МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
СТАРООСКОЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ ИМ. А.А. УГАРОВА
(филиал) федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский технологический университет «МИСИС»

СТИ НИТУ «МИСИС»

Рабочая программа утверждена
решением Ученого совета
СТИ НИТУ «МИСИС»
от «20» июня 2023 г.
протокол № 5

Рабочая программа дисциплины
Основы программирования на Python

Закреплена за кафедрой	**Кафедра автоматизированных и информационных систем управления**
Направление подготовки	13.03.02 Электроэнергетика и электротехника
Профиль	Электропривод и автоматика
Квалификация	**Бакалавр**
Форма обучения	**Очная**
Общая трудоемкость	**3 ЗЕТ**

Формы контроля в семестрах:

Часов по учебному плану 108

в том числе:

зачет, 3

аудиторные занятия 34

самостоятельная работа 74

часов на контроль -

Распределение часов дисциплины по семестрам

Семестр	3		Итого	
Вид занятий	УП	РП	УП	РП
Лабораторные	34	34	34	34
Контактная работа	34	34	34	34
Сам. работа	74	74	74	74
Итого:	108	108	108	108

Год набора 2023

Программу составил:
Ст. преподаватель каф. АИСУ, кандидат технических наук, Цыганков Юрий Александрович
Должность, уч.ст., уч.зв ФИО полностью *подпись*

Рабочая программа дисциплины
Основы программирования на Python
наименование

Разработана в соответствии с ОС ВО НИТУ МИСИС:
Самостоятельно устанавливаемый образовательный стандарт высшего образования – бакалавриат Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский технологический университет «МИСИС» по направлению подготовки 13.03.02 Электроэнергетика и электротехника (приказ от 05.03.20 г. № 95 о.в.)

Составлена на основании учебного плана 2023 года набора.
13.03.02 Электроэнергетика и электротехника
Профиль: Электропривод и автоматика, утвержденного Ученым советом СТИ НИТУ «МИСИС» 20.06.2023 ., протокол № 5.

Рабочая программа одобрена на заседании кафедры
Автоматизированные и информационные системы управления

Протокол от «24» мая 2023 г. № 05

И.о. зав. кафедрой АИСУ *подпись* Д.А. Полещенко

«24» мая 2023 г.

Руководитель ОПОП ВО
И.о. зав. кафедрой АИСУ,
кандидат технических наук, доцент *подпись* Д.А. Полещенко

«24» мая 2023 г.

1. ЦЕЛИ ОСВОЕНИЯ
Цель изучения дисциплины - овладеть базовыми особенностями программирования на высокоуровневом языке Python и принципами его практического применения. Задачи изучения дисциплины: 1. Освоение принципов работы с языком программирования Python, изучение типов данных, базовых структур. 2. Формирование навыков работы с алгоритмами и структурами данных для реализации различных алгоритмов на высокоуровневом языке программирования. 3. Научить применять полученные навыки работы с языком Python для решению прикладных задач.

2. МЕСТО В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ	
Цикл (раздел) ОП:	Б1.В.ДВ
2.1	**Требования к предварительной подготовке обучающихся:**
2.1.1	Компьютерное обеспечение специальности
2.1.2	Информатика
2.1.3	Современные информационные технологии
2.2	**Дисциплины и практики, для которых освоение данной дисциплины необходимо как предшествующее:**
2.2.1	Интеллектуальные системы управления
2.2.3	Роботизированные комплексы и системы

3. ИНДИКАТОРЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ, СОВМЕЩЕННЫЕ С РЕЗУЛЬТАТАМИ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ	
УК-1: Способен осуществлять поиск, критический анализ и синтез информации, умение анализировать процессы и системы с использованием соответствующих аналитических, вычислительных и экспериментальных методов, применять системный подход для решения поставленных задач	
Знать:	УК-1-З1: Знать базовые средства языка Python для анализа возможности решения поставленной отраслевой задачи
Уметь:	УК-1-У1: Уметь систематизировать и анализировать информацию о процессе для разработки решения с использованием Python
Владеть:	УК-1-В1: Владеть навыками поиска базового каркаса кода для реализации конкретного решения
ОПК-1: Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности, разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения, осуществлять поиск, обработку и анализ информации из различных источников	
Знать:	ОПК-1-З1: Знать базовые принципы, алгоритмы, структуры языка Python для применения их при решении практических задач
Уметь:	ОПК-1-У1: Уметь использовать средства языка Python для реализации конкретных алгоритмов
Владеть:	ОПК-1-В1: Владеть навыками декомпозиции практической задачи для реализации ее решения средствами Python
ПК-1: Способен: **- рассчитывать режимы работы объектов профессиональной деятельности;** **- применять методы и технические средства эксплуатационных испытаний и диагностики электроэнергетического и электротехнического оборудования;** **- оценивать техническое состояние и остаточный ресурс оборудования;** **- составлять и оформлять типовую техническую документацию;** **- определять параметры оборудования объектов профессиональной деятельности;** **- обеспечивать требуемые режимы работы объектов профессиональной деятельности и заданные параметры технологического процесса;** **- участвовать в выполнении ремонтов оборудования по заданной методике;** **- составлять заявки на оборудование и запасные части и подготавливать техническую документацию на ремонт.**	
Знать:	ПК-1-З1: Знать основные требования, синтаксис и принципы разработки программного обеспечения на языке Python
Уметь:	ПК-1-У1: Уметь выполнять операции с имеющимися данными для определения режимов процесса и оборудования
Владеть:	ПК-1-В1: Владеть навыками базовой обработки данных для планирования работ с оборудованием

4. СТРУКТУРА И СОДЕРЖАНИЕ

Код занятия	Наименование разделов и тем /вид занятия/	Семестр	Количество часов	Компетенции	Литература и электронные ресурсы	Примечание
1.1	Ввод/вывод данных. Переменные /лр/	3	8	УК-1-З1 УК-1-В1 ОПК-1-З1 ПК-1-З1 ПК-1-У1	Л1.1 Л1.2 Л2.1 Л2.2	Текущий контроль: устный опрос

1.2	Условный оператор. Вложенные условия. Операторы сравнения /лр/	3	8	УК-1-З1 УК-1-В1 ОПК-1-З1 ПК-1-З1 ПК-1-У1	Л1.1 Л1.2 Л2.1 Л2.2	Текущий контроль: устный опрос
1.3	Числовые типы данных и операции с ними. Библиотеки в Python /лр/	3	6	УК-1-З1 УК-1-У1 УК-1-В1 ОПК-1-З1 ПК-1-З1 ПК-1-У1	Л1.1 Л1.2 Л2.1 Л2.2	Текущий контроль: устный опрос
1.4	Циклы в Python /лр/	3	6	УК-1-З1 УК-1-У1 УК-1-В1 ОПК-1-З1 ОПК-1-В1 ПК-1-У1 ПК-1-В1	Л1.1 Л1.2 Л2.1 Л2.2	Текущий контроль: устный опрос
1.5	Строки в Python /лр/	3	6	УК-1-З1 УК-1-В1 ОПК-1-З1 ПК-1-З1 ПК-1-У1	Л1.1 Л1.2 Л2.1 Л2.2	Текущий контроль: устный опрос
1.6	Функции в Python /ср/	3	24	УК-1-З1 УК-1-У1 УК-1-В1 ОПК-1-З1 ОПК-1-У1 ОПК-1-В1 ПК-1-З1 ПК-1-У1 ПК-1-В1	Л1.1 Л1.2 Л2.1 Л2.2 Э1	Промежуточный контроль: защита ДЗ
1.7	Проработка материала для самостоятельного изучения, подготовка к выполнению лабораторных работ /ср/	3	50	УК-1-З1 УК-1-У1 УК-1-В1 ОПК-1-З1 ОПК-1-У1 ОПК-1-В1 ПК-1-З1 ПК-1-У1 ПК-1-В1	Л1.1 Л1.2 Л2.1 Л2.2 Э1	Текущий контроль: устный опрос

5. ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛОВ

5.1. Вопросы для самостоятельной подготовки к экзамену (зачёту с оценкой)

Экзамен не предусмотрен

5.2. Перечень работ, выполняемых по дисциплине

По дисциплине предусмотрено: выполнение 1 домашнего задания; выполнение и защита 5 лабораторных работ, выполнение тестирований по всем темам разделов дисциплины.

5.2.1 Пример условия домашнего задания и вопросов для его защиты

Домашнее задание № 1. Функции в Python (УК-1-З1, УК-1-У1, УК-1-В1, ОПК-1-З1, ОПК-1-У1, ОПК-1-В1, ПК-1-З1, ПК-1-У1, ПК-1-В1).

1. Создать функцию, в которую передается определенное число целочисленных аргументов. Данная функция должна определить сумму значений вводимых аргументов.

2. Простое число — это число, которое делится без остатка на само себя и 1. Реализуйте функцию, которая в качестве аргумента принимает целое число от 1 до 20 и возвращает "YES", если аргумент является простым числом, "NO" - если аргумент не является простым числом и сообщение "NaN" – если аргумент не в пределах диапазона.

Вопросы для защиты домашнего задания № 1 (УК-1-З1, ОПК-1-З1, ПК-1-З1):

1. Дайте определение функции в Python.
2. Виды функций.
3. Этапы работы с функцией.
4. Именование функций.
5. Аргументы функций.

6. Область видимости переменных.

5.2.2 Пример вопросов для защиты лабораторных работ

Лабораторная работа № 1. Ввод/вывод данных. Переменные. (УК-1-З1, УК-1-В1, ОПК-1-З1, ПК-1-З1, ПК-1-У1)
Вопросы для защиты (УК-1-З1, ОПК-1-З1, ПК-1-З1):

1. Способы ввода данных в Python.
2. Способы вывода данных в Python.
3. Необязательные параметры функции print().
4. Необязательные параметры функции input().
5. Объявление переменных.

Лабораторная работа №2. Условный оператор. Вложенные условия. Операторы сравнения. (УК-1-З1, УК-1-В1, ОПК-1-З1, ПК-1-З1, ПК-1-У1)
Вопросы для защиты (УК-1-З1, ОПК-1-З1, ПК-1-З1):

1. Условный оператор if.
2. Операторы сравнения в Python.
3. Логические операторы в Python.
4. Каскадный условный оператор.
5. Способы каскадирования условий.

Лабораторная работа №3. Числовые типы данных и операции с ними. Библиотеки в Python. (УК-1-З1, УК-1-У1, УК-1-В1, ОПК-1-З1, ПК-1-З1, ПК-1-У1)
Вопросы для защиты (УК-1-З1, ОПК-1-З1, ПК-1-З1):

1. Тип данных int.
2. Тип данных float.
3. Преобразование типов числовых данных в Python.
4. Явное и неявное преобразование типов в Python.
5. Способы импорта библиотек в Python.

Лабораторная работа № 4. Циклы в Python. (УК-1-З1, УК-1-У1, УК-1-В1, ОПК-1-З1, ОПК-1-В1, ПК-1-У1, ПК-1-В1)
Вопросы для защиты (УК-1-З1, ОПК-1-З1):

1. Циклы. Типы циклов.
2. Цикл for в Python.
3. Цикл while в Python
4. Функция range () в Python.
5. Множественное условие в циклах.

Лабораторная работа №5. Строки в Python. (УК-1-З1, УК-1-В1, ОПК-1-З1, ПК-1-З1, ПК-1-У1)
Вопросы для защиты (УК-1-З1, ОПК-1-З1, ПК-1-З1):

1. Строковый тип данных.
2. Ввод/вывод строковых переменных.
3. Операции со строками в Python.
4. Преобразование с типом int.
5. Преобразование с типом float.

Примерный перечень тестовых вопросов и заданий
(текущий контроль успеваемости)
(УК-1-З1, УК-1-У1, УК-1-В1, ОПК-1-З1, ОПК-1-У1, ОПК-1-В1, ПК-1-З1, ПК-1-У1, ПК-1-В1)

1) Как получить данные от пользователя?
   i) Использовать метод read()
   ii) Использовать метод get()
   iii) Использовать метод cin()
   iv) Использовать метод readLine()
   v) Использовать метод input()
2) Какая функция выводит что-либо в консоль (на экран монитора)?
   i) out()
   ii) log()
   iii) print()
   iv) write()
3) Какие существуют типы переменных у чисел(выбрать несколько вариантов):
   i) float
   ii) list
   iii) num

iv) int
v) integer

4) Переменная int:
i) вещественная переменная
ii) символьная строка
iii) логическая переменная
iv) целая переменная

5) Переменная float:
i) целая переменная
ii) вещественная переменная
iii) логическая переменная

6) Язык Python:
(1) имеет возможность интеграции с другими языками программирования
(2) является низкоуровневым языком программирования
(3) является высокоуровневым языком программирования
(4) имеет статическую типизацию
(5) имеет динамическую типизацию

7) Язык Python:
(1) подходит для написания сценариев, связывающих остальные компоненты проекта
(2) предназначен для решения задач требовательных к производительности кода
(3) предназначен для быстрой разработки приложений
(4) является компилируемым языком
(5) является интерпретируемым языком

8) Язык Python:
(1) сильно типизирован
(2) слабо типизирован
(3) поощряет повторное использование кода
(4) является низкоуровневым языком программирования
(5) является высокоуровневым языком программирования

9) Язык Python:
(1) имеет сложный и обширный синтаксис
(2) предназначен для быстрой разработки приложений и написания сценариев
(3) имеет статическую типизацию
(4) сильно типизирован
(5) слабо типизирован

10) Язык Python:
(1) является компилируемым языком
(2) является интерпретируемым языком
(3) является низкоуровневым языком программирования
(4) является высокоуровневым языком программирования
(5) является универсальным языком программирования

11) Язык Python:
(1) имеет простой синтаксис
(2) имеет обширный и сложный синтаксис
(3) предназначен для решения задач, требовательных к производительности кода
(4) предназначен для быстрой разработки приложений
(5) поощряет повторное использование кода

12) Язык Python поддерживает следующие парадигмы программирования:
(1) автоматное программирование
(2) структурное программирование
(3) модульное программирование
(4) процедурное программирование
(5) логическое программирование

13) Язык Python поддерживает следующие парадигмы программирования:
(1) программирование в ограничениях
(2) аспектно-ориентированное программирование

(3) объектно-ориентированное программирование
(4) структурное программирование
(5) модульное программирование

14) Язык Python поддерживает следующие стили программирования:
(1) процедурное программирование
(2) декларативное программирование
(3) объектно-ориентированное программирование
(4) функциональное программирование
(5) аспектно-ориентированное программирование

15) Привести переменную x к типу числа с плавающей точкой можно следующим способом:
(1) (float)x
(2) x.float()
(3) x.__float__()
(4) float(x)
(5) x.__class__=float

16) Python поддерживает следующие простые скалярные типы:
(1) целое
(2) беззнаковое целое
(3) булево (логическое)
(4) число с плавающей точкой
(5) строка

17) Тип переменной во время выполнения скрипта определяется по следующим правилам:
(1) тип переменной явно указывается при определении переменной и не изменяется в процессе выполнения скрипта
(2) тип переменной явно указывается при определении переменной и изменяется только при приведении этой переменной к другому типу
(3) тип переменной определяется типом первого значения, которое было ей присвоено, и далее не изменяется
(4) тип переменной изменяется при приведении этой переменной к другому типу, а также может изменяться в зависимости от контекста использования этой переменной
(5) тип переменной изменяется при присваивании, но не может изменяться в зависимости от контекста использования этой переменной

18) Тип переменной x можно узнать следующим способом:
(1) type(x)
(2) x.type()
(3) x.__class__ == имя_типа
(4) x.__class__ is имя_типа
(5) x.__class__ == "имя типа"

19) Из приведенных ниже высказываний укажите истинное:
(1) перед использованием переменной она должна быть определена с указанием типа и инициализирована каким-либо значением
(2) перед использованием переменной она должна быть определена с указанием типа
(3) перед использованием переменной она должна быть инициализирована каким-либо значением
(4) переменные не обязаны быть инициализированы каким-либо значением перед использованием, а тип переменной определяется в зависимости от контекста
(5) перед использованием переменной она будет автоматически проинициализирована значением по умолчанию

19) Укажите результат выполнения скрипта:
```
foo = (1,)
bar = foo
bar += (1,)
print (foo)
```

(1) (1,)
(2) (1,1)
(3) (1)
(4) скрипт не будет выполнен, так как содержит ошибки

20) Укажите результат выполнения скрипта:
```
x = 'Printing'
x[0] = 'p'
print (x)
```

Ответ:
(1) Printing
(2) printing
(3) p
(4) скрипт не будет выполнен, так как содержит ошибки

21) Укажите результат выполнения скрипта:
```
x = {0:'P',1:'r',2:'i',3:'n',4:'t'}
x[0] = 'p'
print (x)
```

Ответ:
(1) {0: 'P', 1: 'r', 2: 'i', 3: 'n', 4: 't'}
(2) {0: 'p', 1: 'r', 2: 'i', 3: 'n', 4: 't'}
(3) {0: 'p'}
(4) скрипт не будет выполнен, так как содержит ошибки

22) Укажите результат выполнения скрипта:
```
x = ('P','r','i','n','t')
x[0] = 'p'
print (x)
```

Ответ:
(1) ('P', 'r', 'i', 'n', 't')
(2) ('p', 'r', 'i', 'n', 't')
(3) ('p')
(4) скрипт не будет выполнен, так как содержит ошибки

23) Укажите результат выполнения скрипта:
```
x = 'Hello'
x[0] = 'h'
print (x)
```

Ответ:
(1) Hello
(2) hello
(3) h
(4) скрипт не будет выполнен, так как содержит ошибки

24) Укажите результат выполнения скрипта:
```
x = {0:'H',1:'e',2:'l',3:'l',4:'o'}
x[0] = 'h'
print (x)
```

Ответ:
(1) {0: 'H', 1: 'e', 2: 'l', 3: 'l', 4: 'o'}
(2) {0: 'h', 1: 'e', 2: 'l', 3: 'l', 4: 'o'}
(3) {0: 'h'}
(4) скрипт не будет выполнен, так как содержит ошибки

25) Укажите результат выполнения скрипта:
```
x = ('H','e','l','l','o')
x[0] = 'h'
print (x)
```

Ответ:
(1) ('H', 'e', 'l', 'l', 'o')

(2) ('h', 'e', 'l', 'l', 'o')
(3) ('h')
(4) скрипт не будет выполнен, так как содержит ошибки

26) В языке Python существуют следующие арифметические операции:

(1) деление с остатком (%)
(2) целочисленное деление (%)
(3) деление по модулю (%)
(4) нахождение дробной части от результата деления (%)
(5) возведение в степень (%)

27) В языке Python существуют следующие арифметические операции:
(1) получение целого результата при делении целых чисел (//)
(2) возведение в степень (**)
(3) возведение в степень (^)
(4) деление по модулю (%)
(5) деление по модулю (\ )

28) В языке Python существуют следующие операции сравнения:
(1) равно (=)
(2) не равно (!=)
(3) много больше (>>)
(4) меньше или равно (<=)
(5) больше или равно (=>)

29) В языке Python существуют следующие операции для работы со строками:
(1) конкатенация (+)
(2) конкатенация (,)
(3) поиск количества вхождений подстроки (%)
(4) объединение (*)
(5) повторение (*)

30) В языке Python существуют следующие циклы:
(1) do-while
(2) while
(3) for
(4) foreach
(5) iterate

31) Для контроля над выполнением условного ветвления (if) можно использовать:
(1) elseif
(2) endif
(3) elif
(4) continue
(5) ничего из вышеперечисленного

32) В языке Python циклом с предусловием является:
(1) do-while
(2) iterate
(3) while
(4) for
(5) в языке Python цикла с предусловием не существует

33) Инструкция for в языке Python:
(1) всегда перебирает элементы арифметической прогрессии, причем шаг и условия выхода задаются пользователем
(2) всегда перебирает элементы арифметической прогрессии, причем шаг и условия выхода фиксированные
(3) предоставляет полный выбор перебираемой последовательности, итератора и условий выхода пользователю
(4) перебирает любую последовательность в порядке, заданным пользователем
(5) перебирает любую последовательность в порядке следования элементов

34) При реализации блока ветвления if в языке Python можно использовать следующие ключевые слова:
(1) else if, elif, else– единственный раз в блоке
(2) else if, elif– неограниченное количество раз и else один раз в блоке
(3) elif, else– неограниченное количество раз и else один раз в блоке
(4) elif– неограниченное количество раз и else один раз в блоке
(5) else if – неограниченное количество раз и else один раз в блоке

35) В языке Python циклом с предусловием является:
(1) do-while
(2) iterate
(3) while
(4) for
(5) в языке Python цикла с предусловием не существует

36) Оператор continue используется
(1) в качестве пустого оператора
(2) для перехода к следующей итерации цикла
(3) для выхода из цикла
(4) для выхода из блока if
(5) для возврата из функции и продолжения программы с точки ее вызова

37) При использовании оператора break в цикле с for, в котором определена ветвь else, цикл исполняется следующем образом:
(1) код в ветви else исполняется вне зависимости от способа выхода из цикла
(2) код в ветви else не исполняется вне зависимости от способа выхода из цикла
(3) код в ветви else исполняется только в том случае, если выход был произведен при помощи оператора break
(4) код в ветви else исполняется после того, как закончит работу код в цикле for
(5) при использовании ветви else применение оператора break недопустимо

38) В языке Python ветвь else используется в:
(1) циклах for
(2) циклах while
(3) циклах foreach
(4) ветвлении if
(5) блоке try

39) Оператор continue используется
(1) в качестве пустого оператора
(2) для перехода к следующей итерации цикла
(3) для выхода из цикла
(4) для возврата из функции и продолжения программы с точки ее вызова

40) Оператор break используется:
(1) в качестве пустого оператора
(2) для перехода к следующей итерации цикла
(3) для выхода из цикла
(4) для возврата из функции и продолжения программы с точки ее вызова
(5) прерывания выполнения условного оператора if

5.3. Оценочные материалы, используемые для экзамена (описание билетов, тестов и т.п.)

Экзамен не предусмотрен

5.4. Методика оценки освоения дисциплины

Формой промежуточной аттестации является зачёт в 3 семестре. Промежуточная аттестация проводится по результатам текущего контроля (выполнения обучающимися 1 домашнего задания, выполнения и защиты 5 лабораторных работ и тестирований по всем темам разделов дисциплины).

Оценка «зачтено» выставляется в случае, если обучающийся имеет оценку «зачтено» по всем видам текущего контроля.
Оценка «не зачтено» выставляется в случае, если обучающийся имеет оценку «не зачтено» по одному или более видам текущего контроля.

Критерии оценивания домашнего задания:
«Зачтено»
Домашнее задание выполнено без ошибок, либо с непринципиальными ошибками, не влияющими на физическую суть результата.
«Не зачтено»
Задание не выполнено, либо выполнено не полностью, либо в решении допущены существенные ошибки, не исправленные после замечаний преподавателя.

Критерии оценивания защиты лабораторных работ:
«Зачтено»
Лабораторная работа выполнена; отчёт по работе оформлен в соответствии предъявляемым требованиям; при ответе на вопросы по теме лабораторной работы студент демонстрирует знание основных теоретических положений работы и умение их применять на практике.
«Не зачтено»
Лабораторная работа не выполнена, либо отчёт по работе отсутствует или его оформление не соответствует предъявляемым требованиям, либо при ответе на вопросы по теме лабораторной работы студент демонстрирует незнание основных теоретических положений работы и неумение их применять на практике.

Критерии оценивания результатов тестирований:
«Зачтено»
Обучающийся ответил правильно более чем на 50 % вопросов теста
«Не зачтено»
Обучающийся ответил правильно на 50 % вопросов теста или менее

Порядок, определяющий процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих уровень сформированности компетенций, определен в Положении «О текущем контроле успеваемости и промежуточной аттестации обучающихся, текущем контроле посещения обучающимися аудиторных занятий в НИТУ «МИСиС» П 239.09-18, выпуск 2».

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Рекомендуемая литература

6.1.1 Основная литература

Обозначение	Авторы, составители	Заглавие	Библиотека	Издательство, год
Л 1.1	Шелудько В. М.	Основы программирования на языке высокого уровня Python: учебное пособие	ЭБС «Университетская библиотека ONLINE» URL: http://biblioclub.ru/index.php?page=book_red&id=500056	Ростов-на-Дону, Таганрог: Южный федеральный университет, 2017
Л 1.2	Хахаев И. А.	Практикум по алгоритмизации и программированию на Python : курс: учебное пособие	ЭБС «Университетская библиотека ONLINE» URL: http://biblioclub.ru/index.php?page=book_red&id=429256	Москва: Национальный Открытый Университет «ИНТУИТ», 2016

6.1.2 Дополнительная литература

Обозначение	Авторы, составители	Заглавие	Библиотека	Издательство, год
Л 2.1	Гэддис, Т.	Начинаем программировать на Python	НТБ СТИ НИТУ «МИСИС»	СПб. : БХВ-Петербург, 2019
Л 2.2	Лесса, А.	Python. Руководство разработчика	НТБ СТИ НИТУ «МИСИС»	СПб. : ООО "Диа Софт ЮП", 2001.

6.1.3 Методические разработки

Обозначение	Авторы, составители	Заглавие	Библиотека	Издательство, год
-	-	-	-	-

6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

Э. 1	Шелудько, В. М. Язык программирования высокого уровня Python: функции, структуры данных, дополнительные модули : учебное пособие : / В. М. Шелудько. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2017. – 108 с.: ил.– URL: https://biblioclub.ru/index.php?page=book&id=500060.

6.3. Перечень программного обеспечения	
П. 1	Microsoft Windows
П. 2	Microsoft office
П. 3	Kaspersky Endpoint Security
6.4. Перечень информационных справочных систем и профессиональных баз данных	
И. 1	- LMS Canvas (приказ НИТУ «МИСиС» № 387 о.в. от 05.06.2018 г. «О применении в учебном процессе ЭОР» https://lms.misis.ru/
И. 2	- Научная электронная библиотека eLIBRARY.RU: http://elibrary.ru/
И. 3	- Федеральный портал «Российское образование»: http://edu.ru
И. 4	- Открытое образование: http://openedu.ru
И. 5	- Российская государственная библиотека: http://www.rsl.ru
И. 6	- Электронно-библиотечная система (ЭБС) «Университетская библиотека онлайн»: http://biblioclub.ru
И. 7	- Электронная библиотека НИТУ «МИСИС»: http://elibrary.misis.ru
И. 8	- Университетская информационная система РОССИЯ: https://uisrussia.msu.ru/
И. 9	- Электронная библиотека РГБ: диссертации: http://diss.rsl.ru/?menu=disscatalog/
И. 10	- аналитическая база (индексы цитирования) Web of Science https://apps.webofknowledge.com
И. 11	- аналитическая база (индексы цитирования) Scopus https://www.scopus.com/
И. 12	- наукометрическая система InCites https://apps.webofknowledge.com

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ		
7.1	**Аудитория №408 «Лаборатория системного программирования»**	Перечень основного оборудования, учебно-наглядных пособий: • монитор - 9шт.; • персональный компьютер - 9шт.; • проектор; • экран настенный; • усилитель-распределитель; • комплект учебной мебели на 25 посадочных мест.
7.2	**Аудитория №306 «Помещение для самостоятельной работы обучающихся»**	Перечень основного оборудования, учебно-наглядных пособий: • проектор; • доска; • экран настенный; • компьютер – 6 шт.; • комплект учебной мебели на 20 человек. В помещении для самостоятельной работы обучающихся имеется подключение к сети «Интернет» и доступ в электронную информационно-образовательную среду организации.

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ИЗУЧЕНИЮ ДИСЦИПЛИНЫ

Обучение дисциплине проводится в один семестр и организуется в соответствии с настоящей программой.
Промежуточная аттестация по дисциплине предусмотрена в виде зачета.
Самостоятельная работа студентов осуществляется и контролируется с помощью:
– выполнения и защиты домашнего задания;
– выполнения и защиты лабораторных работ;
– контрольных тестирований, проводимых в течение семестра.
Зачет проставляется при условии выполнения учебного плана дисциплины по результатам выполненных и защищённого домашнего задания, лабораторных работ, тестирований по всем темам разделов дисциплины.
Оценочные средства позволяют оценить компетенции (части компетенций), сформированные у обучающихся в процессе освоения и по завершению изучения дисциплины.