

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ
Декан факультета экономики менеджмента и
информационных технологий


С.А.Баркалов
«31» 08 2021 г.



РАБОЧАЯ ПРОГРАММА
дисциплины
« Программирование на языке Python»

Направление подготовки 09.03.02 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И
ТЕХНОЛОГИИ

Профиль Отраслевые информационные системы

Квалификация выпускника бакалавр
Нормативный период обучения 4 года
Форма обучения очная
Год начала подготовки 2021

Автор программы  /Ефимова О.Е./

И.о. заведующий кафедрой
Систем управления и
информационных
технологий в строительстве  /Десятирикова Е.Н./

Руководитель ОПОП  /Курипта О.В./

Воронеж 2021

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

Язык программирования Python – это универсальный язык программирования с динамической типизацией, который позволяет разрабатывать программы в соответствии с разными парадигмами программирования.

Целью освоения дисциплины является развитие у студентов систематизированных навыков и знаний в области программирования с использованием языка программирования Python, знакомство с актуальными парадигмами программирования, а также умение применять его для решения профессиональных задач.

1.2. Задачи освоения дисциплины

Основные задачи освоения дисциплины научить студентов:

- основным конструкциям на языке Python;
- технологии разработки программ на Python;
- работе со специализированными библиотеками для обработки, визуализации и анализа данных;
- созданию приложений на языке Python.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Программирование на языке Python» относится к дисциплинам блока ФТД.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Программирование на языке Python» направлен на формирование следующих компетенций:

ПК-3 - Способен разрабатывать технологии интеграции и осуществлять прототипирование компонентов информационных систем

ПК-6 - Способен разрабатывать программные решения, включая верификацию требований, проектирование, отладку, проверку работоспособности и модификацию программных решений по предметной области

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-3	знать объектно-ориентированные языки программирования; инструменты и методы взаимодействия с системами хранения и управления базами данных
	уметь применять современные методы и средства создания программного кода в зависимости от архитектуры и требований заказчика
	владеть навыками разработки технологии обмена данными между информационными системами и существующими системами
ПК-6	знать типовые решения и библиотеки, используемые при разработке программного обеспечения на языке Python
	уметь выбирать средства реализации требований к программному обеспечению на языке Python;

	кодировать и тестировать программное обеспечение на языке Python
	владеть навыками разработки и тестирования прототипа ИС в соответствии с требованиями

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Программирование на языке Python» составляет 2 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры
		5
Аудиторные занятия (всего)	36	36
В том числе:		
Лабораторные работы (ЛР)	36	36
Самостоятельная работа	36	36
Виды промежуточной аттестации - зачет	+	+
Общая трудоемкость:		
академические часы	72	72
зач.ед.	2	2

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

очная форма обучения

№ п/п	Наименование темы	Содержание раздела	Лаб. зан.	СРС	Всего, час
1	Основы программирования на языке Python	Язык программирования Python. Структура программы. Типы данных: простые и структурированные. Условный оператор. Оператор выбора. Циклы. Структурированные типы данных. Работа с файлами.	6	4	10
2	Разработка прикладных программ на Python	Библиотеки Python. Стандартная библиотека. Сетевые возможности языка Python. Использование языка Python для математических расчетов.	2	2	4
3	Объектно-ориентированное программирование в Python	Объекты и классы в Python. Особенности объектно-ориентированного подхода в Python. Концепция полностью объектно-ориентированного языка. Особенности наследования и полиморфизма в Python.	6	8	14
4	Функциональное программирование в Python	Элементы функционального программирования в Python. Генераторы.	6	6	12

		Итераторы.			
5	Разработка Web-приложения на Python	Настройка Web-сервера. Принципы написания Web-приложений на Python. Библиотеки, реализующие шаблоны Web-страниц.	8	8	16
6	Создание графических приложений в Python	Обзор графических библиотек: Tkinter, PyQt. Создание графического интерфейса на Tkinter. Принципы построения интерфейса с использованием библиотеки Tk. Основные объекты.	8	8	16
Итого			36	36	72

5.2 Перечень лабораторных работ

1. Основы программирования в Python. Элементарные вычисления в Python. Переменные в Python. Типы данных в Python: числовой, целочисленный, логический, строковый. Преобразование типов.
2. Основы программирования в Python. Основные типы и структуры данных.
3. Основы программирования в Python. Условные операторы и циклы.
4. Основы программирования в Python. Решение алгоритмических задач.
5. Структуры данных в Python. Списки, кортежи, множества, словари. Изменяемые и неизменяемые типы данных.
6. Использование функций в Python.
7. Работа с файлами в Python: открытие, изменение, сохранение. Разные форматы хранения данных: csv-файлы, json-файлы, txt-файлы.
8. Библиотеки Python. Библиотека NumPy для реализации математических объектов и вычислений.
9. Численные алгоритмы. Матричные вычисления. Модуль Numeric. Модуль LinearAlgebra. Модуль RandomArray.
10. Классы в Python. Определение данных, методов, операций. Наследование.
11. Классы в Python. Множественное наследование. Композиция при разработке классов.
12. Классы в Python. Абстракция и декомпозиция. Инкапсуляция.
13. Классы в Python. Полиморфизм. Отношения между классами.
14. Рекурсия, передача и возврат функций в качестве параметров, обработка последовательностей и итераторы, генератор.
15. Функциональная программа. Функции как параметры и результат.
16. Разработка Web-приложения с использованием универсальных Фреймворков, таких, как Django или Flask.
17. Создание приложений с GUI. Обзор графических библиотек: Tkinter, PyQt.
18. Создание приложений с GUI. Виды графиков, их корректное использование и интерпретация.

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины не предусматривает выполнение курсового проекта (работы) или контрольной работы.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ПК-3	знать объектно-ориентированные языки программирования; инструменты и методы взаимодействия с системами хранения и управления базами данных	Активная работа на лабораторных занятиях. Полнота и правильность ответов на проверочные вопросы в тестах.	Достаточно полное посещение лабораторных занятий. Выполнение и отчет лабораторных работ в установленный срок.	Недостаточное посещение лабораторных занятий. Не выполнение и не сдача в установленные сроки отчетов по лабораторным работам
	уметь применять современные методы и средства создания программного кода в зависимости от архитектуры и требований заказчика	Активная работа на лабораторных занятиях. Правильность ответов на теоретические вопросы при выполнении лабораторных работ.	Достаточно полное посещение лабораторных занятий. Выполнение и отчет лабораторных работ в установленный срок.	Недостаточное посещение лабораторных занятий. Не выполнение и не сдача в установленные сроки отчетов по лабораторным работам
	владеть навыками разработки технологии обмена данными между информационными системами и существующими системами	Выполнение самостоятельной работы. Качественное и правильное оформление отчетов при выполнении лабораторных работ.	Достаточно полное посещение лабораторных занятий. Выполнение и отчет лабораторных работ в установленный срок.	Недостаточное посещение лабораторных занятий. Не выполнение и не сдача в установленные сроки отчетов по лабораторным работам
ПК-6	знать типовые решения и библиотеки, используемые при разработке программного обеспечения на языке Python	Активная работа на лабораторных занятиях. Полнота и правильность ответов на проверочные вопросы в тестах.	Достаточно полное посещение лабораторных занятий. Выполнение и отчет лабораторных работ в установленный срок.	Недостаточное посещение лабораторных занятий. Не выполнение и не сдача в установленные сроки отчетов по лабораторным работам
	уметь выбирать средства реализации требований к программному обеспечению на языке Python; кодировать и тестировать программное обеспечение на языке Python	Активная работа на лабораторных занятиях. Правильность ответов на теоретические вопросы при выполнении лабораторных работ.	Достаточно полное посещение лабораторных занятий. Выполнение и отчет лабораторных работ в установленный срок.	Недостаточное посещение лабораторных занятий. Не выполнение и не сдача в установленные сроки отчетов по лабораторным работам
	владеть навыками разработки и тестирования прототипа ИС в соответствии с требованиями	Выполнение самостоятельной работы. Качественное и правильное оформление отчетов при выполнении лабораторных работ.	Достаточно полное посещение лабораторных занятий. Выполнение и отчет лабораторных работ в установленный срок.	Недостаточное посещение лабораторных занятий. Не выполнение и не сдача в установленные сроки отчетов по лабораторным работам

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 5 семестре для очной формы обучения по двухбалльной системе:

«зачтено»

«не зачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Не зачтено
ПК-3	знать объектно-ориентированные языки программирования; инструменты и методы взаимодействия с системами хранения и управления базами данных	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь применять современные методы и средства создания программного кода в зависимости от архитектуры и требований заказчика	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть навыками разработки технологии обмена данными между информационными системами и существующими системами	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ПК-6	знать типовые решения и библиотеки, используемые при разработке программного обеспечения на языке Python	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь выбирать средства реализации требований к программному обеспечению на языке Python; кодировать и тестировать программное обеспечение на языке Python	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть навыками разработки и тестирования прототипа ИС в соответствии с требованиями	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

1. Что напечатает эта программа?

```
sum = 0
n = 5
for i in range(1, n + 1):
    sum += i
print(sum)
```

a) 15
b) 12
c) 0
d) 20

2. Укажите НЕПРАВИЛЬНЫЙ способ получения списка из 100 нулей:

a) `a = [0] * 100`
b) `a = [0 for i in range(100)]`
c) `a = [0 for i in range(1,100)]`
d) `a = [0,0,0...,0].len(100)`

3. Что покажет этот код?

```
for i in range(5):
    if i % 2 == 0:
```

continue

print(i)

- a) Ошибку из-за неверного вывода
- b) Числа: 1, 3 и 5
- c) Ошибку, так как i не присвоена
- d) Числа: 1 и 3**
- e) Числа: 0, 2 и 4

4. Какая из следующих функций преобразует строку в список в Python?

- a) list(mystring)**
- b) tuple(mystring)
- c) eval(mystring)
- d) repr(mystring)

5. Какое из перечисленных значений может возвращать функция random.randint (4)?

- a) 3,4
- b) 4
- c) error**
- d) Ни один из вышеперечисленных

6. Какой из следующих операторов в Python оценивается как истина, если он не находит переменную в указанной последовательности, и ложь в противном случае?

- a) not in**
- b) is
- c) //
- d) **

7. Какая из следующих функций используется для проверки, все ли символы в строке находятся в нижнем регистре?

- a) istitle()
- b) isspace()
- c) isnumeric()
- d) islower()**

8. Какая из следующих переменных среды для Python является альтернативным путем поиска модуля?

- a) PYTHONHOME**
- b) PYTHONCASEOK
- c) PYTHONSTARTUP
- d) PYTHONPATH

9. Что из перечисленного является недопустимой переменной?

- a) _
- b) foo
- c) 1st_string**
- d) my_string_1

10. Какая функция перегружает оператор == ?

- a) __eq__()
- b) __isequal__()
- c) __eq__()**
- d) Ни один из вышеперечисленных

11. Что из следующего не является ключевым словом?

- a) pass
- b) nonlocal
- c) assert
- d) eval**

12. Какие типы элементов принимает random.shuffle ()?

- a) целые числа

- b) кортежи
 - c) **списки**
 - d) строки
13. Что возвращает `int(math.pow(3, 2))` ?
- a) error, third argument required
 - b) 6
 - c) error, too many arguments
 - d) **9**
14. Каково значение `x`, если `x = math.sqrt(4)` ?
- a) **2.0**
 - b) (2.0, -2.0)
 - c) 2
 - d) Ничего из вышеперечисленного
15. Предположим, `d = {"john": 40, "peter": 45}`, чтобы получить количество элементов в словаре, какую команду мы используем?
- a) `d.len()`
 - b) `size(d)`
 - c) **`len(d)`**
 - d) `d.size()`
16. Что из перечисленного является кортежем Python?
- a) `{}`
 - b) `{1, 2, 3}`
 - c) `[1, 2, 3]`
 - d) **`(1, 2, 3)`**
17. Как получить имя файла из файлового объекта (`fp`)?
- a) `fp.__name__()`
 - b) `self.__name__(fp)`
 - c) `fp.file(name)`
 - d) **`fp.name`**
18. Что будет напечатано?
- ```
kvps = {"user", "bill", "password", "hillary"}
print(kvps['password'])
```
- a) user
  - b) bill
  - c) password
  - d) Hillary
  - e) **Ничего. TypeError**
19. Что выведет следующий фрагмент кода?
- ```
x = 4.5
y = 2
print(x // y)
```
- a) **2**
 - b) 2.25
 - c) 9
 - d) 20.25
 - e) 21
20. Что делает следующий код?
- ```
def a(b, c, d): pass
```
- a) Определяет список и инициализирует его.
  - b) **Определяет функцию, которая ничего не делает.**
  - c) Определяет функцию, которая передает параметры.
  - d) Определяет пустой класс.



21. Для чего в Python используется встроенная функция enumerate()?
- Для определения количества элементов последовательности.
  - Для одновременного итерирования по самим элементам и их индексам.**
  - Для сортировки элементов по значениям id.
22. Как вывести список методов и атрибутов объекта x
- help(x)
  - info(x)
  - ?x
  - dir(x)**
23. Как можно более кратко представить следующую запись?
- ```
if X:
    A = Y
else:
    A = Z
```
- A = Y if Z else Y
 - A = Y if X else Z**
 - A = X if Z else Y
 - A = X if Y else Z
24. При объявлении класса с помощью оператора class что пишется в круглых скобках после имени класса?
- Имена аргументов, принимаемых методом __init__.
 - Имена принимаемых классом аргументов.
 - Имена суперклассов, если класс наследуется от одного или нескольких классов.**
 - Имена классов, порождаемых данным классом.
25. Какую роль в описании метода класса выполняет декоратор @property?
- Декорированный метод становится статическим, экземпляр не передаётся.
 - Декорированный метод становится методом класса: метод получает класс, а не экземпляр.
 - Значение, возвращаемое декорированным методом, вычисляется при извлечении. Можно обратиться к методу экземпляра, как к атрибуту.**

7.2.2 Примерный перечень заданий для решения стандартных задач

- Составить программу калькулятор для выполнения следующих операций: +, -, *, /.
- Составить программу для вычисления выражения

$$S = \sum_{i=1}^n \frac{x^i}{(2 \cdot i)!}$$

и найти наименьшее значение N, при котором данная сумма превышает заданную пользователем величину C.

3. Вычислить множество значений функции $y = ax^2 + bx + c$ (построить таблицу значений) для аргумента x, изменяющегося в диапазоне от -10 до 10 с шагом 2, при заданных a, b, c.

4. Найти $\sqrt[3]{a}$ как предел последовательности x_0, x_1, x_2, \dots , где $x_0 = a/3$, а каждое следующее x_{n+1} получается из предыдущего по формуле:

$$x_{n+1} = \frac{2}{3} \left(x_n + \frac{a}{2x_n^2} \right), \quad n = 0, 1, 2, \dots$$

5. Пусть дана последовательность из 20 целых чисел. Определите количество инверсий в этой последовательности (т.е. таких пар элементов, в которых большее число находится слева от меньшего: $x_i > x_j$ при $i > j$).

7.2.3 Примерный перечень заданий для решения прикладных задач

1. Дано действительное положительное число a и целое неотрицательное число n . Вычислите a^n , не используя циклы, возведение в степень через `**` и функцию `math.pow()`, а используя рекуррентное соотношение $a^n = a \cdot a^{n-1}$. Решение оформите в виде функции `power(a, n)`.

2. Дан список целых чисел. Требуется “сжать” его, переместив все ненулевые элементы в левую часть списка, не меняя их порядок, а все нули - в правую часть. Порядок ненулевых элементов изменять нельзя, дополнительный список использовать нельзя, задачу нужно выполнить за один проход по списку. Распечатайте полученный список.

Пример ввода: 4 0 5 0 0 3 2 0 5 0

Пример вывода: 4 5 3 2 5 0 0 0 0 0

3. Составить программу, моделирующую алгоритм группового сжатия данных RLE – замену нескольких подряд стоящих одинаковых элементов списка – парой вида: элемент, количество, а также программу обратного преобразования.

4. Игра «Цепочка». В заданном списке слов найти все цепочки слов, в которых каждое слово (кроме первого) отличается от предыдущего ровно одним символом:

Пример ввода: [муха, дуб, муза, кус, куб, сук, лупа, луза, вино, лето, кэб, лото]

Пример вывода: [[муха, муза, луза, лупа], [дуб, куб, кус, кэб], [сук], [вино], [лето, лото]]

5. Написать функцию `XOR_cipher`, принимающая 2 аргумента: строку, которую нужно зашифровать, и ключ шифрования, которая возвращает строку, зашифрованную путем применения функции XOR (^) над символами строки с ключом. Написать также функцию `XOR_uncipher`, которая по зашифрованной строке и ключу восстанавливает исходную строку.

7.2.4 Примерный перечень вопросов для подготовки к зачету

1. Язык программирования Python. Структура программы. Синтаксис языка Python.
2. Типы данных: простые и структурированные. Описание встроенных типов данных.
3. Основные алгоритмические конструкции. Условный оператор. Оператор выбора.

Циклы.

4. Работа с файлами.
5. Стили программирования Python.
6. Модули в Python. Встроенные функции.
7. Работа с данными в различных форматах.
8. Библиотеки Python. Стандартная библиотека.
9. Библиотеки по работе с текстами.
10. Регулярные выражения.
11. Сетевые возможности языка Python.
12. Использование языка Python для математических расчетов.
13. Модуль `Numeric`.
14. Модуль `LinearAlgebra`.
15. Модуль `RandomArray`.
16. Основные понятия ООП. Объекты и классы в Python. Отношения между классами.
17. Абстракция и декомпозиция.
18. Инкапсуляция. Полиморфизм.
19. Особенности наследования и полиморфизма в Python.
20. Элементы функционального программирования в Python. Генераторы. Итераторы.
21. Функциональная программа. Рекурсия.
22. Функции как параметры и результат.
23. Настройка Web-сервера.
24. Принципы написания Web-приложений на Python. CGI-сценарии.
25. Библиотеки, реализующие шаблоны Web-страниц.
26. Обзор графических библиотек: `Tkinter`, `PyQT`.
27. Основы Tk.

28. Основы PyQt.
29. Создание графического интерфейса на Tkinter.
30. Принципы построения интерфейса с использованием библиотеки Tk.

7.2.5 Примерный перечень заданий для подготовки к экзамену

Не предусмотрено учебным планом

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Зачет может проводиться по итогам текущей успеваемости и сдачи отчета о выполненных индивидуальных практических заданиях на ЭВМ, и (или) путем организации специального опроса, проводимого в устной и (или) письменной форме, а также в виде тестирования.

Во время проведения зачета студенты могут пользоваться программой дисциплины.

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Основы программирования на языке Python	ПК-3, ПК-6	Тест, устный опрос, защита лабораторных работ
2	Разработка прикладных программ на Python	ПК-3, ПК-6	Тест, устный опрос, защита лабораторных работ
3	Объектно-ориентированное программирование в Python	ПК-3, ПК-6	Тест, устный опрос, защита лабораторных работ
4	Функциональное программирование в Python	ПК-3, ПК-6	Тест, устный опрос, защита лабораторных работ
5	Разработка Web-приложения на Python	ПК-3, ПК-6	Тест, устный опрос, защита лабораторных работ
6	Создание графических приложений в Python	ПК-3, ПК-6	Тест, устный опрос, защита лабораторных работ

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методике выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методике выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка

решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

8. УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

1. Буйначев С.К. Основы программирования на языке Python: учебное пособие / Буйначев С.К., Боклаг Н.Ю. – Екатеринбург: Уральский федеральный университет, ЭБС АСВ, 2014. – 92 с. – ISBN 978-5-7996-1198-9. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: <https://www.iprbookshop.ru/66183.html>

2. Васильев А.Н. Python на примерах: практический курс по программированию / Васильев А.Н. – Санкт-Петербург: Наука и Техника, 2017. – 432 с. – ISBN 978-5-94387-741-4. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: <https://www.iprbookshop.ru/73043.html>

3. Сузи Р.А. Язык программирования Python: учебное пособие / Сузи Р.А. – Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. – 350 с. – ISBN 978-5-4497-0705-5. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: <https://www.iprbookshop.ru/97589.html>

4. Шелудько В.М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули: учебное пособие / Шелудько В.М. – Ростов-на-Дону, Таганрог: Издательство Южного федерального университета, 2017. – 107 с. – ISBN 978-5-9275-2648-2. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: <https://www.iprbookshop.ru/87530.html>

5. Дроботун Н.В. Алгоритмизация и программирование. Язык Python: учебное пособие / Дроботун Н.В., Рудков Е.О., Баев Н.А. – Санкт-Петербург: Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2020. – 119 с. – ISBN 978-5-7937-1829-5. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: <https://www.iprbookshop.ru/102400.html>

6. Маккинли Уэс Python и анализ данных / Маккинли Уэс. – Саратов: Профобразование, 2019. – 482 с. – ISBN 978-5-4488-0046-7. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: <https://www.iprbookshop.ru/88752.html>

7. Северенс Ч. Введение в программирование на Python / Ч. Северенс. – 2-е изд., – Москва: Национальный Открытый Университет «ИНТУИТ», 2016. – 231 с.: Режим доступа: [Электронный ресурс] – URL: <https://biblioclub.ru/index.php?page=book&id=429184>

8. Хахаев И. А. Практикум по алгоритмизации и программированию на Python: учебное пособие / И. А. Хахаев. – 2-е изд., – Москва: Национальный Открытый Университет «ИНТУИТ», 2016. – 179 с.: – Режим доступа: [Электронный ресурс] – URL: <https://biblioclub.ru/index.php?page=book&id=429256>

9. Амоа К.А. Разработка программных пакетов на языке Python: учебное пособие / Амоа К.А., Рындин Н.А., Скворцов Ю.С. – Воронеж: Воронежский государственный технический университет, ЭБС АСВ, 2020. – 61 с. – ISBN 978-5-7731-0887-0. – Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. – URL: <https://www.iprbookshop.ru/108184.html>

10. Sweigart A. Разработка компьютерных игр на языке Python / A. Sweigart. – 2-е изд., – Москва: Национальный Открытый Университет «ИНТУИТ», 2016. – 505 с.: Режим доступа: [Электронный ресурс]. – URL: <http://biblioclub.ru/index.php?page=book&id=429009>

8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:

- Персональные компьютеры с операционной системой Windows 10
- ЭИОС ВГТУ
- Python
- MS VISUAL STUDIO (актуальная версия)
- OpenOffice
- LibreOffice
- Браузеры Internet: Google Chrome, Mozilla Firefox, IE
- WinDjView
- Adobe Acrobat Reader
- 7zip

Информационно-справочные системы и профессиональные базы данных:

- <http://www.python.org>
- <http://www.edu.ru>
- <http://window.edu.ru>
- <https://habr.com/ru/all>
- <https://www.lektorium.tv>
- <http://www.consultant.ru>
- <https://gostexpert.ru>
- <http://www.rugost.com>
- <https://www.elibrary.ru>

Внеаудиторная работа обучающихся сопровождается методическим обеспечением, доступом к электронно-библиотечной системе (<https://cchgeu.ru/university/library>), содержащей издания по изучаемой дисциплине.

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

При освоении дисциплины для выполнения лабораторных работ требуются компьютерные классы с комплектом лицензионного программного обеспечения (при использовании электронных изданий – компьютерный класс с выходом в Интернет), оснащённые мультимедийным оборудованием.

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Программирование на языке Python» проводятся лабораторные работы.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Вид учебных занятий	Деятельность студента
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед зачетом три дня эффективнее всего использовать для повторения и систематизации материала.