификацию изображения генерируемого QR-кода: раскрасить фрагменты изображения в несколько случайно определяемых цветов.	4 4 2
	Итого	36
Семестр 6		
1	1.1 Разработка скрипта, вычисляющего статистические показатели (среднее значение, дисперсия, среднее квадратичное отклонение) для данных, считанных из CSV-файла. 1.2 Осуществить рефакторинг (модификация) скрипта, вычисляющего статистические показатели для данных, считанных из CSV, с использованием библиотеки научных вычислений numpy. 1.3. На основе данных, предоставленных преподавателем, реализовать отображение данных на точечной диаграмме с помощью библиотеки matplotlib. Создать модель (квадратичная функция) для предсказания новых данных и нанести график этой функции на точечную диаграмму. Вычислить отклонение данных модели от реальных данных. 1.4 Формирование отчета по выполненной самостоятельной работе и публикация его в портфолио.	4 4 6 2
2	2.1 На основе кода, предоставленного преподавателем, реализовать генератор чисел ряда Фибоначчи. Генератор требуется создать двумя вариантами: с помощью генератора списков, с помощью функции, внутри которой yield. 2.2 Разработать программу, позволяющую генерировать уникальные идентификаторы: UUID (universally unique identifier). Структура UUID — на усмотрение студента. 2.3 На основе кода, предоставленного преподавателем, реализовать корутину, позволяющую использовав метод send() для возврата генерируемой сущности. В основе корутины должен использоваться принцип блокчейна (цепочки блоков). Кроме механизма возврата нового блока требуется создать механизм, позволяющий вернуть историю сгенерированных блоков.	4 4 6 2



	2.4 Формирование отчета по выполненной самостоятельной работе и публикация его в портфолио.	
3	3.1 Разработать программу, позволяющую решать квадратное уравнение через вычисление дискриминанта. В программе должен быть предусмотрен ввод значений коэффициентов a , b , c пользователем. Требуется протестировать программу с помощью одной из специальных библиотек. Сформировать отчет по выполненной самостоятельной работе и опубликовать его в портфолио.	6
	3.1 Разработать программу, позволяющую решать систему уравнений. Программа должна позволять вводить коэффициенты при неизвестных, а также должна учитывать возможность несовместного решения системы. Графический интерфейс реализовать с помощью PyQt или TKinter. Требуется протестировать программу с помощью библиотеки. Сформировать отчет по выполненной самостоятельной работе и опубликовать его в портфолио.	6
	3.2 Формирование отчета по выполненной самостоятельной работе и публикация его в портфолио.	4
4	4.1 На основе материалов преподавателя, рекомендованной литературы, а также ресурсов сети интернет (видеозаписи выступления по теме на конференциях) разработать конспект с примерами по использованию различных систем виртуализации проектов на Python. Требуется рассмотреть <code>virtualenv</code> , <code>pipfile</code> , <code>ruenv</code> , <code>pipenv</code> , <code>venv</code> . Привести примеры с разбором конфигурационных файлов 2 систем.	8
	4.2 Создать глоссарий команд, используемых для создания виртуального окружения для одной из описанных систем. Записать скринкаст об использовании этой системы с демонстрацией основных этапов работы с окружением (развертывание, установка, обновление, модификация версий пакетов и их удаление).	4
	Подготовка к экзамену	36
	Итого	60+ 36
Семестр 7		
1	1.1 На основе кода, предложенного преподавателем, создать программу, позволяющую вычислить произведение двух матриц, не использовать при этом сторонние библиотеки, проверить размерность матриц и корректно обрабатывать эту ситуацию. Размерность и значения элементов матриц пользователь вводит с клавиатуры. Организовать тестирование получившегося решения.	8
	1.2 Выполнить рефакторинг кода программы, вычисляющей произведение матриц, реализовав функционал, позволяющий использовать механизм потоков для выполнения вычислений.	10
	1.3 Формирование отчета по практическому заданию и публикация его в портфолио.	2

2	2.1 На основе кода, предоставленного преподавателем, реализовать программу, позволяющую записывать данные в файл (используйте файл формата CSV) в несколько процессов. Для реализации используйте менеджер контекста. Оцените скорость выполнения кода.	6
	2.2 Осуществите рефакторинг кода так, чтобы программа позволяла записывать данные в файл (используйте файл формата CSV) в несколько потоков. Оцените скорость выполнения кода.	4
	2.3 Формирование отчета по практическому заданию, сравните быстродействие полученных решений. Опубликуйте отчет по заданию в портфолио.	2
3	3.1 На основе кода, предложенного разработчиками фреймворка Flask, используя официальной документации, разработать простейшее клиент-серверное веб-приложение, позволяющее сохранять данные из формы со страницы, взаимодействовать с базой данных и сохранять в неё информацию, вводимую пользователем.	6
	3.2 Разработать фрагмент программы, реализующей программу чат с помощью фреймворка Tornado. Для отправки сообщений использовать механизм сокетов.	8
	3.3 Формирование отчета по заданию и публикация его в портфолио.	2
4	4.1 Разработайте фрагмент приложения с использованием фреймворка Django и его ORM инструментов, реализующей веб-приложение «Блог». Приложение должно позволять добавлять, удалять и редактировать данные. Должна быть реализована концепция CRUD.	10
	4.2 Формирование отчета по заданию и публикация его в портфолио.	2
	Подготовка к экзамену	36
	Итого:	60+ 36

Содержание вариативной составляющей самостоятельной работы

Таблица 6

№ темы	Содержание самостоятельной работы обучающихся	Количество часов
Семестр 2		
1	1.1. Регистрация в электронных библиотечных системах и поиск учебных изданий по программированию на Си в их каталогах.	3
	1.1. Поиск информационных ресурсов по дисциплине «Программирование на Си» в электронных библиотечных системах	0
	1.2. Поиск полнотекстовых ресурсов по вопросам программирования и языка программирования СИ в Интернет. Формирование Хрестоматии по учебной	3

	дисциплине. 1.2 Поиск видео ресурсов по вопросам программирования и языка программирования СИ в Интернет. Формирование Хрестоматии по учебной дисциплине.	0
2	2.1. Выполнение вариативных заданий по выбору студента (Пример типового вариативного задания в Приложении 1). 2.1. Выполнение вариативных заданий по выбору студента (Пример типового вариативного задания в Приложении 1).	6 0
3	3.1. Выполнение вариативных заданий по выбору студента (Пример типового вариативного задания в Приложении 1). 3.1. Выполнение вариативных заданий по выбору студента (Пример типового вариативного задания в Приложении 1).	6 0
4	4.1. Выполнение вариативных заданий по выбору студента (Пример типового вариативного задания в Приложении 1). 4.1. Выполнение вариативных заданий по выбору студента (Пример типового вариативного задания в Приложении 1).	4 0
5	5.1. Выполнение вариативных заданий по выбору студента (Пример типового вариативного задания в Приложении 1). 5.1. Выполнение вариативных заданий по выбору студента (Пример типового вариативного задания в Приложении 1).	4 0
6	6.1 Выполнение вариативных заданий по выбору студента (Пример типового вариативного задания в Приложении 1). 6.1. Выполнение вариативных заданий по выбору студента (Пример типового вариативного задания в Приложении 1).	4 0
	Итого:	30
Семестр 3		
1	1.1. Создание ЭОР на тему «Развертывание и настройка среды для разработки на Python» 1.2. Создание ЭОР на тему «Использование платформы IPython для решения научных и исследовательских задач» 1.3. Создание таблицы со сравнительным анализом REPL-сред для написания программ на языке Python.	4 0 0
2	2.1. Исследовать способы проверки программного кода Python на совместимость со стандартом PEP8. Составить сравнительную таблицу с анализом. Сформировать отчет, опубликовать отчет и таблицу в портфолио. 2.2. Исследовать способы преобразования программного кода Python в соответствии со стандартом PEP8. Составить сравнительную таблицу с анализом. Сформировать отчет, опубликовать отчет и таблицу в портфолио. 2.3. Разработать программу, которая для заданного количества значений возвращала бы список из уникальных элементов, содержащихся во входном	2 0 2

	<p>наборе значений. Используйте упаковку и распаковку элементов.</p> <p>2.4. Разработать программу, которая для заданного количества значений возвращала бы список из повторяющихся элементов, содержащихся во входном наборе значений. Используйте упаковку и распаковку элементов.</p> <p>2.5. Реализуйте программу с реализацией работы функции zip через функцию map.</p> <p>2.6. Перепишите лямбда-функцию, генерирующую квадраты чисел из переменной типа list, через генератор списка.</p> <p>2.7. Перепишите генератор списка, позволяющий получить модуль числа, через лямбда-функцию.</p> <p>2.8. Создайте программу, реализующие одноместные, двухместные, трехместные арифметические действия (сложение, вычитание, умножение, деление) с помощью лямбда-функций.</p>	<p>0</p> <p>2</p> <p>0</p> <p>0</p> <p>0</p>
3	<p>3.1. Разработать программу, которая выводит на экран с помощью ASCII-графики таблицу истинности на основе переданных ей на вход аргументов (логическое выражение, аргументы, результат вычисления выражения). Для вывода на экран информации использовать метод format.</p> <p>3.2. Реализовать программу-игру «Угадай число», в которой для вывода на экран информации использовать метод format.</p> <p>3.3. Написать программу, выводящую на экран последовательно символы английского и кириллического алфавита с использованием кодов из таблицы unicode-символов.</p> <p>3.4. Реализовать программу шифрующую строку, задаваемую пользователем, с помощью алгоритма шифрования, использующего сдвиг на определенное количество знаков (шифр Цезаря). Сдвиг задается пользователем</p>	<p>2</p> <p>0</p> <p>2</p> <p>0</p>
4	<p>4.1. Создание таблицы со сравнительным анализом библиотек для тестирования. Формирование отчета по выполнению задания и размещение его в портфолио, персональном репозитории.</p> <p>4.2. Подготовить набор тестов для модульного тестирования разработанной игры «Крестики-нолики». Проверить работоспособность функционала по сохранению результатов игры, загрузке конфигурационного файла, проверке выигрышной ситуации. Формирование отчета по выполнению задания и размещение его в портфолио, персональном репозитории.</p> <p>4.3. Разработать программу для считывания данных из JSON-файла и вывода их в табличном виде на экран и протестировать работоспособность с использованием unittest. Формирование отчета по выполнению задания и размещение его в портфолио, персональном репозитории.</p> <p>4.4. Написать программу для вычисления факториала натурального числа от 0 до n (где, n — целое, натуральное число, помещающееся в переменную целого числа). Для всех других случаев функция должна поднимать исключение TypeError, ValueError. Протестировать работу этой программы с использованием unittest. Учтите ситуации, для которых должны подниматься исключения. Формирование отчета по выполнению задания и размещение его</p>	<p>2</p> <p>0</p> <p>2</p> <p>0</p>



	в портфолио, персональном репозитории.	
	Итого:	18
Семестр 4		
1	1.1 Разработать фрагмент программы, который будет сохранять вводимые пользователем данные, по выбору в json, или csv-файле (использовать модули csv, json) с использованием протокола менеджеров контекста, а также расширенного синтаксиса исключений. 1.2 Разработать фрагмент программы, сериализующей вводимые пользователем данные, в базе данных sqlite (использовать модуль sqlite3) с использованием расширенного синтаксиса исключений, 1.3 Создание программы для считывания данных формата CSV с использованием функционала модуля contextlib.	4 0 0
2	2.1. Разработка фрагмента веб-приложения, позволяющего фиксировать в журнале (текстовом файле) действия пользователя. 2.2 Разработка фрагмента веб-приложения, позволяющего осуществлять проверку авторизации пользователя. 2.3 Разработка функции-декоратора, вычисляющей время выполнения декорируемой функции. 2.4 Разработка функции-декоратора, позволяющей выполнять декорируемую функцию единожды.	4 0 0 0
3	3.1 Разработка прототипа приложения "Регистрация на конференцию" на основе фрагмента технического задания с использованием ООП. 3.2 Разработка прототипа приложения "Калькулятор", реализующего паттерн MVC (Model View Controller). 3.3 Разработка прототипа приложения «Гостевая книга» с авторизацией с помощью механизма OAuth2, OpenID или API VK, Facebook). 3.4 Разработка скрипта для получения и сохранения данных социальных сетей Twitter или Instagram.	2 0 4 0
4	3.1 Создание программы с реализацией вручную одного из алгоритмов сортировки (вставки, плавной сортировки). 3.2 Создание программы по разделению одного словаря на произвольное количество словарей по определенному критерию, задаваемому в виде лямбда функции. 3.3 Создание программы, позволяющей выполнять основные операции (объединение, пересечение, вычитание) над множествами (количество множеств и их элементы вводятся вручную).	4 0 0
	Итого:	18
Семестр 5		
1	1.1 Исследовать функционал одного модуля не из стандартной библиотеки	4



	<p>(например, joblib) и создать фрагмент ЭОР с описанием и примерами его использования при работе в Jupyter Notebook и в скриптах. Для выполнения задания использовать Jupyter Notebook, опубликовать результат выполнения задания в портфолио в HTML и PDF формате.</p> <p>1.2 Создание пользовательского пакета с организованным в нем механизмом «фасад» для импорта модулей. Публикация собственного пакета в общем репозитории пакетов рурі.</p> <p>1.3 Реализация скрипта, позволяющего выполнять импорт удаленных файлов по URL. Запись скринкаста с демонстрацией работы скрипта. Формирование отчета по практическому заданию и публикация его в портфолио.</p>	0 0
2	<p>2.1 Разработать функцию, возвращающую список чисел ряда Фибоначчи с использованием бесконечных итераторов (модуль itertools).</p> <p>2.2 На основе предоставленной программы-чата разработать функцию, позволяющую вычислять количество реплик, обращенных к участникам в журнале чата. Статистику представить в виде «имя пользователя»: «количество фраз». Посчитать среднее число символов, приходящееся на одного участника.</p>	4 0
3	<p>3.1 Разработать фрагмент электронного ресурса, посвященного шаблону проектирования «Декоратор». Привести пример фрагмента кода, реализующего указанный шаблон, протестировать его работоспособность. Оформить задание в виде корректно работающего приложения и опубликовать его в портфолио.</p> <p>3.2 Разработать фрагмент электронного ресурса, посвященного шаблону проектирования «Адаптер». Привести пример фрагмента кода, реализующего указанный шаблон, протестировать его работоспособность. Оформить задание в виде корректно работающего приложения и опубликовать его в портфолио.</p> <p>3.3 Разработать фрагмент электронного ресурса, посвященного шаблону проектирования «Абстрактная фабрика». Привести пример фрагмента кода, реализующего указанный шаблон, протестировать его работоспособность. Оформить задание в виде корректно работающего приложения и опубликовать его в портфолио.</p> <p>3.4 Разработать фрагмент электронного ресурса, посвященного шаблону проектирования «Заместитель». Привести пример фрагмента кода, реализующего указанный шаблон, протестировать его работоспособность. Оформить задание в виде корректно работающего приложения и опубликовать его в портфолио.</p> <p>3.5 Разработать фрагмент электронного ресурса, посвященного шаблону проектирования «Мост». Привести пример фрагмента кода, реализующего указанный шаблон, протестировать его работоспособность. Оформить задание в виде корректно работающего приложения и опубликовать его в портфолио.</p>	6 0 0 0 0

4	4.1 На основе кода, позволяющего визуализировать данные о ценах на недвижимость (точечная диаграмма), отобразить с помощью библиотеки matplotlib линейный график и график полинома второй степени (квадратичный) соответствующий изменениям цен на недвижимость.	4
	4.2 На основе кода, позволяющего визуализировать данные о ценах на недвижимость (точечная диаграмма), отобразить с помощью библиотеки matplotlib полиномиальный график (степеней полинома 3, 4, 10) изменений цен на недвижимость.	0
	4.3 На основе кода, позволяющего визуализировать данные о ценах на недвижимость (точечная диаграмма), отобразить с помощью библиотеки matplotlib линейные графики изменений цен на недвижимость, разбив всё множество данных на 3 подмножества.	0
Итого:		18
Семестр 6		
1	2.1 Разработка скрипта, вычисляющего произведение матриц произвольной размерности с использованием Cython. Замер времени вычисления. Создание отчета по результатам анализа производительности.	8
	2.2 Разработка скрипта, вычисляющего произведение матриц произвольной размерности с использованием Numba и замером времени вычисления. Создание отчета по результатам анализа производительности.	0
	2.3 Разработка скрипта, вычисляющего произведение матриц произвольной размерности с использованием библиотеки numru и замер времени вычисления. Создание отчета по результатам анализа производительности.	0
2	2.1 Написать программу, позволяющую выполнять подсчет слов в тексте, а также вычислять размер (в символах) каждого слова. Используйте для возвращения результатов подсчета механизм генераторов. Решение сопроводить тестами и опубликовать в портфолио.	8
	2.2 Написать программу, позволяющую выполнять проверку свойства парности скобок в строке текста, а также вычислять их количество. Используйте для возвращения результатов подсчета механизм генераторов. Решение сопроводить тестами и опубликовать в портфолио.	0
	2.3 Создать опорный конспект лекции Дэвида Бизли (dabeaz.com/coroutines/), посвященной сопрограммам (корутинам). Оформить конспект средствами IPython Notebook и опубликовать в портфолио.	0
3	3.1 Реализация графического интерфейса и формы для приложения «Гостевая книга» с возможностью сохранения данных из полей формы в файл. Сформировать отчет по выполненной самостоятельной работе и опубликовать его в портфолио.	8
	3.2 Реализация графического интерфейса программы, позволяющего создавать изображения флагов и сохранять их в формате svg с помощью библиотеки svgwrite. Сформировать отчет по выполненной самостоятельной	0



	работе и опубликовать его в портфолио. 3.3 Реализация графического интерфейса и функционала, позволяющего отображать графические примитивы для игры «Крестики-нолики». Сформировать отчет по выполненной самостоятельной работе и опубликовать его в портфолио.	0
4	4.1 Написать программу, в которой пользователь вводит число от 0 до 9 включительно, а программа выводит название введенного числа, а если второй входной аргумент type имеет значение bin, oct, hex, то функция преобразует это число в бинарную, восьмеричную или шестнадцатеричную форму. Предусмотреть проверку корректности введенного пользователем значения. При реализации используемые библиотеки должны находиться в виртуальном окружении (использовать virtualenv). 4.2 Написать программу, в которой пользователь вводит число от 0 до 9 включительно, а программа выводит название введенного числа, а если второй входной аргумент type имеет значение bin, oct, hex, то функция преобразует это число в бинарную, восьмеричную или шестнадцатеричную форму. Предусмотреть проверку корректности введенного пользователем значения. При реализации используемые библиотеки должны находиться в виртуальном окружении (использовать pip).	6 0
	Итого:	30
Семестр 7		
1	1.1 Создать фрагмент электронного образовательного ресурса «Примитивы синхронизации в Python». Привести примеры программного кода, указать типичные ситуации, в которых их применение будет уместно. Публикация в портфолио. 1.2 На основе рекомендованной литературы, материалов сети интернет (выступления на конференциях, митапах и т. д.) Создать опорный конспект по теме асинхронного программирования в Python, его достоинствах и недостатках. Опубликовать в портфолио 1.3 Создать сравнительную таблицу для двух подходов оптимизации вычислений: с помощью потоков и процессов. Указать достоинства и недостатки. Публикация в портфолио.	4 0 0
2	2.1 Создание программы, позволяющей вычислить произведение двух матриц в несколько процессов, количество процессов должно задаваться как параметр и оценка производительности с помощью %%timeit. Публикация отчета с анализом вычисления. 2.2 Создание программы, вычисляющей произведение двух матриц в несколько процессов или потоков с оптимизацией вычислений на Cython, оценка производительности с помощью %%cython и режима аннотации. Публикация отчета с анализом вычисления. 2.3 Создание программы, вычисляющей произведение двух матриц в несколько потоков с использованием пакета joblib, оценка	12 0 0



	производительности. Публикация отчета с анализом вычисления.	
3	3.1 На основе кода, используя официальную документацию RESTful фреймворка Eve, разработать модели для основных сущностей, необходимых для веб-приложения «Система голосования» (пользователь, опрос). Хранение данных организовать в SQLite базе данных, используя библиотеку sqlalchemy. 3.2 На основе кода, реализующего программу чат с помощью фреймворка Tornado, разработать функционал отправки сообщения лично пользователю чата, при этом не должна происходить широковещательная отправка сообщений всем пользователям.	8 0
4	4.1 На основе кода, используя официальную документацию RESTful ORM Eve, разработать модели для основных сущностей, необходимых для веб-приложения «Система голосования» (пользователь, опрос). Хранение данных организовать в SQLite базе данных, используя библиотеку sqlalchemy. 4.2 На основе кода, используя официальную документацию RESTful ORM Pony, разработать модели для основных сущностей, необходимых для веб-приложения «Гостевая книга» (пользователь, опрос). Хранение данных организовать в SQLite базе данных, используя библиотеку sqlalchemy. 4.3 На основе кода, используя официальную документацию RESTful ORM Peewee, разработать модели для основных сущностей, необходимых для веб-приложения «Блог» (пользователь, опрос). Хранение данных организовать в SQLite базе данных, используя библиотеку sqlalchemy.	6 0 0
	Итого:	30

Учебно-методическое обеспечение для самостоятельной работы обучающихся по дисциплине

Таблица 7

№ п/п	Наименование учебников, учебно-методических, методических пособий, разработок и рекомендаций
1	Перечень основной и дополнительной литературы
2	Электронные учебные курсы в системе дистанционного обучения Moodle: <ul style="list-style-type: none">• https://moodle.herzen.spb.ru/course/view.php?id=1490• https://moodle.herzen.spb.ru/course/view.php?id=3731

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Этапы формирования компетенций в процессе освоения ОПОП

Таблица 8

Код Компетенции	1 - этап*	2 - этап*	3 - этап*
ОПК-2	Б. 1.4 Модуль "Естественно-математический", Б. 1.6.1 Модуль "Системное и прикладное программное обеспечение". Программирование , Б. 1.7 Модуль "Дискретные структуры", Б. 1.8 Модуль "Технологии и методы вычислений", Б. 1.9 Модуль "Информационные технологии в математике и физике"	Б. 1.6 Модуль "Системное и прикладное программное обеспечение", Б. 1.6.1 Модуль "Системное и прикладное программное обеспечение". Программирование , Б. 1.12 Модуль "Информационные технологии"	Б. 1.6.1 Модуль "Системное и прикладное программное обеспечение". Программирование , Б. 1.11 Модуль "Информационные технологии в управлении в IT-компаниях", Б. 1.15 Модуль "Информационные технологии и системы", Б. 1.16 Модуль "Учебно-исследовательский", Б. 2.2 Производственная практика (технологическая)
ПК-2	Б. 1.5 Модуль "Организация ЭВМ", Б. 1.6 Модуль "Системное и прикладное программное обеспечение", Б. 1.6.1 Модуль "Системное и прикладное программное обеспечение". Программирование , Б. 1.8 Модуль "Технологии и методы вычислений", Б. 1.9 Модуль "Информационные технологии в математике и физике", Б. 1.10 Модуль "Проектирование и разработка веб-решений"	Б. 1.6.1 Модуль "Системное и прикладное программное обеспечение". Программирование , Б. 1.12 Модуль "Информационные технологии", Б. 1.14 Модуль "Информационные ресурсы и средства профессиональной деятельности инженера"	Б. 1.6.1 Модуль "Системное и прикладное программное обеспечение". Программирование , Б. 1.14 Модуль "Информационные ресурсы и средства профессиональной деятельности инженера", Б. 1.15 Модуль "Информационные технологии и системы", Б. 1.16 Модуль "Учебно-исследовательский", Б. 2.2 Производственная практика (технологическая)

*Примечание: 1 этап - 1, 2 курсы; 2 этап - 3 курс; 3 этап - 4 курс

Уровни формирования компетенции

Таблица 9

Код компетенции	Знает	Умеет	Владеет
ОПК-2	<ul style="list-style-type: none"> нормы, правила и способы использования программных средств для решения практических задач в типовых ситуациях; 	<ul style="list-style-type: none"> демонстрировать способности к адекватному выбору используемых методик и базовых программных средств для решения типовых 	<ul style="list-style-type: none"> приемами работы с типовыми программными средствами; навыками решения типовых задач с использованием программных

	<ul style="list-style-type: none"> • сложившиеся профессиональные, юридические и этические практики НОРМЫ работы с программными средствами; тенденции развития программных средств; • основные стандарты оформления программного кода Python (PEP8); • методику работы с файлами, основные принципы обработки данных, описание стандартных потоков ввода/вывода, методику работы с ними; • правила применения декораторов и принципы их работы; • основные синтаксические конструкции, необходимые для создания и работы в парадигме ООП; • названия и функциональные возможности основных модулей 	<p>практических задач;</p> <ul style="list-style-type: none"> • логически верно и грамотно строить устную и письменную речь при объяснении решения типовых задач; • применять программные средства для оформления программного кода согласно стандартам; • применять функции для обработки строковых типов данных, для работы с потоками данных, файлами; • обоснованно использовать встроенные декораторы и создавать собственные декораторы; • применять доступные языковые средства для создания объектно-ориентированных структур данных различного уровня сложности; • использовать предустановленные с Python модули и пакеты, устанавливать 	<p>средств и пониманием важности качества итогового продукта;</p> <ul style="list-style-type: none"> • навыками самостоятельного освоения программных средств. • основными программными средствами для написания и запуска программного кода на выполнение, а также его форматирования; • способами преобразования и обработки структур, данных, строк, файлов; • навыками модификации работы функции с помощью декораторов, способами сохранения информации декорируемых функций; • инструментами моделирования объектов бизнес-логики доступными языковыми средствами; • навыками по созданию
--	--	--	---

	<p>и пакетов, предустановленных с Python;</p> <ul style="list-style-type: none"> • устройство итераторов, их протокол и семантику, методы работы с ними; • устройство генераторов и корутин, их протокол и семантику, методы работы с ними; • семантику и синтаксис, необходимые для создания графических интерфейсов, основные принципы и языковые конструкции, основные графические модули и пакеты для работы с графикой; • основные понятия, синтаксические конструкции и методы для организации работы программы в режиме разделения потоков, более чем одного процесса; • названия и функциональные возможности программных 	<p>новые с помощью <code>pip</code>;</p> <ul style="list-style-type: none"> • применять итераторы, синтаксические конструкции для работы с ними; • применять генераторы и корутины, синтаксические конструкции для работы с ними; • использовать библиотеки их методы и структуры для создания графического интерфейса, его элементов; • применять методы, модули и пакеты для организации многопоточных и многопроцессных вычислений; • применять программные средства для профилирования программ, оптимизации вычислений; • рационально применять методы, программные инструменты при использовании фреймворка; • исполнять методы 	<p>собственных пакетов, поиску модулей и пакетов, анализа возможностей содержимого, их установки и использования;</p> <ul style="list-style-type: none"> • навыками разработки собственных итераторов для различных структур данных; • навыками использования комбинации методов из различных пакетов для научных вычислений при решении комплексных задач; • способностью к настройке сложных зависимостей различных пакетов в виртуальном окружении с учетом версий, совместимости, других особенностей; • навыками оптимизации работы скриптов с использованием программных реализаций примитивов
--	--	--	--



	инструментов для оценки времени вычислений и его оптимизации;	для подключения и работы с БД, реализовывать концепцию CRUD с помощью языковых средств;	синхронизации, специализированных модулей и пакетов; <ul style="list-style-type: none">• навыками анализа и оптимизации скорости работы вычислительных процессов с использованием языковых программных инструментов;
ПК-2	<ul style="list-style-type: none">• основы современных подходов и технологий разработки программного обеспечения и их классификацию;• основные синтаксические конструкции, основные типы данных и понятия языка Python (циклы, условные операторы, функции, модули, области видимости, основные типы аргументов функций и т.д.);• основные модели данных языка Python, а также дискретные структуры (список, кортеж, словарь, множество), основные методы	<ul style="list-style-type: none">• разрабатывать модели и алгоритмы и программно реализовывать их на ЭВМ;• работать с современными системами программирования;• производить тестирование и отладку программ.• обоснованно использовать условные операторы, операторы циклов и модели и дискретные структуры данных и функции для работы с ними;• грамотно оформлять функции, создаваемые пользователем, документировать и тестировать	<ul style="list-style-type: none">• современным инструментарием разработки ПО;• навыками использования изученных структур данных и алгоритмов при решении прикладных задач.• способами грамотной передачи и извлечения аргументов функции различных типов и значений из различных областей видимости;• способами обработки данных в потоках ввода, вывода, ошибок;• навыками и способами тестирования реализованных решений с

	<p>перебора и сортировки.</p> <ul style="list-style-type: none"> • основные принципы и техники тестирования; • основные типы исключений и методы их обработки; • встроенные коллекции, специализированные типы данных, методы работы с ними; • основные шаблоны проектирования и ситуации для их применения; • семантику и синтаксис, необходимые для построения графиков функций, основные принципы и языковые конструкции, основные графические модули и пакеты для работы с графикой; • названия, предназначение и функционал основных модулей и пакетов, предназначенных для организации научных 	<p>корректность их работы, использовать области видимости при создании переменных;</p> <ul style="list-style-type: none"> • анализировать и обоснованно использовать программное обеспечение и встроенные механизмы для тестирования и отладки Python-сценариев; • применять расширенный функционал обработки исключений; • исполнять методы для обработки данных во встроенных коллекциях и специализированных типах данных; • обоснованно применять шаблоны проектирования; • рационально применять методы для построения различных видов графиков функций, модулей и пакетов для работы с графикой; • выполнять методы пакетов для 	<p>использованием различных библиотек, осваивать новые современные библиотеки для тестирования кода;</p> <ul style="list-style-type: none"> • языковыми средствами для создания программ, устойчивых к исключениям; • навыками обработки вложенных исключений; • способами обработки данных встроенных коллекций, специализированных типов данных, навыками адекватного использования соответствующих структуре языковых средств. • методикой использования актуальных моделей/паттернов при проектировании приложений (в т.ч. реализации объектно-ориентированных решений), их комбинированию; • способностью к поиску и
--	--	---	--



	<p>вычислений в Python;</p> <ul style="list-style-type: none">• принципы использования виртуального окружения и методику работы с ним;• основные понятия, концепции, связанные с термином «фреймворк», а также синтаксис и семантику методов при его использовании;• базовыми понятиями и концепциями, связанными с базами данных, а также синтаксис и семантику программных методов при работе с ними;	<p>научных вычислений;</p> <ul style="list-style-type: none">• использовать инструментальные средства для создания и работы с виртуальным окружением;• рационально применять методы, программные инструменты при использовании фреймворка;• исполнять методы для подключения и работы с БД, реализовывать концепцию CRUD с помощью языковых средств;	<p>использованию современных графических библиотек сторонних разработчиков, решению с их помощью практических задач;</p> <ul style="list-style-type: none">• навыками разработки корутин (сопрограмм) и собственных генераторов для различных структур данных;• методикой реализации кроссплатформенных интерфейсов с использованием возможностей современных графических пакетов;• способностью использования возможностей фреймворков (например, Django) и расширений (например, requests, plywood) для создания веб-приложений, разработки API, REST-сервисов;• методологией разработки приложений с БД;
--	---	--	--



Средства оценки компетенций

Таблица 10

Семестр 2

Код компетенции	Средства оценки компетенций
ОПК-2	<p>Задания Инвариантной самостоятельной работы:</p> <p>Тема 1. Задание 1.2</p> <p>Тема 2. Задание 2.1</p> <p>Тема 3. Задание 3.1</p> <p>Тема 4. Задание 4.1</p> <p>Тема 5. Задание 5.1</p> <p>Тема 6. Задание 6.1</p> <p>Задания Вариантной самостоятельной работы:</p> <p>Тема 1. Задание 1.1.1, 1.1.2, 1.1.3, 1.1.4, 1.1.5, 1.1.6</p> <p>Тема 2. Задание 2.1.1, 2.1.2, 2.1.3, 2.1.4, 2.1.5, 2.1.6, 2.1.7, 2.1.8, 2.1.9, 2.1.10</p> <p>Тема 3. Задание 3.1.1, 3.1.2, 3.1.3, 3.1.4, 3.1.5, 3.1.6, 3.1.7</p> <p>Тема 4. Задание 4.1.1, 4.1.2, 4.1.3, 4.1.4, 4.1.5, 4.1.6, 4.1.7</p> <p>Тема 5. Задание 5.1.1, 5.1.2</p> <p>Тема 6. Задание 6.1.1, 6.1.2, 6.1.3</p>
ПК-2	<p>Задания Инвариантной самостоятельной работы:</p> <p>Тема 1. Задание 1.1</p> <p>Тема 2. Задание 2.1</p> <p>Тема 3. Задание 3.1</p> <p>Тема 4. Задание 4.1</p> <p>Тема 5. Задание 5.1</p> <p>Тема 6. Задание 6.1</p> <p>Задания Вариантной самостоятельной работы:</p> <p>Тема 1. Задание 1.1.1, 1.1.2, 1.1.3, 1.1.4, 1.1.5, 1.1.6</p> <p>Тема 2. Задание 2.1.1, 2.1.2, 2.1.5, 2.1.6, 2.1.7, 2.1.8, 2.1.9, 2.1.10, 2.1.11</p> <p>Тема 3. Задание 3.1.1, 3.1.2, 3.1.3, 3.1.4, 3.1.5, 3.1.6, 3.1.7</p> <p>Тема 4. Задание 4.1.1, 4.1.2, 4.1.3, 4.1.4, 4.1.5, 4.1.6, 4.1.7</p> <p>Тема 5. Задание 5.1.1, 5.1.2</p> <p>Тема 6. Задание 6.1.1, 6.1.2, 6.1.3</p>

Семестр 3

Код	Средства оценки компетенций
-----	-----------------------------



компетенции	
ОПК-2	Задания Инвариантной самостоятельной работы: Тема 1, задания 1,2,3,4 Тема 3, задания 1,2,3,4 Задания Вариантной самостоятельной работы: Тема 1, задания 1,2,3 Тема 3, задания 1,2,3,4
ПК-2	Задания Инвариантной самостоятельной работы: Тема 2, задания 1,2,3,4 Тема 4, задания 1,2,3,4 Задания Вариантной самостоятельной работы: Тема 2, задания 1,2,3,4,5,6,7,8 Тема 4, задания 1,2,3,4

Семестр 4

Код компетенции	Средства оценки компетенций
ОПК-2	Задания Инвариантной самостоятельной работы: Тема 2, задания 1, 2, 3, 4 Тема 3, задания 1, 2, 3 Задания Вариантной самостоятельной работы: Тема 2, задания 1, 2, 3, 4 Тема 3, задания 1, 2, 3, 4
ПК-2	Задания Инвариантной самостоятельной работы: Тема 1, задания 1, 2, 3, 4 Тема 4, задания 1, 2, 3 Задания Вариантной самостоятельной работы: Тема 1, задания 1, 2, 3 Тема 4, задания 1, 2, 3

Семестр 5

Код компетенции	Средства оценки компетенций
ОПК-2	Задания Инвариантной самостоятельной работы: Тема 1, задания 1, 2 Тема 2, задания 1, 2, 3 Задания Вариантной самостоятельной работы:



	Тема 1, задания 1, 2, 3 Тема 2, задания 1, 2
ПК-2	Задания Инвариантной самостоятельной работы: Тема 3, задания 1, 2, 3 Тема 4, задания 1, 2, 3 Задания Вариантной самостоятельной работы: Тема 3, задания 1, 2, 3 Тема 4, задания 1, 2, 3

Семестр 6

Код компетенции	Средства оценки компетенций
ОПК-2	Задания Инвариантной самостоятельной работы: Тема 2, задания 1, 2, 3, 4 Тема 3, задания 1, 2, 3 Задания Вариантной самостоятельной работы: Тема 2, задания 1, 2, 3 Тема 3, задания 1, 2, 3
ПК-2	Задания Инвариантной самостоятельной работы: Тема 1, задания 1, 2, 3, 4 Тема 4, задания 1, 2 Задания Вариантной самостоятельной работы: Тема 1, задания 1, 2, 3 Тема 4, задания 1, 2

Семестр 7

Код компетенции	Средства оценки компетенций
ОПК-2	Задания Инвариантной самостоятельной работы: Тема 1, задания 1, 2, 3 Тема 2, задания 1, 2, 3 Задания Вариантной самостоятельной работы: Тема 1, задания 1, 2, 3 Тема 2, задания 1, 2, 3
ПК-2	Задания Инвариантной самостоятельной работы: Тема 3, задания 1, 2, 3 Тема 4, задания 1, 2



Задания Вариантной самостоятельной работы:

Тема 3, задания 1, 2

Тема 4, задания 1, 2, 3

Методические материалы для проведения текущего контроля

Текущий контроль по дисциплине осуществляется в форме проверки выполненных инвариантных и вариативных заданий для самостоятельной работы.

Типовые задания для проведения процедур оценивания результатов освоения дисциплины в ходе текущего контроля, шкалы и критерии оценивания представлены в Приложении 1.

Методические материалы для проведения промежуточной аттестации

Промежуточная аттестация по дисциплине осуществляется в форме экзамена (2 семестр), экзамена (6 семестр), экзамена (7 семестр).

Типовые задания для проведения процедур оценивания результатов освоения дисциплины представлены в Приложении 2.

Шкала критериев оценивания

Таблица 11

Семестр 2

Шкала	Критерии
отлично	обучающийся должен: продемонстрировать знания изучаемых аспектов языка Python в полном объеме: распознавать операторы, структуры данных языка, исправлять ошибки в языковых конструкциях, дать корректные ответы на 85% тестовых заданий и более, быть способным корректно сформулировать определения всех ключевых понятий, воспроизвести по запросу информацию о функциональности изученных средств разработки, должно быть выполнено 85% и более заданий лабораторных работ, заданий для самостоятельной работы, представленных в виде корректно работающего веб-портфолио
хорошо	обучающийся должен: продемонстрировать знания изучаемых аспектов языка Python в достаточном объеме: распознавать операторы, структуры данных языка, исправлять ошибки в языковых конструкциях, дать корректные ответы на 70% тестовых заданий и более, быть способным корректно сформулировать определения основных понятий, воспроизвести по запросу информацию о основной функциональности изученных средств



	разработки, должно быть выполнено 70% и более заданий лабораторных работ, заданий для самостоятельной работы, представленных в виде корректно работающего веб-портфолио
удовлетворительно	обучающийся должен: продемонстрировать способность в целом ориентироваться в структуре изученного материала: дать корректные ответы на не менее чем 55% тестовых заданий, показать владение общим понятийным аппаратом дисциплины, уметь объяснить назначение основных элементов программы. У обучающегося должны присутствовать представления об основных управляющих конструкциях, операторах и структурах данных; должно быть выполнено не менее чем 55% заданий лабораторных работ, заданий для самостоятельной работы, представленных в виде корректно работающего веб-портфолио
неудовлетворительно	обучающийся: не знает значительной части изучаемого программного материала (ответил менее чем на 55% тестовых заданий); не владеет понятийным аппаратом дисциплины; не способен продемонстрировать знание значительной части структур данных и управляющих языковых конструкций; выполнено менее 60% заданий для самостоятельной работы, которые не отражены в веб-портфолио

Семестр 6

Шкала	Критерии
отлично	обучающийся должен: продемонстрировать знания изучаемых аспектов языка Python в полном объеме: распознавать операторы, структуры данных языка, исправлять ошибки в языковых конструкциях, дать корректные ответы на 85% тестовых заданий и более, быть способным корректно сформулировать определения всех ключевых понятий, воспроизвести по запросу информацию о функциональности изученных средств разработки, должно быть выполнено 85% и более заданий лабораторных работ, заданий для самостоятельной работы, представленных в виде корректно работающего веб-портфолио
хорошо	обучающийся должен: продемонстрировать знания изучаемых аспектов языка Python в достаточном объеме: распознавать операторы, структуры данных языка, исправлять ошибки в языковых конструкциях, дать корректные ответы на 70% тестовых заданий и более, быть способным корректно сформулировать определения основных понятий, воспроизвести по запросу информацию о основной функциональности изученных средств разработки, должно быть выполнено 70% и более заданий лабораторных работ, заданий для самостоятельной работы, представленных в виде корректно работающего веб-портфолио



удовлетворительно	обучающийся должен: продемонстрировать способность в целом ориентироваться в структуре изученного материала: дать корректные ответы на не менее чем 55% тестовых заданий, показать владение общим понятийным аппаратом дисциплины, уметь объяснить назначение основных элементов программы. У обучающегося должны присутствовать представления об основных управляющих конструкциях, операторах и структурах данных; должно быть выполнено не менее чем 55% заданий лабораторных работ, заданий для самостоятельной работы, представленных в виде корректно работающего веб-портфолио
неудовлетворительно	обучающийся: не знает значительной части изучаемого программного материала (ответил менее чем на 55% тестовых заданий); не владеет понятийным аппаратом дисциплины; не способен продемонстрировать знание значительной части структур данных и управляющих языковых конструкций; выполнено менее 60% заданий для самостоятельной работы, которые не отражены в веб-портфолио

Семестр 7

Шкала	Критерии
отлично	обучающийся должен: продемонстрировать знания изучаемых аспектов языка Python в полном объеме: распознавать операторы, структуры данных языка, исправлять ошибки в языковых конструкциях, дать корректные ответы на 85% тестовых заданий и более, быть способным корректно сформулировать определения всех ключевых понятий, воспроизвести по запросу информацию о функциональности изученных средств разработки, должно быть выполнено 85% и более заданий лабораторных работ, заданий для самостоятельной работы, представленных в виде корректно работающего веб-портфолио
хорошо	обучающийся должен: продемонстрировать знания изучаемых аспектов языка Python в достаточном объеме: распознавать операторы, структуры данных языка, исправлять ошибки в языковых конструкциях, дать корректные ответы на 70% тестовых заданий и более, быть способным корректно сформулировать определения основных понятий, воспроизвести по запросу информацию о основной функциональности изученных средств разработки, должно быть выполнено 70% и более заданий лабораторных работ, заданий для самостоятельной работы, представленных в виде корректно работающего веб-портфолио
удовлетворительно	обучающийся должен: продемонстрировать способность в целом ориентироваться в структуре изученного материала: дать корректные ответы на не менее чем 55% тестовых заданий,



	показать владение общим понятийным аппаратом дисциплины, уметь объяснить назначение основных элементов программы. У обучающегося должны присутствовать представления об основных управляющих конструкциях, операторах и структурах данных; должно быть выполнено не менее чем 55% заданий лабораторных работ, заданий для самостоятельной работы, представленных в виде корректно работающего веб-портфолио
неудовлетворительно	обучающийся: не знает значительной части изучаемого программного материала (ответил менее чем на 55% тестовых заданий); не владеет понятийным аппаратом дисциплины; не способен продемонстрировать знание значительной части структур данных и управляющих языковых конструкций; выполнено менее 60% заданий для самостоятельной работы, которые не отражены в веб-портфолио

Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература

Таблица 12

№ п/п	Наименование учебников, учебно-методических, методических пособий, разработок и рекомендаций
1	Прохоренок Н.А. Python 3. Самое необходимое/Н.А. Прохоренок, В.А. Дронов / Прохоренок Н.А., Дронов В.А. - Санкт-Петербург:БХВ-Петербург, 2015. - 464 с. - URL: http://ibooks.ru/reading.php?short=1&isbn=978-5-9775-3631-8
2	Сузи Р. Python/Р. Сузи. - Санкт-Петербург:БХВ-Петербург, 2015. - 768 с. - URL: http://ibooks.ru/reading.php?short=1&isbn=978-5-9775-1417-0
3	Саммерфилд, М. Python на практике [Электронный ресурс] : учебное пособие / М. Саммерфилд ; пер. с англ. Слинкин А.А.. — Электрон. дан. — Москва : ДМК Пресс, 2014. — 338 с. — Режим доступа: https://e.lanbook.com/book/66480 . — Загл. с экрана.
4	Буйначев, С.К. Основы программирования на языке Python [Электронный ресурс] : учебное пособие / С.К. Буйначев, Н.Ю. Боклаг. — Электрон. дан. — Екатеринбург : УрФУ, 2014. — 91 с. — Режим доступа: https://e.lanbook.com/book/98262 . — Загл. с экрана.

Дополнительная литература

Таблица 13

№ п/п	Наименование учебников, учебно-методических, методических пособий, разработок и
-------	---



	рекомендаций
1	Гринберг М. Разработка веб-приложений с использованием Flask на языке Python/Гринберг М.. - Москва:ДМК Пресс, 2014 - URL: https://e.lanbook.com/book/90103 . - ЭБС издательства «Лань». Коллекция «Информатика»
2	Давыдова, Н.А. Программирование [Электронный ресурс] : учебное пособие / Н.А. Давыдова, Е.В. Боровская. — Электрон. дан. — Москва : Издательство "Лаборатория знаний", 2015. — 241 с. — Режим доступа: https://e.lanbook.com/book/66124 . — Загл. с экрана.

Обеспеченность печатными изданиями дисциплины, перечисленными в разделе основной литературы программы, соответствует не менее 50 экземпляров каждого из изданий основной литературы на 100 обучающихся. Обеспеченность печатными изданиями дополнительной литературы соответствует не менее 25 экземпляров на 100 обучающихся.

Профессиональные базы данных и информационные справочные системы

Информационные справочные системы

- Федеральный портал «Российское образование» <https://edu.ru/>. Режим доступа: индивидуальный неограниченный доступ из любой точки, в которой имеется доступ к сети Интернет.
- Справочная правовая система «КонсультантПлюс». Режим доступа: доступ предоставляется в помещениях для самостоятельной работы обучающихся в фундаментальной библиотеке имени императрицы Марии Федоровны.

Электронные образовательные ресурсы и профессиональные базы данных

- Программирование на Си (<http://cpp.com.ru/>)
- О программировании вообще. О программировании на Си (<http://ermak.cs.nstu.ru/cprog/HTML/index.htm>)
- Учебное пособие «Программирование на языке Си» (http://publ.lib.ru/ARCHIVES/P/PODBEL'SKIY_Vadim_Valerievich/_Podbel'skiy_V.V..html#002)
- Сборник статей молодых учёных факультета ВМК МГУ (http://www.croco.net/croco/papers/stolyarov_2010.pdf)
- Язык программирования Python 3 для начинающих и чайников (<http://pythonworld.ru>)
- ПИТОНТЮТОР (<http://pythontutor.ru>)
- Курс “Python: основы и применение” Stepic (<https://stepik.org/course/512>)
- Курс “Программирование на Python” Stepic. (<https://stepik.org/course/67>)

Электронно-библиотечные системы

Таблица 14

Ссылка на информационный ресурс	Наименование разработки в электронной форме	Доступность
http://e.lanbook.com	Электронно-библиотечная система (ЭБС) на платформе издательства «Лань». Учебники и учебные пособия для университетов издательства «Лань» и десятков российских издательств	Индивидуальный неограниченный доступ из любой точки, в которой имеется доступ к сети Интернет
http://ibooks.ru	Электронно-библиотечная система (ЭБС) iBooks.Ru. Учебники и учебные пособия для университетов издательств «Питер», «БХВ-Петербург»	Индивидуальный неограниченный доступ из любой точки, в которой имеется доступ к сети Интернет
http://www.znaniyum.com/	Электронно-библиотечная система (ЭБС) Znaniyum. Учебники и учебные пособия для университетов издательства «Инфра-М» и десятков российских издательств	Индивидуальный неограниченный доступ из любой точки, в которой имеется доступ к сети Интернет
http://www.biblioclub.ru	Электронно-библиотечная система (ЭБС) Университетская библиотека онлайн. Учебники и учебные пособия для университетов издательства «ДиректМедиа» и десятков российских издательств	Индивидуальный неограниченный доступ из любой точки, в которой имеется доступ к сети Интернет

Каждый обучающийся в течение всего периода обучения должен быть обеспечен индивидуальным неограниченным доступом к одной или нескольким электронно-библиотечным системам (электронным библиотекам).

Электронно-библиотечные системы и электронная информационно-образовательная среда обеспечивает одновременный доступ не менее 25 процентов обучающихся по основной профессиональной образовательной программе, изучающих дисциплину.

Методические указания для обучающихся по освоению дисциплины

Знания по дисциплине формируются посредством изучения материала и самостоятельного изучения студентами основной и дополнительной литературы, рекомендованной программой.



При изучении дисциплины используются следующие формы занятий: лекционные занятия, лабораторные занятия.

Рекомендации по подготовке к аудиторным занятиям

Лекционные занятия могут проходить:

- в традиционном формате с использованием мультимедийных презентаций;
- с использованием элементов электронного обучения, мобильных устройств.

Лабораторные занятия

В ходе подготовки к лабораторным занятиям необходимо изучить основную литературу, ознакомиться с дополнительной литературой, интернет – ресурсами. Важно учитывать рекомендации преподавателя и требования программы учебной дисциплины. Рекомендуется пользоваться материалами в электронных учебных курсах в системе Moodle.

В ходе занятия важно опираться на предложенный преподавателем алгоритм выполнения заданий и использования предложенных методических материалов. При необходимости задавать уточняющие вопросы, активно участвовать в обсуждении изучаемых вопросов. В ходе образовательного процесса рекомендуется активно использовать мобильные устройства.

Организация самостоятельной работы обучающихся

Самостоятельная работа обучающегося по дисциплине предполагает активное использование широкого спектра информационных технологий, технического оборудования, в частности - мобильные телефоны, планшеты и т.д., программное обеспечение для осуществления коммуникаций (студент-преподаватель, студент-студент, группа студентов) в электронной информационно-образовательной среде.

Подготовка к экзамену

В процессе подготовки к экзамену, обучающемуся рекомендуется так организовать свою учебу, чтобы все виды работ и заданий, предусмотренные программой учебной дисциплины, были выполнены в срок. Основное в подготовке к экзамену - это повторение всего материала учебной дисциплины. Целесообразно повторять пройденный материал в строгом соответствии с программой учебной дисциплины, примерным перечнем вопросов, заданий, которые выносятся на экзамен и содержащихся в данной программе. Рекомендуется уделять особое внимание разработке своих компьютерных программ и их анализу.

Образовательный процесс по дисциплине осуществляется с использованием возможностей электронной информационной образовательной среды университета, электронных учебных курсов по дисциплине.

В ходе изучения дисциплины студент формирует электронное портфолио по дисциплине. В состав электронного портфолио включаются отчеты по лабораторным работам, а также результаты самостоятельной работы студентов.

Для качественного освоения дисциплины «Программирование» студентам предлагаются темы, содержание и процесс изучения которых предполагает активное использование широкого спектра



технологий электронного обучения, электронных образовательных ресурсов.

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

Лицензионное программное обеспечение

Таблица 15

Программное обеспечение	Аналог
OC Microsoft Windows Desktop Education ALNG LicSAPk AcademicEdition 7/8/10	OC Linux (Ubuntu, Astra Linux)
MS Office ProPlus for Students/Faculty ALNG AcademicEdition 2010/ 2013/ 2016 / Office 365	OpenOffice, LibreOffice
Антивирус Касперского Endpoint Security 10	Clam AntiVirus, Rkhunter
Система проведения вебинаров «TrueConf Online»TrueConf Online»	Приложение Skype
Справочная правовая система Консультант Плюс»	
MATLAB Basic suite MathWorks	Пакет прикладных математических программ Scilab
Набор программ MS Imagine Academy ALNG Subscriptions	
Растровый графический редактор GIMP	
Векторный графический редактор Inkscape	
Приложение для верстки документов Scribus	
Аудиоредактор звуковых файлов Audacity	
Визуальная событийно-ориентированная среда программирования Scratch	
Архиватор 7-Zip	
Проигрыватель аудио и видео файлов VLC player	
Система управления онлайн курсами LMS Moodle	
Текстовый редактор Notepad++	
Файловый менеджер Far manager	
Система управления базами данных Firebird	



server 2.5	
Среда разработки программного обеспечения Lazarus	
Среда разработки модульных кроссплатформенных приложений Eclipse	

Электронное обучение по дисциплине применяется при организации самостоятельной работы обучающихся на СДО Moodle и не требует замены специально оборудованных помещений их виртуальными аналогами, позволяющими обучающимся осваивать умения, предусмотренные профессиональной деятельностью.

Дистанционные образовательные технологии при реализации дисциплины не используются.

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Для проведения занятий лекционного типа, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы используются учебные аудитории, отвечающие противопожарным правилам и нормам, обеспечивающих проведение всех видов деятельности обучающихся при освоении дисциплины, а также помещения для хранения и профилактического обслуживания учебного оборудования. Учебные аудитории укомплектованы специализированной мебелью и техническими средствами обучения (мультимедийными комплексами), служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа имеются наборы демонстрационного оборудования и учебно-наглядных пособий, обеспечивающие тематические иллюстрации, соответствующие темам программы дисциплины.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с подключением к сети "Интернет" и обеспечены доступом в электронную информационно-образовательную среду вуза.

Разработчики:

кафедра
компьютерных
технологий и
электронного
обучения

заведующий кафедрой,
доктор педагогических
наук, профессор

Е.З. Власова



кафедра
компьютерных
технологий и
электронного
обучения

доцент, кандидат
педагогических наук,
доцент

С.В. Гончарова

кафедра
компьютерных
технологий и
электронного
обучения

доцент, кандидат
педагогических наук,
доцент

Е. Ю. Авксентьева

кафедра
компьютерных
технологий и
электронного
обучения

ассистент

Н.Н. Жуков