МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Б1.О.01
(индекс дисциплины)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Средства программной разработки

(наименование дисциплины)

по направлению подготовки
40.03.01 Юриспруденция

направленность (профиль)
Гражданско-правовой

Форма обучения: очная

Год набора: 2023

Общая трудоемкость: 3 ЗЕ

Распределение часов дисциплины по семестрам

Семестр	3	**Итого**
Форма контроля / **Вид занятий**	зачет с оценкой	
Лекции	-	-
Лабораторные		
Практические	48	48
Руководство: курсовые работы (проекты) / РГР		
Промежуточная аттестация	0,25	0,25
Контактная работа	48.25	48.25
Самостоятельная работа	59.75	59.75
Контроль		
Итого	108	108

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 36AF32830467F939D7BA79FD8101372D
Владелец: Гущина Оксана Михайловна
Действителен: с 10.09.2026 до 04.12.2027

Рабочую программу составил(и):
Заведующий кафедрой «Прикладная математика и информатика», доцент, канд. пед. наук, Гущина О.М.

(должность, ученое звание, степень, Фамилия И.О.)

Ассистент кафедры «Прикладная математика и информатика», Герасимов А.В.

(должность, ученое звание, степень, Фамилия И.О.)

Рецензирование рабочей программы дисциплины:

☒ Отсутствует

☐ Рецензент

(должность, ученое звание, степень, Фамилия И.О.)

Программа практики составлена на основании ФГОС ВО и учебного плана направления подготовки 40.03.01 Юриспруденция

Срок действия рабочей программы дисциплины до «31» августа 2028 г.

СОГЛАСОВАНО

Руководитель центр медицинской химии

«___» _____ 20___ г. _____

(подпись) *(И.О. Фамилия)*

УТВЕРЖДЕНО

На заседании кафедры «Прикладная математика и информатика»

(протокол заседания № 1 от «31» августа 2023 г.).

1. Цель освоения дисциплины

Цель освоения дисциплины – формирование у обучающихся цифровых компетенций в области создания алгоритмов и компьютерных программ, пригодных для практического применения в цифровой среде для взаимодействия с обществом и решения цифровых задач в профессиональной деятельности.

2. Место дисциплины в структуре ОПОП ВО

Дисциплины и практики, на освоении которых базируется данная дисциплина – «Высшая математика. Элементы высшей алгебры и геометрии»

Дисциплины и практики, для которых освоение данной дисциплины необходимо как предшествующее: Производственная практика (практика в ИТ сфере).

3. Планируемые результаты обучения

Формируемые и контролируемые компетенции (код и наименование)	**Индикаторы достижения компетенций** (код и наименование)	**Планируемые результаты обучения**
УК-1. Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.1 Знает принципы сбора, отбора и обобщения информации	Знать: методы сбора, отбора и обобщения информации, понятие системного подхода, виды алгоритмов Уметь: проводить сбор, отбор и обобщение информации, разрабатывать оригинальные алгоритмы Владеть: методикой системного подхода для решения профессиональных задач
	УК-1.2 Умеет соотносить разнородные явления и систематизировать их в рамках избранных видов профессиональной деятельности	Знать: инструменты анализа и систематизации информации, методы анализа проблем и принятия решений, современные технологии разработки алгоритмов и программных решений Уметь: анализировать и систематизировать информацию, разрабатывать алгоритмы и программные решения Владеть: техникой применения системного подхода для решения поставленных задач, навыками формализации процессов для использования интеллектуальных программ

Формируемые и контролируемые компетенции (код и наименование)	**Индикаторы достижения компетенций** (код и наименование)	**Планируемые результаты обучения**
	УК-1.3 Имеет практический опыт работы с информационными источниками, опыт научного поиска, создания научных текстов	Знать: виды информационных источников, технологии поиска, критического анализа и синтеза информации, методы разработки алгоритмов и программных приложений Уметь: искать, обрабатывать информацию с использованием современных ИТ инструментов Владеть: инструментальными средствами поиска и практической работы с информационными источниками, методами формализации процессов в реализации интеллектуальных программных решений

4. Структура и содержание дисциплины

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля
Модуль 1. Введение в язык программирования Python	Пр 1	ПР1.1. Установка Python 3, установка и настройка редактора кода Visual Studio Code	3	2	10	-	Отчет по практическим заданиям модуля 1
	Пр 2	ПР1.2. Переменные, операции с ними, базовый ввод / вывод. Строки, циклы, условия.	3	2		-	
	Пр 3	ПР1.3. Коллекции объектов (списки, кортежи, словари, sets)	3	2		-	
	Пр 4	ПР1.4. Работа с файлами. Функции, модули, исключения	3	2		-	
	СР	Самостоятельное изучение методических рекомендаций при подготовке к практическим работам.	3	**8**		-	
Модуль 2. Введение в искусственный интеллект и основные методы машинного обучения.	Пр 5	ПР2.1. Методы работы с таблицами в Python.	3	2	20	-	Отчет по практическим заданиям модуля 2
	Пр 6	ПР2.2. Агрегация и визуализация данных.	3	2		-	

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля
	Пр 7	ПР2.3. Проведение первичного анализа данных	3	2		-	
	Пр 8	ПР2.4. Использование и сравнение алгоритмов классификации: kNN	3	2		-	
	Пр 9	ПР2.5. Быстрый поиск ближайших соседей	3	2		-	
	Пр 10	ПР2.6. Решающие деревья и их ансамбли.	3	2		-	
	Пр 11	ПР2.7. Логистическая регрессия	3	2		-	
	Пр 12	ПР2.8. Использование и оценка алгоритмов регрессии.	3	2		-	
	Пр 13	ПР2.9. Подбор оптимальных параметров регрессии	3	2		-	
	Пр 14	ПР2.10. Оптимизационные задачи и их решения	3	2		-	
	Пр 15	ПР2.11. Подбор гиперпараметров алгоритма с помощью методов	3	2		-	
	Пр 16	ПР2.12. Классификация изображений	3	2		-	
	Пр 17	ПР2.13. Работа с текстами и их векторными представлениями текстов	3	2		-	

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля
	Пр 18	ПР2.13. Работа с текстами и их векторными представлениями текстов	3	2		-	
	СР	Самостоятельное изучение методических рекомендаций при подготовке к практическим работам.	3	**12**		-	
Модуль 3. Объектно-ориентированное программирование	Пр 19	ПР3.1. Понятие классов и их создание. Объекты и их атрибуты. Создание объектов	3	2	10	-	Отчет по практическим заданиям модуля 3
	Пр 20	ПР3.2. Наследование. Использование декораторов	3	2		-	
	СР	Самостоятельное изучение методических рекомендаций при подготовке к практическим работам.	3	**12**		-	
Модуль 4. Основы web-разработки	Пр 21	ПР4. Создание проекта в Django. Создание веб-страницы в Django. Использование шаблонов в Django	3	2	15	-	Отчет по практическим заданиям модуля 4
	СР	Самостоятельное изучение методических рекомендаций при подготовке к практическим работам.	3	**7,75**		-	
Модуль 5. Базы данных и ORM	Пр 22	ПР5.1. Работа с данными, хранящимися в разных таблицах с использованием Django ORM. ПР5.2. Работа с формами в Django	3	2	15	-	Отчет по практическим

Модуль (раздел)	Вид учебной работы	Наименование тем занятий (учебной работы)	Семестр	Объем, ч.	Баллы	Интерактив, ч.	Формы текущего контроля
							заданиям модуля 5
	СР	Самостоятельное изучение методических рекомендаций при подготовке к практическим работам.	3	**12**		-	
Модуль 6. Сборка и запуск приложения	Пр 23	ПР6.1. Сборка и отладка веб-приложения. ПР6.2. Дополнительные приложения	3	2	20	-	Отчет по практическим заданиям модуля 6
	СР	Самостоятельное изучение методических рекомендаций при подготовке к практическим работам.	3	**8**		-	
	ПА	Промежуточная аттестация	3	0,25	-	-	
	Псц	Посещаемость	3	-	10	-	
	Пр 24	Дифференцированный зачет	3	2	100	-	Итоговый тест
	Итого:			**108**			

Схема расчета итогового балла:
Текущий рейтинг + Результат итогового теста и все делится на 2

5. Образовательные технологии

В рамках учебного курса предусмотрены следующие образовательные технологии:

– технологии традиционного обучения в форме практических работ и самостоятельной работы обучающихся;

– технология проектного обучения в форме: реализация и защита отчетов по практическим работам.

Технологии традиционного обучения - организация учебного процесса в вузе, основанная на лекционных и практических формах обучения: объяснительно-иллюстративное обучение. Данная технология применяется во всех модулях курса.

- Технология интерактивного обучения - организация учебного процесса, которая предполагает максимальную активность обучающихся в процессе формирования ключевых компетенций. На практическом занятии обучающиеся представляют результат выполнения заданной работы.

Для обучающихся всех форм обучения предусмотрено получение консультационной помощи. Особое внимание необходимо уделить самостоятельному изучению нормативных источников и рекомендованной литературы.

6. Методические указания по освоению дисциплины

6.1. Рекомендации по подготовке к практическим занятиям

Обучающимся следует:

- при подготовке к занятиям обязательно использовать не только лекции, учебную литературу, но и другие источники;
- в начале занятий задать преподавателю вопросы по материалу, вызвавшему затруднения в его понимании и освоении при решении задач, заданных для самостоятельного решения;
- на занятии доводить каждую задачу до окончательного решения, демонстрировать понимание проведенных расчетов (анализов, ситуаций), в случае затруднений обращаться к преподавателю.

Для того чтобы практические занятия приносили максимальную пользу, необходимо помнить, что решение задач проводятся по рассмотренному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться обучающимся на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях обучающийся не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если обучающийся видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные

результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

6.2. Рекомендации по подготовке к тестированию по темам курса

Подготовка к зачету способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к зачету, обучающийся ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На зачете обучающийся демонстрирует то, что он приобрел в процессе обучения по конкретной учебной дисциплине.

Необходимо ориентировать обучающихся на систематическую подготовку к занятиям в течение семестра, что позволит использовать время экзаменационной сессии для систематизации знаний.

7. Оценочные средства

7.1. Паспорт оценочных средств

Семестр	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
3	УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	Отчеты по практическим работам по модулям 1-6 Тестовые задания по модулям 1-6 Вопросы к зачету №1-300

7.2. Типовые задания или иные материалы, необходимые для текущего контроля

7.2.1. Типовые тестовые материалы
(наименование оценочного средства)

Типовые примеры заданий
Модуль 1. Введение в язык программирования Python

1. Строковые выражения функции print() могут заключаться в

– круглые скобки
+ одинарные кавычки
+ двойные кавычки
+ тройные кавычки

2. Что напечатает следующий код?

```
a=[1,2,3]
a[2]=2
print(a)
```

+ [1, 2, 2]
– [1, 2, 2, 3]
– [1, 2, 3]
– [1, 2]

3. Вспомогательные строки, не обрабатываемые программой, обозначаются знаком # перед началом строки и действуют до конца строки называются ….

– команды
– списки
– записи

+ комментарии

4. Почему вложенная инструкция должна отступать на 4 пробела?

+ чтобы программа понимала, где начинается вложенная инструкция и к какой основной она относится

- т.к. 4 отступа совпадают с табуляцией
- чтобы передать управление команде на следующей строке
- для создания «визуально чистого» и понятного кода в виде «лесенки»

5. Что будет выведено на экран?

```
x=25
def function(x):
  x+=100
function(x)
print(x)
```

+ 25
- 100
- 125
- 200

6. Язык Python характерен своей неявной динамической типизацией. Что это означает?

+ что при задании какой-либо переменной, нам не надо объявлять ее

+ что достаточно просто присвоить значение переменной и в зависимости от того, какое это значение, Python сам определит тип переменной

- что нам необходимо указывать раздел var для объявления переменных
- что тип переменной можно указать в любом месте программы

7. Что будет результатом выполнения кода?

```
a = 0
b = 0
print(a < b)
print(a > b)
print(a == b)
```

- False
- False
+ True
- False
- True
- False
- False
- True

8. Какой из представленных ниже типов данных являются своего рода хранилищем данных разного типа (другими словами массивами, только хранить они могут данные разных типов)?

- массивы
- множества
+ списки
- кортежи

9. Благодаря какому методу в Python после открытия файла можно прочитать из него информацию?

- open()
+ read()

- range()
- close()

10. При использовании какого типа цикла необходимо всегда помнить и следить за тем, чтобы было прописано условие выхода из цикла?
+ while
- for
- continue
- else

Модуль 2 Введение в искусственный интеллект и основные методы машинного обучения. Системы глубокого обучения

1. Какая библиотека в Python обычно используется для работы с таблицами?
+ Pandas
- Numpy
- Matplotlib
- Scikit-learn

2. Какая функция в Pandas используется для чтения данных из файла CSV?
+ read_csv()
- load_csv()
- import_csv()
- open_csv()

3. Какая функция в Pandas используется для агрегации данных по группам?
+ groupby()
- combine()
- merge()
- aggregate()

4. Какая библиотека в Python обычно используется для визуализации данных?
- Pandas
- Numpy
+ Matplotlib
- Scikit-learn

5. Какой алгоритм классификации склонен к переобучению при большом количестве признаков?
- kNN
- Решающие деревья
- Логистическая регрессия
+ Ансамбли решающих деревьев

6. Какие алгоритмы часто используются для классификации изображений?
+ Сверточные нейронные сети (Convolutional Neural Networks)
- Решающие деревья
- Логистическая регрессия
- Метод k-ближайших соседей (kNN)

7. Какой метод используется для оценки качества модели классификации изображений?
- Mean Absolute Error (MAE)

- R-squared (коэффициент детерминации)
- Precision
+ Accuracy (точность)

8. Какой алгоритм классификации изображений основан на использовании градиентных бустингов?
- Логистическая регрессия
- Метод k-ближайших соседей (kNN)
- Случайный лес (Random Forest)
+ Градиентный бустинг деревьев

9. Какое представление текста используется для обработки текстовых данных в машинном обучении?
- Числовые значения
- Бинарные значения
+ Векторные представления
- Символьные представления

10. Какой метод позволяет учитывать контекст и порядок слов при создании векторных представлений текста?
- Метод Bag-of-Words (мешок слов)
- TF-IDF (Term Frequency-Inverse Document Frequency)
+ Рекуррентные нейронные сети (RNN)
- Word2Vec

Модуль 3 Объектно-ориентированное программирование

1. Объектом объектно-ориентированного программирования называется …

+ экземпляр некоторого класса
- класс, образующий иерархии
- абстракция
- любой тип данных

2. Что относится к основным принципам ООП?

+ инкапсуляция
+ полиморфизм
+ наследование
- делегирование

3. Какой принцип ООП является способностью использовать общий интерфейс для нескольких форм (типов данных)?

- инкапсуляция
+ полиморфизм
- наследование
- абстракция

4. Конструктор класса задается методом с именем:

- __new__
- __int__
+ __init__
- new

5. Что будет результатом исполнения кода?

```
class test:
    test = 4
print(Test.test)
```

- None
+ Ошибка
- True
- False

6. Если в классе определены два метода с одинаковыми именами и разными списками параметров, то:

+ второе определение заменит первое, не выдав ошибок
- сохранится первое определение и выведется ошибка
- не сохранится ни первое, ни второе определение, выйдет ошибка
- сохранится второе определение и будет выведено предупреждение об ошибке

7. В языке Python объектами являются:

- экземпляры классов и переменные
+ все типы данных
- экземпляры классов, классы и переменные
- переменные и функции

8. Как создать конструктор класса A?

+ def__init__(параметры конструктора)
- def__A__(параметры конструктора)
- def__A__:(параметры конструктора)
- def__init__:(параметры конструктора)

9. Сколько конструкторов в классе может быть в Python?

- 0
- до 100
+ 1
- бесконечно

10. В элементе кода

```
class Rectangle:
    default_color = "green"
```

цвет объявлен ….

+ статистическим атрибутом класса
- динамическим атрибутом класса
- постоянной переменной
- динамической переменной

Модуль 4 Основы web-разработки

1. Веб-фреймворк — это …

+ программный пакет, который упрощает процесс создания веб-приложений
+ каркас для написания веб-приложений
- редактор кода для разработчиков
- ПО, которое объединяет инструменты для разработки приложений и их тестирования в едином интерфейсе

2. IDE (Integrated Development Environment) – это

+ ПО, которое объединяет инструменты для разработки приложений и их тестирования в едином интерфейсе
- каркас для написания веб-приложений

- программа, работающая как текстовый редактор, только более подходящая для написания кода

- специальная библиотека, упрощающая написание кода

3. IDE состоит из:

+ самого текстового редактора для написания и редактирования кода;

+ компилятора – инструмента, позволяющего перевести текст, написанный на языке программирования, в набор машинных кодов;

+ отладчика, проверяющего код и устраняющего в нем ошибки;

+ инструментов для автоматизации сборки кода, ускоряющих процесс разработки.

4. Какие из перечисленных IDE объединяют инструменты для разработки приложений Python и их тестирования в едином интерфейсе

+ PyCharm

- Microsoft SQL Server

+ Visual Studio Code

- SQLite

5. Какая информация хранится в файле проекта settings.py?

- это специальный пустой файл, который сообщает Python, что текущий каталог следует рассматривать как пакет Python

+ этот файл содержит все настройки и конфигурацию проекта Django

- скрипт на Python для хранения шаблонов URL-адресов проекта Django

- скрипт на Python для запуска сервера разработки, а также развертывания проекта в производственной среде

6. По умолчанию manage.py сервер разработки всегда запускается с порта 8000, но вы можете изменить его на какой-либо другой порт, используя следующую команду

+ ./manage.py сервер запуска <порт>

- ./manage.py

- ./manage.ru

./manage.ru <порт>

7. Согласно номенклатуре Django, проект Django представляет собой - набор веб-приложений

+ набор конфигураций и приложений, эти конфигурации и приложения вместе составляют целое веб-приложение

- набор конфигураций

- набор функций и конфигураций, которые совместно составляют веб-приложение

8. Архитектура MVC делит приложение на следующие уровни:

+ Модель

+ Вид

+ Контроллер

- Инструмент

9. Является ли Django фреймворком с шаблоном MVC?

- в точности совпадает с MVC

+ следует шаблону MVC, но использует немного другую терминологию

- не имеет ничего общего с шаблоном MVC

- отличается от шаблона MVC наличием СУБД

10. Утилитой командной строки для взаимодействия с вашим проектом Django или его поддержки является …
- settings.py
- urls.py
- wsgi.py
+ manage.py

Модуль 5. Базы данных и ORM

1. Какие из перечисленных ниже СУБД можно подключить в Visual Studio Code
+ Microsoft SQL Server
+ SQLite
- PyCharm
+ MySQL

2. В Django вместо выполнения необработанных SQL-запросов для доступа к базе данных можно использовать …
- Visual Studio Code
- IDE
- SQL запросы
+ Django ORM

3. Какие типы отношений могут существовать внутри моделей?
+ «один к одному»
+ «один ко многому»
+ «много ко многому»
+ «бесконечно к бесконечному»

4. Чтобы сохранить объект в базе данных, необходимо вызвать …
- now()метод
+ save()метод
- delete()метод
- create()метод

5. Какой из методов позволяет создавать и передавать объект в базу данных за один раз, вместо отдельного вызова save()метода
- now()метод
- objects метод
- delete()метод
+ create()метод

6. Какой из методов позволяет создавать и фиксировать несколько объектов
+ bulk_create() метод
- all() метод
- delete()метод
- create()метод

7. Какой из методов позволяет извлекать все записи из таблицы
- bulk_create() метод
+ all() метод
- delete()метод
- create()метод

8. Что означает NOT NULL ограничение на столбец таблицы?

+ он не может принимать NULL значения
- он не может быть равным нулю
- он может принимать только значения ноль
- он может принимать только значения, указанные по умолчанию

9. Для подключения данных при работе с отношениями "многие ко многим" Django использует менеджер, называемый …

- delete manager
+ related manager
- create manager
- all manager

10. В записи функции get_object_or_404(klass, **kwargs) , в роли klass может быть …

+ модель
+ менеджер
+ набор запросов
- метод

Модуль 6. Сборка и запуск приложения

1. На странице индекса сайта администратора Django отображается …

+ список моделей, доступных в вашем проекте
- список методов, использованных в вашем проекте
- список объектов базы данных, используемых в проекте
- список отношений внутри моделей

2. В дополнение к отображению User объектов, страница списка пользователей позволяет нам выполнять следующие задачи:

+ добавьте нового пользователя
+ измените существующего пользователя
+ поиск и фильтрация пользователей
+ удалить пользователя

3. По умолчанию сайт администратора Django не загружает ни одну модель из приложений, которые мы создаем. Чтобы добавить модель в Django admin, вам необходимо изменить _____, доступный в каждом каталоге приложения.

- settings.py файл
- urls.py файл
- wsgi.py файл
+ admin.py файл

4. Приложение, которое используется для отслеживания различных типов контента в проекте называется …

+ django.contrib.contenttype
- django.contrib.auth
- django.contrib.admin
- django.contrib.staticfiles

5. Какие типы пользователей могут быть в Django?

+ анонимный пользователь
+ пользователь
- статичный пользователь

- новый пользователь

6. Django предоставляет пользовательский метод для objects менеджера с именем create_user()для создания пользователей. Он принимает параметры, представленные ниже:

– тип пользователя
+ имя пользователя
+ пароль
+ адрес электронной почты

7. Какие из представленных ниже полей могут являться полями User модели?

+ password
+ email
+ is_active
+ last_login

8. Каким количеством символов ограничены поля, относящиеся к имени/ фамилии пользователя к в User модели?

- ограничений нет
+ 30
- 50
- 100

9. Какой алгоритм используется по умолчанию в Django для создания хэшей паролей?

- SHA256
+ PBKDF2
- NIST
- SHA512

10. Какие из методов, перечисленных ниже, предоставляются User моделью?

+ get_username()
+ get_full_name()
+ check_password(pass)
+ is_authenticated()

7.2.2 Пример практической работы

(наименование оценочного средства)

Практическая работа 6. Функции, модули, исключения

Задача 1.

Напишите функцию is_even(n), которая принимает на вход целое число n и возвращает True, если число четное, и False, если нечетное.

Задача 2.

Напишите функцию plus_number(), которая выполняет одну простую задачу — выводит результат сложения 2 и значения переменной number. В переменную number должно быть передано число. Обработайте ситуацию, если в эту переменную переданы данные какого-то другого типа. В случае ошибки напечатайте в консоли сообщение «Ожидаемый тип данных — число!».

Требования к оформлению

Отчет должен содержать подробное описание (включая иллюстрации). Отчёт по практическому занятию выполняется на страницах формата А4 в электронном виде.

При оформлении отчёта используется сквозная нумерация страниц, считая титульный лист первой страницей. Номер страницы на титульном листе не ставится. Номера страницы ставятся по центру вверху.

При оформлении отчёта соблюдать следующие требования:

- Для заголовков: полужирный шрифт, 14 пт, центрированный.
- Для основного текста: нежирный шрифт, 14 пт, выравнивание по ширине.
- Во всех случаях тип шрифта – Times New Roman, отступ абзаца 1.25 см, полуторный междустрочный интервал.
- Поля: левое – 2 см, правое, верхнее и нижнее – 1 см.

Процедура оценивания

Оценка выполненного практического занятия проводится по следующим критериям:

1. Наличие всей существенной информации по работе
2. Точность и полнота предоставляемых сведений
3. Непротиворечивость приводимой информации
4. Правильность интерпретаций и выводов, которые сделаны по результатам работы
5. Степень достижения обучающимся поставленной цели
6. Обоснованность применяемого решения
7. Грамотность (содержательная) используемых формулировок

Критерии оценки за отчеты по практическим работам**:**

Формы текущего контроля	**Критерии и нормы оценки**
Отчет по практическим работам модуля 1, 3	10 баллов – задание выполнено в полном объёме без замечаний 7 баллов – задание выполнено в полном объёме, присутствуют замечания по выполнению задания 5 баллов – задание выполнено не в полном объёме, присутствуют несущественные замечания 2 балла – задание выполнено не в полном объёме, присутствуют замечания по выполнению задания 0 баллов – задание не выполнено «- 10 баллов» - работа не сдана обучающимся
Отчет по практическим работам модуля 4, 5	15 баллов – задание выполнено в полном объёме без замечаний 10 баллов – задание выполнено в полном объёме, присутствуют замечания по выполнению задания 5 баллов – задание выполнено не в полном объёме, присутствуют несущественные замечания 3 балла – задание выполнено не в полном объёме, присутствуют замечания по выполнению задания 0 баллов – задание не выполнено «- 15 баллов» - работа не сдана обучающимся

Отчет по практическим работам модуля 2, 6	20 баллов – задание выполнено в полном объёме без замечаний 15 баллов – задание выполнено в полном объёме, присутствуют замечания по выполнению задания 10 баллов – задание выполнено не в полном объёме, присутствуют несущественные замечания 5 баллов – задание выполнено не в полном объёме, присутствуют замечания по выполнению задания 0 баллов – задание не выполнено «- 20 баллов» - работа не сдана обучающимся
Посещаемость	10 баллов. Отсутствие на занятиях ведет к пропорциональному снижению баллов. Полное отсутствие соответствует – 100 (минус 100 баллов)

Типовые задания для итогового теста

Задание 1

Выберите один правильный вариант ответа.

Какую команду необходимо написать в командной строке, чтобы найти информацию об установленной версии Python?

а) python

б) py

в) «python»

г) ver

Правильный ответ: а, б.

Задание 2

Выберите один правильный вариант ответа.

Какой из следующих операторов Python может быть использован для фильтрации и сортировки результатов поиска?

а) if-else

б) for-in

в) while

г) lambda

Правильный ответ: б.

Задание 3

Выберите несколько правильных вариантов ответа.

Что подразумевается под системным подходом в программировании?

а) разработка программного обеспечения для операционных систем

б) проектирование программ с учетом взаимодействия компонентов системы

в) использование системных вызовов в программном коде

г) оптимизация программы с целью повышения производительности

Правильный ответ: б, г.

Задание 5

Заполните пропуск: вставьте пропущенное слово.

Оптимизация программы является одним из аспектов системного подхода и направлена на повышение _____.

Правильный ответ: производительности.

Задание 6

Дайте развернутый ответ.

Какой принцип отбора информации, используемый при программировании на Python, означает «не повторяйте самого себя»?

Правильный ответ:

Принцип отбора информации, используемый при программировании на Python и означающий "не повторяйте самого себя", называется DRY (Don't Repeat Yourself). Этот принцип подразумевает, что код должен быть организован таким образом, чтобы избегать дублирования логики или функциональности. Вместо этого, повторяющиеся фрагменты кода следует выносить в отдельные функции, классы или модули, чтобы их можно было повторно использовать и изменять только в одном месте.

Задание 7

Дайте развернутый ответ.

В чем заключается принцип модульности в Python?

Правильный ответ:

Модульность предполагает разделение программы на отдельные модули или компоненты, каждый из которых выполняет определенную функцию или решает определенную задачу. Эти модули могут быть независимыми и могут взаимодействовать друг с другом через определенные интерфейсы. Принцип модульности помогает создавать гибкие и масштабируемые программы, которые легко поддерживать и развивать.

Задание 8

Дайте развернутый ответ.

Каким образом можно осуществлять поиск и фильтрацию данных веб-приложения на языке программирования Python?

Правильный ответ:

Для поиска и фильтрации данных веб-приложения на языке программирования Python можно использовать различные подходы. Например, можно применять SQL-запросы к базе данных для извлечения нужных данных с использованием ORM-библиотек, таких как SQLAlchemy или Django ORM. Также можно использовать специализированные инструменты и библиотеки, например, Elasticsearch или Apache Solr, для выполнения расширенного и эффективного поиска и фильтрации данных.

Задание 9

Дайте развернутый ответ.

Как системный подход влияет на производительность приложений, разработанных на Python?

Правильный ответ:

Ответ: Системный подход влияет на производительность приложений, разработанных на Python, путем оптимизации структуры и взаимодействия компонентов. Системный анализ позволяет выявить узкие места и оптимизировать их работу. Кроме того, системный подход способствует использованию эффективных алгоритмов и структур данных, что также может повысить производительность приложений.

Задание 10

Дайте развернутый ответ.

Каким образом можно интегрировать поиск и критический анализ информации в веб-приложение на Python?

Правильный ответ:

Для интеграции поиска и критического анализа информации в веб-приложение на Python можно использовать встроенные и внешние библиотеки и инструменты для критического анализа информации (анализ тональности, текстовые модели и пр.). Для интеграции всех этих инструментов в веб-приложение можно использовать фреймворк Django.

7.3. Оценочные средства для промежуточной аттестации по итогам освоения дисциплины

7.3.1. Вопросы к промежуточной аттестации

Семестр _____3_____

п/п	Вопросы к зачету
1.	Переменные и типы данных языка Python. Преобразование типов
2.	Ввод-вывод данных, оператор присваивания
3.	Какие структуры данных есть в Python
4.	На какие два больших класса можно разделить типы/структуры данных? (mutable, immutable)
5.	Условный оператор и каскадная условная конструкция в Python
6.	Строки в Python. Методы find, count, replace, strip, upper, lower и другие
7.	Строки в Python. Срезы с двумя и тремя параметрами
8.	Списки в Python. Методы списков и операции со списками
9.	Цикл for и его особенности в Python. Функция range()
10.	В чем разница между списком и кортежем?
11.	В чем разница между методами экземпляра, класса и статическими методами в Python?
12.	Что такое функция в Python?
13.	Классы в Python. Перегрузка операторов.
14.	Какие библиотеки Python часто используются для работы с таблицами и агрегацией данных?
15.	Каким образом можно загрузить таблицу данных в Python?
16.	Какие методы агрегации данных можно применять к таблицам в Python?
17.	Какие методы визуализации данных доступны в Python?
18.	Какие статистические метрики и графики могут быть полезны при проведении первичного анализа данных?
19.	Что такое алгоритм k ближайших соседей (kNN)?
20.	Как работает алгоритм решающих деревьев?
21.	Что такое ансамбли решающих деревьев?
22.	Что такое логистическая регрессия?
23.	Как осуществить сравнение производительности алгоритмов классификации?
24.	Какие алгоритмы регрессии можно использовать в Python?
25.	Каким образом можно оценить качество работы алгоритма регрессии?
26.	Что такое переобучение в контексте алгоритмов регрессии?
27.	Как можно подобрать оптимальные параметры для алгоритма регрессии?
28.	Что такое классификация изображений?
29.	Какие методы и библиотеки можно использовать для классификации изображений в Python?

п/п	Вопросы к зачету
30.	Что такое сверточные нейронные сети (CNN) и как они применяются для классификации изображений?
31.	Как можно представить текст в векторном виде?
32.	Какие методы и библиотеки можно использовать для работы с текстами и их векторными представлениями в Python?
33.	Какие инструменты и библиотеки используются для разработки систем глубокого обучения?
34.	Классы в Python. Конструктор класса.
35.	Классы в Python. Наследование.
36.	Что такое локальные и глобальные переменные в Python?
37.	Какие типы данных поддерживаются в Python?
38.	Какие существуют фреймворки Python?
39.	Что такое словарь в Python?
40.	Как добавить новое значение в объект списка?
41.	Как создать пустой класс в Python?
42.	Как преобразовать список в строку?
43.	Как обрабатывать входные данные в Python?
44.	Что делает оператор with в Python?
45.	Что такое модули Python?
46.	Основные понятия объектно-ориентированного программирования
47.	Классы и объекты в Python
48.	Статические и динамические атрибуты класса
49.	Конструктор класса и инициализация экземпляра класса
50.	Уровни доступа атрибута и метода
51.	Множественное наследование в Python
52.	Полиморфизм с позиции переопределения методов базового класса в классе наследнике
53.	Веб-фреймворк как каркас для написания веб-приложений
54.	Архитектура и особенности веб-фреймворков
55.	Понятие проекта в веб-фреймворке
56.	Опишите способы создания проекта в Django
57.	В чем смысл использования шаблонов MVC?
58.	Django как MTV фреймворк
59.	Как создаются URL-адреса в Django?
60.	Обработка в Django Ошибки 404
61.	Что происходит, когда вы запрашиваете страницу в Django?
62.	Вывод динамических данных в Django
63.	Функции reverse(),redirect(),include()
64.	Что такое шаблоны Django?
65.	Использование системы шаблонов Django
66.	Теги шаблонов в Django
67.	Фильтры шаблонов в Django
68.	Понятие баз данных, их использование в веб-приложениях
69.	Типы отношений между объектами
70.	Опишите модельную часть архитектуры Django MTV
71.	Способы изменения схемы базы данных. Миграции
72.	Основы Django ORM
73.	Опишите методы извлечения записей из таблицы
74.	Опишите методы фильтрации данных

п/п	Вопросы к зачету
75.	Какие методы поиска данных Вам известны?
76.	Миграция данных в Django
77.	Основы формы Django
78.	Приложение администратора Django
79.	Добавление моделей в Django admin
80.	Основы фреймворка аутентификации Django
81.	Как осуществить регистрацию пользователей в Django?
82.	Как осуществить сборку проекта?
83.	Настройка итогового проекта.
84.	Что входит в отладку кода проекта?
85.	Как и где развернуть итоговый проект?
86.	Какие встроенные функции Python используются для работы со списками?
87.	Какой метод позволяет удалить элемент из списка в Python?
88.	Как создать и использовать генератор списков в Python?
89.	Как работают множества (set) в Python?
90.	Чем отличается deepcopy от copy в Python?
91.	Что такое итераторы и генераторы в Python?
92.	Как работает ключевое слово yield в Python?
93.	Что такое декораторы в Python и как их использовать?
94.	Как передавать функции в качестве аргументов в Python?
95.	Что такое замыкание (closure) в Python?
96.	Как работают исключения в Python и какие есть встроенные исключения?
97.	Что такое контекстный менеджер в Python?
98.	Как реализовать собственный контекстный менеджер?
99.	Что такое GIL (Global Interpreter Lock) в Python?
100.	Как работают процессы и потоки в Python?
101.	Какие библиотеки используются для работы с многопоточностью в Python?
102.	Как управлять версиями Python в проекте?
103.	Какие преимущества и недостатки у динамической типизации Python?
104.	Чем None отличается от False в Python?
105.	Как можно проверить тип переменной в Python?
106.	Как использовать списковые включения (list comprehension) в Python?
107.	Как работает ключевое слово pass в Python?
108.	Какие существуют способы форматирования строк в Python?
109.	Чем отличается sorted() от .sort()?
110.	Как работает функция zip() в Python?
111.	Как создать и использовать итератор в Python?
112.	Что такое enumerate() и как его использовать?
113.	Какие бывают способы записи функций в Python?
114.	Как определить, является ли строка палиндромом в Python?
115.	Чем отличается атрибут класса от атрибута экземпляра?
116.	Как создать абстрактный класс в Python?
117.	Что такое super() и как его использовать?
118.	Как определить, принадлежит ли объект определенному классу?
119.	Как реализовать перегрузку операторов в Python?
120.	Как работают магические методы в Python (__str__, __repr__ и др.)?
121.	Чем отличается __eq__ от __cmp__?
122.	Как можно запретить наследование в Python?

п/п	Вопросы к зачету
123.	Какие методы отвечают за сериализацию объектов в Python?
124.	В чем разница между статическими и классовыми методами?
125.	Как открыть файл на запись и на чтение в Python?
126.	Как обработать ошибки при работе с файлами?
127.	Что такое CSV-файл и как с ним работать в Python?
128.	Как получить размер файла в Python?
129.	Как работает метод .seek() при работе с файлами?
130.	Как эффективно работать с большими файлами в Python?
131.	Какие существуют типы регрессии в машинном обучении?
132.	Как оценить качество модели классификации?
133.	Чем отличается train_test_split от cross-validation?
134.	Как выбрать оптимальные гиперпараметры модели?
135.	В чем разница между GridSearchCV и RandomizedSearchCV?
136.	Как избежать переобучения модели?
137.	Что такое балансировка классов и зачем она нужна?
138.	Как работает система маршрутизации (urls.py) в Django?
139.	Как создать пользовательскую модель пользователя в Django?
140.	Что такое middleware в Django и как его использовать?
141.	Как защитить Django-приложение от SQL-инъекций?
142.	Чем class-based views отличаются от function-based views?
143.	Как развернуть Django-приложение на сервере?
144.	Как работает система сигналов (signals) в Django?
145.	Как оптимизировать SQL-запросы в Django ORM?
146.	Как проводить тестирование моделей Django-приложения?
147.	Как читать и записывать двоичные файлы в Python?
148.	Как работает библиотека shutil для управления файлами и директориями?
149.	Какие основные типы связей между моделями существуют в Django ORM и в чем их различия?
150.	Как работает система прав доступа (permissions) в Django и как ее настроить?
151.	Какие основные типы данных поддерживает Python?
152.	Чем отличаются неизменяемые (immutable) и изменяемые (mutable) типы данных?
153.	Как проверить тип переменной в Python?
154.	Как преобразовать один тип данных в другой?
155.	Что такое динамическая типизация в Python?
156.	Как создать и использовать аннотированные типы в Python?
157.	Чем отличается is от == при сравнении объектов?
158.	Что такое NoneType в Python?
159.	Как работает метод id() и что он показывает?
160.	Почему изменяемые типы данных нельзя использовать в качестве ключей словаря?
161.	Как использовать функцию input() для ввода данных?
162.	Чем отличается print() от sys.stdout.write()?
163.	Как работает множественное присваивание?
164.	Как считать несколько значений из одной строки ввода?
165.	Чем отличается = от += в Python?
166.	Как использовать sep и end в print()?
167.	Как вывести строку без переноса на новую строку?
168.	Как записать данные в файл с помощью print()?

п/п	Вопросы к зачету
169.	Как вывести переменную с текстом без пробела между ними?
170.	Что делает оператор := (моржовый оператор)?
171.	Как создать многострочную строку?
172.	Чем отличаются одинарные и двойные кавычки в Python?
173.	Что делает метод .format()?
174.	Как проверить, содержит ли строка определенное слово?
175.	Как использовать f-строки?
176.	Как разделить строку на части?
177.	Как заменить подстроку в строке?
178.	Как сделать строку полностью заглавной или строчной?
179.	Как удалить пробелы и спецсимволы из начала и конца строки?
180.	Как перевернуть строку?
181.	Как создать пустой список?
182.	Как скопировать список?
183.	Как добавить элемент в список?
184.	Как удалить элемент из списка?
185.	Чем отличаются .append() и .extend()?
186.	Как проверить, содержится ли элемент в списке?
187.	Как отсортировать список?
188.	Как объединить два списка?
189.	Как поменять местами два элемента списка?
190.	Как перевернуть список?
191.	Как создать кортеж?
192.	Чем отличается кортеж от списка?
193.	Можно ли изменить кортеж?
194.	Как распаковать кортеж в переменные?
195.	Как объединить два кортежа?
196.	Как проверить длину кортежа?
197.	Как проверить, содержится ли элемент в кортеже?
198.	Как преобразовать список в кортеж?
199.	Можно ли создать кортеж с одним элементом?
200.	Что произойдет при попытке изменить элемент кортежа?
201.	Как создать словарь?
202.	Как получить значение по ключу?
203.	Как проверить наличие ключа в словаре?
204.	Чем отличается .get() от [] при доступе к ключу?
205.	Как удалить элемент из словаря?
206.	Как получить список всех ключей словаря?
207.	Как объединить два словаря?
208.	Как перевернуть словарь (ключи ⇄ значения)?
209.	Как удалить все элементы словаря?
210.	Как создать словарь с значениями по умолчанию?
211.	Как написать условный оператор if?
212.	Чем elif отличается от if?
213.	Как работает тернарный оператор в Python?
214.	Как создать бесконечный цикл while?
215.	Чем for отличается от while?
216.	Как выйти из цикла досрочно?

п/п	Вопросы к зачету
217.	Как работает continue?
218.	Как использовать else с циклом?
219.	Как создать вложенные циклы?
220.	Что такое list comprehension?
221.	Как объявить функцию в Python?
222.	Чем отличается аргумент от параметра функции?
223.	Что такое позиционные и именованные аргументы?
224.	Как сделать параметр функции необязательным?
225.	Как вернуть несколько значений из функции?
226.	Чем *args отличается от **kwargs?
227.	Как использовать lambda-функции?
228.	Можно ли передавать функцию как аргумент?
229.	Что делает global внутри функции?
230.	Как документировать функции?
231.	Что такое класс?
232.	Как создать экземпляр класса?
233.	Чем __init__ отличается от других методов?
234.	Как создать метод внутри класса?
235.	Как объявить статический метод?
236.	Как определить метод класса?
237.	Что такое инкапсуляция?
238.	Что такое наследование?
239.	Что делает super()?
240.	Что такое полиморфизм?
241.	Как открыть файл на чтение?
242.	Как записать текст в файл?
243.	Как проверить, существует ли файл?
244.	Как прочитать файл построчно?
245.	Чем with open() лучше open()?
246.	Как определить размер файла?
247.	Как переместить курсор в файле?
248.	Как удалить файл?
249.	Как импортировать модуль?
250.	Как создать свой модуль?
251.	Что такое исключение?
252.	Как обработать исключение?
253.	Что делает try-except?
254.	Можно ли перехватить несколько исключений?
255.	Как создать свое исключение?
256.	Как использовать finally?
257.	Чем raise отличается от assert?
258.	Как обработать исключение в функции?
259.	Как повторить операцию после исключения?
260.	Что такое ZeroDivisionError?
261.	Как создать новый проект Django?
262.	Как создать приложение в Django?
263.	Что такое settings.py?
264.	Как настроить базу данных в Django?

п/п	Вопросы к зачету
265.	Как определить модель в Django?
266.	Как сделать миграции?
267.	Как создать суперпользователя?
268.	Как настроить админ-панель Django?
269.	Как добавить URL-маршрут?
270.	Что такое views.py?
271.	Как загрузить CSV-файл в Pandas?
272.	Как отобразить первые 5 строк таблицы?
273.	Как получить информацию о DataFrame?
274.	Как удалить столбец из DataFrame?
275.	Как заменить пропущенные значения?
276.	Как агрегировать данные?
277.	Как фильтровать строки?
278.	Как создать новый столбец?
279.	Как группировать данные?
280.	Как сортировать DataFrame?
281.	Как построить линейный график?
282.	Как построить столбчатую диаграмму?
283.	Как настроить заголовок и подписи осей?
284.	Как изменить цвет графика?
285.	Как сохранить график в файл?
286.	Как создать несколько подграфиков?
287.	Как отобразить легенду?
288.	Как построить scatter plot?
289.	Как добавить текст на график?
290.	Как изменить стиль графика?
291.	Что такое машинное обучение?
292.	Какие бывают типы ML?
293.	Что такое обучающая и тестовая выборка?
294.	Что такое overfitting?
295.	Как работает kNN?
296.	Как оценить модель?
297.	Как подобрать гиперпараметры?
298.	Как работает Random Forest?
299.	Что такое логистическая регрессия?
300.	Как работает нейронная сеть?

7.3.2. Критерии и нормы оценки

Семестр	Форма проведения промежуточной аттестации	Критерии и нормы оценки	
3	Зачет с оценкой (по накопительному рейтингу)	«Отлично»	рейтинговый балл 85-100
		«Хорошо»	рейтинговый балл 70-84
		«Удовлетворительно»	рейтинговый балл 55-69
		«Неудовлетворительно»	рейтинговый балл 0-54

8. Учебно-методическое и информационное обеспечение дисциплины

8.1. Обязательная литература

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно-методическое пособие, практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
1.	Амоа К. А.	Разработка программных пакетов на языке Python : учебное пособие / К. А. Амоа, Н. А. Рындин, Ю. С. Скворцов. — Воронеж : Воронежский государственный технический университет, ЭБС АСВ, 2020. — 61 с. — ISBN 978-5-7731-0887-0. — URL: https://www.iprbookshop.ru/108184.html	Учебное пособие	2020	ЭБС «IPRbooks»
2.	Букунов С. В.	Объектно ориентированное программирование на языке Python : учебное пособие / С. В. Букунов, О. В. Букунова. — Санкт-Петербург : Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2020. — 119 с. — ISBN 978-5-9227-1128-9. — URL: https://www.iprbookshop.ru/117194.html	Учебное пособие	2020	ЭБС «IPRbooks»
3.	Дроботун Н. В.	Алгоритмизация и программирование. Язык Python : учебное пособие / Н. В. Дроботун, Е. О. Рудков, Н. А. Баев. — Санкт-Петербург : Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2020. — 119 с. — ISBN 978-5-7937-	Учебное пособие	2020	ЭБС «IPRbooks»

		1829-5. — URL: https://www.iprbookshop.ru/102400.html			
4.	Сузи Р. А.	Язык программирования Python : учебное пособие / Р. А. Сузи. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 350 c. — ISBN 978-5-4497-0705-5. — URL: https://www.iprbookshop.ru/97589.html	Учебное пособие	2020	ЭБС «IPRbooks»
5.	Шелудько В. М.	Основы программирования на языке высокого уровня Python : учебное пособие / В. М. Шелудько. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2017. — 146 с. — ISBN 978-5-9275-2649-9. — URL: https://www.iprbookshop.ru/87461.html	Учебное пособие	2017	ЭБС «IPRbooks»
6.	Шелудько В. М.	Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули : учебное пособие / В. М. Шелудько. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2017. — 107 с. — ISBN 978-5-9275-2648-2. — URL: https://www.iprbookshop.ru/87530.html	Учебное пособие	2017	ЭБС «IPRbooks»
7.	Мохов В. А., Кузнецова А. В	Системы искусственного интеллекта: современные методы программной инженерии	Учебное пособие	2021	ЭБС «Лань»
8.	Ясницкий Л. Н.	Интеллектуальные системы	Учебник	2020	ЭБС «Лань»

8.2. Дополнительная литература

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно-методическое пособие, практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
1.	Баранов Р. Д.	Практические аспекты разработки веб-ресурсов : учебное пособие / Р. Д. Баранов, С. А. Иноземцева, А. А. Рябова. — Саратов : Вузовское образование, 2018. — 121 c. — ISBN 978-5-4487-0263-1. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : URL: https://www.iprbookshop.ru/75692.html	Учебное пособие	2018	ЭБС «IPRbooks»
2.	Меле А.	Django 2 в примерах / А. Меле ; перевод с английского Д. В. Плотниковой. — Москва : ДМК Пресс, 2019. — 408 с. — ISBN 978-5-97060-746-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/123711		2019	ЭБС «Лань»
3.	Барков И. А.	Объектно-ориентированное программирование : учебник / И. А. Барков. — Санкт-Петербург : Лань, 2022. — 700 с. — ISBN 978-5-8114-3586-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/206699	Учебник	2022	ЭБС «Лань»
4.	Васильев А. Н.	Python на примерах : практический курс по программированию / А. Н. Васильев. — 2-е изд. — Санкт-Петербург : Наука и Техника, 2017. —	Практический курс	2017	ЭБС «IPRbooks»

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно-методическое пособие, практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
		432 с. — ISBN 978-5-94387-741-4. — URL: https://www.iprbookshop.ru/73043.html			
5.	Елисеев А. И.	Разработка веб-приложений с использованием фреймворка Flask. В 2 частях. Ч.2 : учебное пособие / А. И. Елисеев, Ю. В. Минин, В. А. Гриднев. — Тамбов : Тамбовский государственный технический университет, ЭБС АСВ, 2021. — 80 с. — ISBN 978-5-8265-2438-1. —URL: https://www.iprbookshop.ru/123042.html	Учебное пособие	2021	ЭБС «IPRbooks»
6.	Елисеев А. И.	Разработка веб-приложений с использованием фреймворка Flask. В 2 частях. Ч. 1 : учебное пособие / А. И. Елисеев, Ю. В. Минин, В. А. Гриднев. — Тамбов : Тамбовский государственный технический университет, ЭБС АСВ, 2020. — 81 с. — ISBN 978-5-8265-2188-5 (ч.1), 978-5-8265-2187-8. — URL: https://www.iprbookshop.ru/115741.html	Учебное пособие	2020	ЭБС «IPRbooks»
7.	Ефромеев Н. М.	Основы web-программирования : учебное пособие / Н. М. Ефромеев, Е. В. Ефромеева. — Саратов : Вузовское образование, 2019. — 128 с. — ISBN 978-5-4487-0529-8. — URL: https://www.iprbookshop.ru/86300.html	Учебное пособие	2019	ЭБС «IPRbooks»

№ п/п	Авторы, составители	Заглавие (заголовок)	Тип (учебник, учебное пособие, учебно-методическое пособие, практикум, др.)	Год издания	Количество в научной библиотеке / Наименование ЭБС
8.	Сычев А. В.	Перспективные технологии и языки веб-разработки : практикум / А. В. Сычев. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019. — 493 c. — ISBN 978-5-4486-0507-9. — URL: https://www.iprbookshop.ru/79730.html	Учебное пособие	2019	ЭБС «IPRbooks»
9.	Сычев А. В.	Теория и практика разработки современных клиентских веб-приложений : учебное пособие / А. В. Сычев. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 482 c. — ISBN 978-5-4497-0943-1. — URL: https://www.iprbookshop.ru/102067.html	Учебное пособие	2021	ЭБС «IPRbooks»

8.3. Перечень профессиональных баз данных и информационных справочных систем:

№ пп	Наименование	Ссылка
1	Springer Nature (Полнотекстовая коллекция журналов)	https://www.springernature.com/gp/products
2	Springer eBooks (Полнотекстовая коллекция электронных книг издательства Springer Nature)	https://link.springer.com/
3	«Кодекс»	https://kodeks.ru/
4	Техэксперт	https://cntd.ru/

8.4. Перечень программного обеспечения

№ п/п	Наименование ПО	Реквизиты договора (дата, номер, срок действия)
1	Python 3.11	Free Software
2	Visual Studio Code 1.75	Free Software
3	Django 1.11.29	Free Software
4	Windows	2013г., № 00179-40183-81808-AAOEM, бессрочный
5	Microsoft Office 13	№61935138 от 28.05.2012 (бессрочный)

8.5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

№ п/п	Наименование оборудованных учебных кабинетов, лабораторий, мастерских и др. объектов для проведения практических и лабораторных занятий, помещений для самостоятельной работы обучающихся (номер аудитории)	Перечень основного оборудования
1	Компьютерный класс. Учебная аудитория для проведения занятий лекционного типа. Учебная аудитория для проведения занятий семинарского типа. Учебная аудитория для проведения лабораторных работ. Учебная аудитория для курсового проектирования (выполнения курсовых работ). Учебная аудитория для проведения групповых и индивидуальных консультаций Учебная аудитория для проведения занятий текущего контроля и промежуточной аттестации. (УЛК-408).	Компьютер (монитор 17”, системный блок Intel (R) Celeron (R) 2,66 GHz / 1 Gb / 80 Gb), маршрутизатор 2801 Router, коммутатор Catalyst, экран / интерактивная доска Smart Board TB, проектор Acer P1303W, стол преподавательский, столы ученические, столы компьютерные, стулья, доска аудиторная (маркерная).
2	Помещение для самостоятельной работы обучающихся (УЛК-105).	Столы, стулья, стеллажи (в т.ч. выставочные) с книгами, компьютеры, мобильные рабочие места.

№ п/п	Наименование оборудованных учебных кабинетов, лабораторий, мастерских и др. объектов для проведения практических и лабораторных занятий, помещений для самостоятельной работы обучающихся (номер аудитории)	Перечень основного оборудования
3	Помещение для самостоятельной работы обучающихся (УЛК-406).	Столы компьютерные, стулья, микрокомпьютеры raspberry pi 32 bit.