

Министерство сельского хозяйства Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования «Вологодская государственная молочнохозяйственная академия
имени Н.В. Верещагина»

Инженерный факультет

Кафедра «Энергетические средства и технический сервис»

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
ПРОГРАММИРОВАНИЕ НА PYTHON

Направление подготовки: 35.03.06 Агроинженерия

Профили подготовки: Искусственный интеллект

Квалификация (степень) выпускника: Бакалавр

Вологда — Молочное

2025

Программа составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 35.03.06 «Агроинженерия»

Разработчик: канд. техн. наук, доцент Бирюков А.Л.

Программа одобрена на заседании кафедры энергетических средств и технического сервиса 20.02.25, протокол № 6.

Зав. кафедрой: канд. техн. наук, доцент Бирюков А.Л.

Рабочая программа дисциплины согласована на заседании методической комиссии инженерного факультета 20.02.25, протокол № 6.

Председатель методической комиссии: канд. техн. наук, доцент Берденников Е.А.

1 Цель и задачи учебной дисциплины

Цель — изучение конструкций языка программирования Python, формирование навыков программирования на языке Python для решения профессиональных задач.

Задачи дисциплины:

- изучение основных конструкций языка программирования Python;
- формирование умений самостоятельного создания и использования различных структур данных;
- формирование навыков автоматизация задач по сбору и обработке данных;
- развитие умений разработки эффективных алгоритмов и программ на основе языка программирования Python.

2 Место дисциплины в структуре ООП

Дисциплина «Программирование на языке Python» относится к блоку ФТД, Факультативные дисциплины федерального государственного образовательного стандарта высшего образования по направлению подготовки 35.03.06 Агроинженерия профиль «Искусственный интеллект» (ФТД.В.03).

Дисциплина «Программирование на языке Python» изучается в 1 семестре. Результаты изучения дисциплины востребованы в ходе изучения последующих дисциплин, учебной и производственной практики, а также при подготовке выпускной квалификационной работы. Форма промежуточной аттестации — зачет.

3 Компетенции обучающегося, формируемые в результате освоения дисциплины

Дисциплина направлена на формирование следующих компетенций выпускника:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
ПК-13. Способен участвовать в процессе создания систем искусственного интеллекта, на различных этапах жизненного цикла в качестве эксперта и ключевого пользователя	ИПК-13.1. Участвует в коллективной работе по созданию систем искусственного интеллекта в качестве эксперта ИПК-13.2. Проводит тестирование и опытную эксплуатацию систем искусственного интеллекта

4 Структура и содержание дисциплины

Общий объем дисциплины составляет 3 зачетные единицы.

4.1 Структура дисциплины

Вид учебной нагрузки	Всего часов (очная форма)
	1 семестр
Аудиторные занятия (всего)	54
<i>В том числе:</i>	

Вид учебной нагрузки	Всего часов (очная форма)
	2 семестр
Лекции	14
Лабораторные работы	40
Практические занятия	–
Самостоятельная работа (всего)	54
Вид промежуточной аттестации	зачет
Общая трудоёмкость, часы	108
Зачётные единицы	3

4.2 Содержание разделов дисциплины

Тема 1. Введение в программирование на языке Python

Интерпретатор, среда разработки. Комментарии, типы данных, операции с ними, переменные, операторы и их виды.

Тема 2. Управляющие структуры

Условный оператор, многовариантное ветвление, циклы while и for.

Тема 3. Строковые методы

Работа с текстом и строками. Обработка строк, поиск значения в строке, форматирование строк.

Тема 4. Функции, работа с файлами

Функция. Локальные и глобальные переменные. Передача параметров и возврат значений. Основы функционального программирования. Создание файла, чтение содержимого файла, изменение, удаление файла.

Тема 5. Базовые структуры данных

Управление списками. Коллекции данных: понятие коллекции, работа с кортежами, со словарями, стеком, очередью, двусторонней очередью.

Тема 6. Объектно-ориентированное программирование

Введение в объектно-ориентированное программирование. Классы. Инкапсуляция и конструкторы. Наследование и полиморфизм.

4.3 Разделы дисциплины и вид занятий

№ п/п	Раздел дисциплины	Лекции	ЛЗ	СРС	Всего
1.	Введение в программирование на языке Python	2	4	9	15
2.	Управляющие структуры	2	4	9	15
3.	Строковые методы	2	4	9	15
4.	Функции, работа с файлами	2	8	9	19
5.	Базовые структуры данных	2	8	9	19
6.	Объектно-ориентированное программирование	4	12	9	25
Всего		14	40	54	108

Примечание: ЛЗ – лабораторные занятия; СРС – самостоятельная работа студента

5 Матрица формирования компетенций по дисциплине

№ п/п	Разделы, темы дисциплины	Компетенции	Общее количество компетенций
1.	Введение в программирование на языке Python	ПК-13	1
2.	Управляющие структуры	ПК-13	1
3.	Строковые методы	ПК-13	1
4.	Функции, работа с файлами	ПК-13	1
5.	Базовые структуры данных	ПК-13	1
6.	Объектно-ориентированное программирование	ПК-13	1

6 Образовательные технологии

Объем аудиторных занятий всего – 108 часов, в том числе лекции – 14 часов, лабораторные работы – 40 часов.

Удельный вес занятий, проводимых в интерактивной форме, – 100% от объема аудиторных занятий.

Семестр	Вид занятия	Наименование темы	Используемые интерактивные образовательные технологии	Количество часов
2	Лекция	Введение в программирование на языке Python	Лекция-визуализация	2
2	Лекция	Управляющие структуры	Лекция-визуализация	2
2	Лекция	Строковые методы	Лекция-визуализация	2
2	Лекция	Функции, работа с файлами	Лекция-визуализация	2
2	Лекция	Базовые структуры данных	Лекция-визуализация	2
2	Лекция	Объектно-ориентированное программирование	Лекция-визуализация	4
2	ЛЗ	Работа в IDLE. Использование арифметических операций. создание списков и словарей. работа с циклами	Лабораторная работа	4
2	ЛЗ	Работа с основными встроенными функциями	Лабораторная работа	4
2	ЛЗ	Работа с итераторами, генераторами. Работа с генераторными выражениями	Лабораторная работа	4
2	ЛЗ	Работа с основными модулями	Лабораторная работа	4
2	ЛЗ	Работа с файлами, разработка синтаксического анализатора, вывод форматированных данных	Лабораторная работа	4
2	ЛЗ	Разработка приложения работы с базой данных	Лабораторная работа	4
2	ЛЗ	Разработка приложения с использованием ООП	Лабораторная работа	4
2	ЛЗ	Обработка изображений с применением библиотеки PIL	Лабораторная работа	4
2	ЛЗ	Разработка GUI приложения с помощью графических библиотек	Лабораторная работа	4

Семестр	Вид занятия	Наименование темы	Используемые интерактивные образовательные технологии	Количество часов
2	ЛЗ	Визуализация результатов работы математических алгоритмов с использованием NUMPY и MATPLOTLIB	Лабораторная работа	4
Итого:				54

7 Учебно-методическое обеспечение самостоятельной работы студентов. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

7.1 Перечень оценочных средств

Компетенции	Индикаторы достижения компетенций	Оценочные средства
ПК-13. Способен участвовать в процессе создания систем искусственного интеллекта, на различных этапах жизненного цикла в качестве эксперта и ключевого пользователя	ИПК-13.1. Участвует в коллективной работе по созданию систем искусственного интеллекта в качестве эксперта ИПК-13.2. Проводит тестирование и опытную эксплуатацию систем искусственного интеллекта	1. Задания для выполнения лабораторных работ. 2. Задания для самостоятельной работы. 3. Вопросы к зачету.

7.2 Образцы заданий для самостоятельной работы:

По итогам самостоятельной работы студент готовит отчет, включающий в себя ответы на вопросы и решение заданий, предполагавшихся к выполнению в ходе самостоятельной работы. Отчет сдается преподавателю в электронной форме.

Образцы заданий для самостоятельной работы

ЗАДАНИЕ 1

IDE: Anaconda + (Spyder | VS Code)

1. Написать программу на Python, вычисляющую значение выражения с использованием арифметических и логических операторов;
2. Предусмотреть возможность ввода пользователем значений a, b, c, d;
3. Наглядно показать приоритет операторов для некоторых формул;
4. Проверить работу программы.

Варианты заданий:

1. $(a - b) / (2 \text{ and } c) \cdot \log_c d$
2. $|a| \text{ and } 5 - (b + \log c + d) / 2$
3. $e^a \text{ or } b / c - d \cdot 5 \text{ or } \log_2 d$
4. $\text{not } a + b / (2 - c) \cdot \sqrt{d} \text{ and } 1$
5. $(a - b \text{ and } c) / (\sqrt[3]{|d|} \bmod 5)$
6. $a \text{ or } b / (c \cdot d) - 5 + e - \sin(d)$

7. $a - b \text{ div } c \cdot (3 \text{ or } d) + 2$
8. $(a^2 - 4) / (b + c) - \text{not } d$
9. $\text{not } a - b / (c + d \cdot 2) + \pi$
10. $2 \cdot a^{b^c} / (3 \cdot b - c)$ and НОД (d, c)

log – логарифм

e – экспонента

div – целочисленное деление

mod – деление по модулю

and – бинарный оператор «И»

or – бинарный оператор «ИЛИ»

not – бинарный комплиментарный оператор «НЕ»

НОД – наибольший общий делитель

|| - модуль числа

ЗАДАНИЕ 2

Тема: Условные операторы. Организация циклов.

1. Импортировать модуль **Turtle** (<https://docs.python.org/3/library/turtle.html>)
2. Написать программу на Python, рисующую геометрические фигуры с использованием условных операторов (*if, if - else, if - elif - else*) и операторов цикла (*for, while*);
3. Предусмотреть возможность ввода пользователем геометрических размеров;
4. Проверить работу программы.

ЗАДАНИЕ 3

IDE: Anaconda + (Spyder | VS Code) или Microsoft Visual Studio

Тема: Структуры данных. Ввод/вывод.

1. Использовать пару стандартных структур данных согласно своему варианту.
2. Продемонстрировать основные операции, которые можно применить к каждой из структур.
3. Определить, какая из структур быстрее в операциях: доступа, добавления, удаления, объединения, пересечения, разницы, сравнения, поиска, сортировки и т.п., при условии, что обе структуры поддерживают данные операции.
4. Обеспечить возможность сохранения данных в файл и загрузке из файла.
5. Проверить работу программы.

Варианты

1. list и dict
2. set и defaultdict
3. dict и deque
4. defaultdict и heapq
5. deque и counter
6. heapq и list
7. counter и set

ЗАДАНИЕ 4

IDE: Anaconda + (Spyder | VS Code) или Microsoft Visual Studio

Тема: Классы

Написать скрипт, содержащий в себе все возможности языка, связанные с объектно-ориентированным программированием:

Создание класса и -объекта;

Инкапсуляция и -статические методы;

Наследование и ?композиция;

Полиморфизм: -перегрузка операторов и -методов, -абстрактные и -виртуальные методы.

Проверить работу программы.

ЗАДАНИЕ 5

IDE: Anaconda + (Spyder | VS Code) или Microsoft Visual Studio

Тема: Обработка и генерация исключений. Пользовательские исключения.

1. За основу взять задание 4.
2. Продемонстрировать возможности языка по генерации и обработке исключений, блоки: try/except, finally и raise.
3. Применить assert для проверки предположения о значениях произвольных данных в произвольном месте программы.
4. Создать и задействовать свое собственное исключение.
5. Проверить работу программы.

ЗАДАНИЕ 6

IDE: Anaconda + (Spyder | VS Code)

1. Сделать обработку предлагаемого двоичного файла, в котором содержится заголовок и двоичные данные в формате, согласно приложению 1.
2. Обработанные данные вывести средствами Python на 3D график (surface chart).
3. Данные содержат отклонения от горизонтали поверхности в мм (примеры значений: 0...100 мм, обычно 7-25 мм).
4. В итоге должен получиться скрипт Python в параметрах которому можно передать имя файла, и он выведет график.

Данный код (файл Data.pas) использовать как подсказку для расшифровки файла **data.dat**

Код был написан на Delphi



data.dat



Data.pas

Образцы заданий для лабораторных работ:

По итогам выполнения лабораторной работы студент демонстрирует результаты работы программы преподавателю, предварительно разработав тестовые случаи.

Лабораторная работа №1

РАБОТА В IDLE. ИСПОЛЬЗОВАНИЕ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ. СОЗДАНИЕ СПИСКОВ И СЛОВАРЕЙ. РАБОТА С ЦИКЛАМИ

Цель работы: научиться использовать IDLE при работе с Python; изучить списки и словари, освоить основные методы для работы с ними; изучить виды циклов в Python.

Краткая теория и рекомендации по выполнению лабораторной работы изложены в <https://e.lanbook.com/book/180938> (с. 26-33).

Варианты заданий

1. Написать функцию, которая принимает целочисленный список с количеством элементов 1 или более и возвращает True, если цифра б является первым или последним элементом списка.

2. Написать функцию, которая принимает два целочисленных списка и возвращает True, если первые или последние элементы данных списков равны. Оба списка содержат 1 или более элементов.
3. Написать функцию, которая принимает целочисленный список, состоящий из трех элементов, и возвращает новый перевернутый список. Например, на входе {1,2,3}, а на выходе {3,2,1}.
4. Написать функцию, которая принимает два целочисленных списка, содержащих по три элемента каждый, и возвращает новый список, состоящий из двух элементов, являющихся средними во входящих списках. Например, для следующих входящих списков {1,2,3} и {3,4,5} итоговым будет {2,4}.
5. Написать функцию, которая принимает целочисленный список и возвращает True, если длина списка больше нуля и первый и последний элемент списка равны.

Лабораторная работа №2 РАБОТА С ОСНОВНЫМИ ВСТРОЕННЫМИ ФУНКЦИЯМИ

Цель работы: рассмотреть основные встроенные функции языка программирования Python и научиться с ними работать.

Краткая теория и рекомендации по выполнению лабораторной работы изложены в <https://e.lanbook.com/book/180938> (с. 36-38).

Варианты заданий

1. Написать функцию, которая принимает целое число и возвращает сгенерированный список, содержащий количество элементов от 0 до входящего числа включительно.
2. Написать функцию, которая принимает список, который содержит строки и выводит на экран новые списки из входящих строк.
3. Написать функцию, которая принимает список и выводит в консоль значения списка через точку с запятой. Требуется решить задачу одной строкой с использованием списка с оператором «*».
4. Написать функцию, которая принимает список, состоящий из p элементов, и возвращает их сумму.
5. Написать функцию, которая принимает список, состоящий из строк, которые являются целочисленными значениями или значениями с плавающей точкой, и возвращает их сумму.

Лабораторная работа №3 РАБОТА С ИТЕРАТОРАМИ, ГЕНЕРАТОРАМИ. РАБОТА С ГЕНЕРАТОРНЫМИ ВЫРАЖЕНИЯМИ

Цель работы: изучить понятия итератора и генератора в Python, а также их преимущества; ознакомиться с примерами их пользования.

Краткая теория и рекомендации по выполнению лабораторной работы изложены в <https://e.lanbook.com/book/180938> (с. 40-43).

Варианты заданий

1. Написать функцию, которая принимает целое число и с помощью генераторного выражения создает и возвращает новый список случайных чисел с длиной входящего числа.
2. Написать функцию, которая принимает три целых числа x, a, b и с помощью генераторного выражения создает и возвращает новый список длиной x случайных чисел

от a до b. Для решения данного задания рекомендуется использовать функцию `random.randint()`.

3. Написать функцию, которая принимает целочисленный список, состоящий из `n` элементов, и с помощью генераторного выражения создает и возвращает список, элементами которого являются удвоенные элементы входящего списка.

4. Написать функцию, которая принимает целочисленный список, состоящий из `n` элементов, и с помощью генераторного выражения создает и возвращает список, содержащий только четные элементы входящего списка.

5. Написать функцию, которая принимает целочисленный список, состоящий из `n` элементов, и с помощью генераторного выражения создает и возвращает список, содержащий только положительные элементы входящего списка.

Лабораторная работа №4 РАБОТА С ОСНОВНЫМИ МОДУЛЯМИ

Цель работы: изучить основные модули стандартной библиотеки Python 3

Краткая теория и рекомендации по выполнению лабораторной работы изложены в <https://e.lanbook.com/book/180938> (с. 46-50).

Варианты заданий

1. Написать функцию, которая принимает объект `datetime` и возвращает `True`, если год полученного значения високосный.

2. Написать функцию, которая принимает строку и возвращает объект `datetime` из этой строки. Формат даты и времени можно использовать любой.

3. Написать функцию, которая принимает целочисленное значение `x` и возвращает текущую дату, прибавив к году `x`.

4. Написать функцию, которая принимает объект `datetime` и возвращает номер недели.

5. Написать функцию, которая принимает объект `datetime` и возвращает номер дня в году.

Лабораторная работа №5 РАБОТА С ФАЙЛАМИ. РАЗРАБОТКА СИНТАКСИЧЕСКОГО АНАЛИЗАТОРА. ВЫВОД ФОРМАТИРОВАННЫХ ДАННЫХ

Цель работы: изучить работу с файлами с помощью функций из стандартной библиотеки; рассмотреть понятие синтаксического анализа текста и ознакомиться с его применением на языке Python; изучить возможности взаимодействия Python с форматом хранения данных JSON.

Краткая теория и рекомендации по выполнению лабораторной работы изложены в <https://e.lanbook.com/book/180938> (с. 52-57).

Варианты заданий

1. Написать функцию, которая принимает `Json` строку и конвертирует её в словарь.

2. Написать функцию, которая принимает путь к файлу и количество строк, которые требуется прочитать и возвращает считанные строки в файле.

3. Написать программу, которая принимает путь к файлу и возвращает наиболее длинное слово из него.

4. Написать функцию, которая принимает путь к файлу, текст и номер строки и записывает в файл полученный текст в указанный номер строки.

5. Написать программу, которая принимает путь к файлу и возвращает True, или False в зависимости от того, доступен ли файл для чтения и записи или нет.

Лабораторная работа №6 РАЗРАБОТКА ПРИЛОЖЕНИЯ РАБОТЫ С БАЗОЙ ДАННЫХ

Цель работы: изучить возможности взаимодействия Python с реляционными базами данных.

Краткая теория и рекомендации по выполнению лабораторной работы изложены в <https://e.lanbook.com/book/180938> (с. 59-62).

Варианты заданий

1. Написать функцию, которая принимает наименование таблицы, поля и его значение и возвращает идентификатор записи, в которой значение полученного поля соответствует переданному функции, или возвращает *None*.
2. Написать функцию, которая принимает наименование таблицы и выводит количество одинаковых значений в её полях и наиболее часто повторяющееся значение полей.
3. Написать функцию, которая возвращает наименование всех таблиц связки, которые находятся в базе данных, если они существуют. Стоит отметить, что в данном задании предполагается, что таблицы связки именуются склеиванием имени двух таблиц через знак нижнего подчеркивания «_».
4. Написать функцию, которая принимает наименование таблицы и список списков, которые содержат данные, и обновляет данные полученной таблицы на указанные во внутренних списках, где в качестве идентификаторов записей следует принимать номер внутреннего списка плюс единица, т.к. идентификаторы записей СУБД нумеруются с единицы.
5. Написать функцию, которая принимает словарь, в котором ключами являются наименования таблиц, а значениями список списков, содержащих данные для таблицы, и выполняет вставку полученных данных в указанные таблицы.

Лабораторная работа №7 РАЗРАБОТКА ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ ООП

Цель работы: ознакомиться с методологией объектно-ориентированного программирования, изучить реализацию данной методологии в языке Python 3.

Краткая теория и рекомендации по выполнению лабораторной работы изложены в <https://e.lanbook.com/book/180938> (с. 64-69).

Варианты заданий

1. Написать родительский и дочерний классы с методами «display». В каждом методе в консоль выводится различная строка.
2. Написать класс Exception с возможностью передачи в него сообщения и реализовать класс, в котором будет пробрасываться данный Exception.
3. Написать класс, который реализует паттерн Singleton.
4. Написать класс, который содержит два метода «get_string» и «printjipper string», где первый метод принимает строку, а второй выводит данную строку в верхнем регистре.
5. Написать класс UserLanguagePreference, который в конструкторе принимает список языков в виде строки, которые использует пользователи и содержит метод add lang(lang

str), который добавляет язык в список, если его там не существует. Данный класс использует инкапсуляцию, для получения используемого списка.

Лабораторная работа №8 ОБРАБОТКА ИЗОБРАЖЕНИЙ С ПРИМЕНЕНИЕМ БИБЛИОТЕКИ PIL

Цель работы: рассмотреть возможности работы библиотеки PIL, изучить её базовые инструменты для обработки изображений.

Краткая теория и рекомендации по выполнению лабораторной работы изложены в <https://e.lanbook.com/book/180938> (с. 71-73).

Варианты заданий

1. Написать функцию, которая принимает путь к директории и создает новую директорию по тому же пути с именем thumbnail, в которую записывает иконки изображений, находящихся в принятой функцией директории.
2. Написать функцию, которая принимает путь к изображению и конвертирует его в формат ICO, сохраняя её по тому же пути, что и исходное изображение.
3. Написать функцию, которая принимает путь к изображению и выполняет над ней autoccontrast, сохраняя новое изображение по тому же пути.
4. Написать функцию, которая принимает путь к изображению и создает отраженное изображение, сохраняя его по тому же пути.
5. Написать функцию, которая принимает текст и стиль шрифта в виде строки и создает изображение с полученным текстом и шрифтом.

Лабораторная работа №9 РАЗРАБОТКА GUI ПРИЛОЖЕНИЯ С ПОМОЩЬЮ ГРАФИЧЕСКИХ БИБЛИОТЕК

Цель работы: рассмотреть возможности библиотеки Tkinter, её базовые виджеты и изучить основные принципы создания приложений с графическим интерфейсом с помощью данной библиотеки.

Краткая теория и рекомендации по выполнению лабораторной работы изложены в <https://e.lanbook.com/book/180938> (с. 76-79).

Варианты заданий

1. Написать GUI приложение, которое имеет текстовое поле и кнопку для сохранения текста, при нажатии на которую открывается модальное окно указания пути для сохранения текстового файла.
2. Написать GUI приложение, которое позволяет просматривать PDF файлы.
3. Написать GUI приложение, которое имеет кнопку «Сделать скриншот», по нажатию на которую создается скриншот экрана и вставляет в окно приложение и появляются возможности настройки яркости изображения, некоторые фильтры и поворот изображения. По нажатию на кнопку «Сохранять» должно сохраняться сконфигурированное изображение.
4. Написать GUI приложение, которое представляет собой упрощенный файловый менеджер, с возможностью создания, удаления и переименования директорий и файлов.
5. Написать GUI приложение, которое представляет собой галерею изображений с возможностью просмотра плитки изображений. каждого изображения по отдельности и добавления новых изображений в галерею.

Лабораторная работа №10
ВИЗУАЛИЗАЦИЯ РЕЗУЛЬТАТОВ РАБОТЫ МАТЕМАТИЧЕСКИХ АЛГОРИТМОВ С
ИСПОЛЬЗОВАНИЕМ NUMPY И MATPLOTLIB

Цель работы: рассмотреть возможности пакета NumPy, его основные составляющие и возможности для обработки многомерных массивов данных: изучить библиотеку Matplotlib и её основные инструменты для создания графиков и диаграмм.

Краткая теория и рекомендации по выполнению лабораторной работы изложены в <https://e.lanbook.com/book/180938> (с. 82-88).

Варианты заданий

1. Написать функцию, которая принимает список целых чисел и возвращает True, если в списке отсутствуют нули, в противном случае возвращает False.
2. Написать функцию, которая создает массивы из десяти нулей, единиц и пятерок.
3. Написать функцию, которая создает массив из всех четных элементов между 50 и 90.
4. Написать функцию, которая принимает два вектора и возвращает их внешнее произведение.
5. Написать функцию, которая принимает вектор или матрицу и возвращает их норму.

7.3 Вопросы для зачета

1. Интерпретатор.
2. Среда разработки.
3. Комментарии.
4. Типы данных.
5. Операции с данными.
6. Переменные.
7. Операторы и их виды.
8. Условный оператор.
9. Многовариантное ветвление.
10. Циклы while и for.
11. Работа с текстом и строками.
12. Обработка строк.
13. Поиск значения в строке.
14. Форматирование строк.
15. Функция.
16. Локальные и глобальные переменные.
17. Передача параметров и возврат значений.
18. Основы функционального программирования.
19. Создание файла.
20. Чтение содержимого файла.
21. Изменение и удаление файла.
22. Управление списками.
23. Понятие коллекции.
24. Работа с кортежами.
25. Работа со словарями, стеком, очередью, двусторонней очередью.
26. Работа стеком.
27. Работа очередью.
28. Работа двусторонней очередью.

29. Принципы объектно-ориентированного программирования.
30. Классы.
31. Инкапсуляция и конструкторы.
32. Наследование.
33. Полиморфизм.
34. Пакет для быстрой обработки матриц и векторов *NumPy*.
35. Пакет для построения и отображения графиков *Matplotlib*.

7.3 Примеры тестовых заданий для промежуточной аттестации

1. Каков будет результат выполнения `int("88")`?
 1. "88"
 2. 88
 3. 88.00
2. Какие имена переменных являются правильными?
 1. N
 2. ABC
 3. sum
 4. 41And
 5. A+B
 6. _mam
3. Каким будет результат выполнения кода?


```
first = [1, 2, 3]
second = first[:]
second[2] = 4
print(first[2])
```

 - a) 4
 - b) 3
 - c) 2
 - d) 14
4. Каким будет результат выполнения программы?


```
a = {'model': '1', 'color': 'red'}
type(a)
```

 1. tuple
 2. set
 3. dict
5. Что выведет программа?


```
list = []
for i in range(100):
    list.append(lambda x: x + i)
list[42](3)
```

 1. 45
 2. 42
 3. 102
6. Что будет в результате выполнения программного кода?


```
class Account():
    def __init__(self):
        self.money = 0
    def deposit(self, amount):
        self.money += amount

account = Account()
money = 100
account.deposit(50)
print(money, account.money)
```

 - a) 100 100
 - b) 100 150
 - c) 50 100
 - d) 100 50
 - e) 150 150
 - f) 50 50

Уровни оценки компетенций:

- базовый 55-69 баллов,
- повышенный 70-100 баллов.

Преподаватель проводит систематический контроль знаний студентов, ориентируясь на перечень вопросов для проведения зачета.

7.4 Критерии оценивания ответа студента на зачете

Ответ на зачете оценивается исходя из 40 баллов (максимум). Билет содержит теоретический вопрос и практическое задание, преподаватель может задавать дополнительные вопросы. Полный ответ на основной вопрос оценивается максимум в 20 баллов, предполагает свободное изложение (не чтение) всего необходимого материала, ответы студента на уточняющие вопросы, если они есть. Правильный ответ на дополнительный вопрос оценивается максимум в 5 баллов. Правильное выполнение практического задания оценивается в 20 баллов.

7.5 Критерии оценки лабораторных работ и самостоятельной работы студента (от 0 до 10 баллов):

- **9-10 баллов** выставляется студенту, если работа выполнена самостоятельно и полностью верно; представлен отчет, содержащий результаты выполнения заданий работы и ответы на контрольные вопросы; студент анализирует результаты, полученные в ходе выполнения работы, делает выводы.
- **7-8 баллов** выставляется студенту, если работа выполнена самостоятельно, в целом правильно, но имеются некоторые неточности в выполнении заданий или ответах на контрольные вопросы; представлен отчет, содержащий результаты выполнения заданий и ответы на контрольные вопросы; студент анализирует результаты, полученные в ходе выполнения работы, делает выводы.
- **5-6 баллов** выставляется студенту, если работа выполнена самостоятельно, в целом правильно, но имеются некоторые неточности в выполнении заданий или ответах на контрольные вопросы; представлен отчет, содержащий результаты выполнения заданий лабораторной работы и ответы на контрольные вопросы; студент испытывает затруднения при проведении анализа результатов, полученных в ходе выполнения лабораторной работы, и формулировке выводов.
- **3-4 балла** выставляется студенту, если студент не до конца справился с заданием, не совсем верно ответил на контрольные вопросы, однако оформил отчет по результатам работы.
- **1-2 балла** выставляется студенту, если студент не до конца справился с заданием, не совсем верно ответил на контрольные вопросы, не оформил отчет по результатам работы.
- **0 баллов** выставляется студенту, если студент не справился с заданием, неверно ответил на представленные вопросы.

7.6 Шкала и критерии оценивания компетенций на различных этапах их формирования

Шкала оценивания компетенций:

Оценка в 100-балльной шкале	Оценка в 5-ти балльной шкале	Уровень сформированности компетенций
0-54 баллов	неудовлетворительно (не зачтено)	Недостаточный
55-69 баллов	удовлетворительно (зачтено)	Пороговый уровень
70-85 баллов	хорошо (зачтено)	Продвинутый уровень
86-100 баллов	отлично (зачтено)	Высокий уровень

Критерии оценивания компетенций:

Индикаторы достижения компетенций	Критерии оценивания компетенций		
	Пороговый уровень	Продвинутый уровень	Высокий уровень
ПК-13. Способен участвовать в процессе создания систем искусственного интеллекта, на различных этапах жизненного цикла в качестве эксперта и ключевого пользователя	<p>Частично знает методы и средства взаимодействия с инженерами в процессе создания систем искусственного интеллекта в части применения языка программирования Python; основные критерии качества систем искусственного интеллекта.</p> <p>В основном умеет взаимодействовать с инженерами в процессе создания систем искусственного интеллекта и, принимает участие в координации работ по созданию, внедрению и сопровождению систем искусственного интеллекта и в тестировании работоспособности и качества функционирования систем искусственного интеллекта, и в проверке выполнения требований к системе на основе применения языка программирования Python</p>	<p>Знает основные методы и средства взаимодействия с инженерами в процессе создания систем искусственного интеллекта в части применения языка программирования Python; основные критерии качества систем искусственного интеллекта.</p> <p>Умеет взаимодействовать с инженерами в процессе создания систем искусственного интеллекта и, принимает участие в координации работ по созданию, внедрению и сопровождению систем искусственного интеллекта и в тестировании работоспособности и качества функционирования систем искусственного интеллекта, и в проверке выполнения требований к системе на основе применения языка программирования Python</p>	<p>Демонстрирует глубокое знание и понимание: методов и средств взаимодействия с инженерами в процессе создания систем искусственного интеллекта в части применения языка программирования Python; основных критериев качества систем искусственного интеллекта.</p> <p>Умеет полностью верно и самостоятельно взаимодействовать с инженерами в процессе создания систем искусственного интеллекта и, принимает участие в координации работ по созданию, внедрению и сопровождению систем искусственного интеллекта и в тестировании работоспособности и качества функционирования систем искусственного интеллекта, и в проверке выполнения требований к системе на основе применения языка программирования Python</p>

8 Учебно-методическое и информационное обеспечение дисциплины

8.1 Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

Основная литература:

1. Никитина, Т. П. Программирование. Основы Python / Т. П. Никитина, Л. В. Королев. – Санкт-Петербург: Лань, 2023. – 156 с. – ISBN 978-5-507-45283-5. – Текст: электронный // Лань: электронно-библиотечная система. – URL: <https://e.lanbook.com/book/302714> (дата обращения: 05.06.2023).
2. Титов, А. Н. Python. Обработка данных: учебно-методическое пособие / А.Н. Титов, Р.Ф. Тагиева. – Казань: Изд-во КНИТУ, 2022. - 104 с. - ISBN 978-5-7882-3171-6. – Текст: электронный. – URL: <https://znanium.com/catalog/product/2069264> (дата обращения: 05.06.2023).
3. Жуков, Р. А. Язык программирования Python: практикум: учебное пособие / Р.А. Жуков. – Москва: ИНФРА-М, 2023. – 216 с. + Доп. материалы [Электронный ресурс]. – (Высшее образование: Бакалавриат). – DOI 10.12737/textbook_5cb5ca35aaa7f5.89424805. - ISBN 978-5-16-016971-2. – Текст: электронный. – URL: <https://znanium.com/catalog/product/1915716> (дата обращения: 05.06.2023).
4. Широбокова, С. Н. Программирование на языке Python для лабораторных занятий : учебное пособие / С. Н. Широбокова, А. А. Кацупеев, А. В. Сулыз. – Новочеркасск : ЮРГПУ, 2020. – 104 с. – ISBN 978-5-9997-0725-3. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/180938> (дата обращения: 05.06.2023).

Дополнительная литература:

1. Рагимханова, Г. С. Программирование на Python : учебное пособие / Г. С. Рагимханова. — Махачкала : ДГПУ, 2022. — 126 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/330071> (дата обращения: 05.06.2023).
2. Программирование на Python в примерах и задачах / А. Васильев. – М.: Эксмо, 2021. – 616 с.
3. Косицин, Д. Ю. Язык программирования Python : учебно-методическое пособие / Д. Ю. Косицин. – Минск : БГУ, 2019. – 136 с. – ISBN 978-985-566-746-0. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/180546> (дата обращения: 05.06.2023).

8.2 Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине (модулю)

Учебно-методические материалы для выполнения лабораторных работ.

6.3 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля), включая перечень информационных справочных систем (при необходимости)

1. Информационно-справочная система «КонсультантПлюс. – URL: <http://www.consultant.ru/>
2. Программирование на Python. URL: <https://stepik.org/course/67/syllabus> (дата доступа 04.06.2023).
3. «Поколение Python»: курс для продвинутых URL: <https://stepik.org/course/68343/promo#toc> (дата доступа 04.06.2023).
4. Добрый, добрый Python – обучающий курс от Сергея Балакирева. URL: <https://stepik.org/course/100707/promo> (дата доступа 04.06.2023)
5. Научная электронная библиотека eLIBRARY.RU – URL: <http://elibrary.ru>
6. ЭБС ЛАНЬ – URL: <https://e.lanbook.com/>,

7. ЭБС Znanium.com – URL: <https://znanium.com/>

6.4 Перечень информационных технологий, используемых при изучении дисциплины, включая программное обеспечение

1. ОС семейства Microsoft Windows.
2. MS Office 365.
3. Браузер.
4. Язык программирования Python – URL: <https://www.python.org/>
5. Среда программирования на языке Python, например, *JupyterLab* URL: <https://jupyter.org/>.
6. NumPy – пакет для научных вычислений с Python. – URL: <https://numpy.org/>
7. Программное обеспечение с открытым исходным кодом для математики, науки и техники – URL: <https://scipy.org/>
8. Библиотека на языке программирования Python для визуализации данных двумерной и трёхмерной графикой Matplotlib – URL: <https://matplotlib.org/>
9. Инструмент для анализа и обработки данных с открытым исходным кодом Pandas – URL: <https://pandas.pydata.org/>
10. Anaconda – платформа для быстрой разработки и развертывания безопасных решений Python – URL: <https://www.anaconda.com/>
11. Google Colab – URL: https://colab.research.google.com/#scrollTo=5fCEDCU_qrC0
12. API глубокого обучения Keras – URL: <https://keras.io/>
13. Комплексная платформа машинного обучения TensorFlow – URL: <https://www.tensorflow.org/>
14. Машинное обучение с открытым исходным кодом и визуализация данных Orange Data Mining – URL: <https://orangedatamining.com/>
15. Data Analytics Platform KNIME – URL: <https://www.knime.com/knime-analytics-platform>
16. Некоммерческий проект с открытым исходным кодом Project Jupyter – URL: <https://jupyter.org/>

9 Материально-техническое обеспечение дисциплины

Учебная аудитория 4205 для проведения занятий лекционного и семинарского типа (практические занятия); групповых и индивидуальных консультаций; текущего контроля и промежуточной аттестации.

Оснащенность:

Учебная мебель: столы – 35, стулья – 75, доска меловая, кафедра.

Основное оборудование: экран для проектора 1 шт., проектор – 1 шт., компьютер в комплекте - 1 шт.

Учебная аудитория 4202 Компьютерный класс, для проведения лабораторных занятий, самостоятельной работы.

Оснащенность:

Учебная мебель: стол преподавателя, компьютерные столы – 15, компьютерные кресла – 16.

Основное оборудование: 15 компьютеров с доступом в электронно-образовательную среду Академии, ЭБС и сети Интернет.

Учебная аудитория 4203 Компьютерный класс.

Оснащенность:

Учебная мебель: стол преподавателя, компьютерные столы – 15, компьютерные кресла – 16;

Основное оборудование: 15 компьютеров с доступом в электронно-образовательную среду Академии, ЭБС и сети Интернет.

Особенности реализации дисциплины (модуля) для инвалидов и лиц с ограниченными возможностями здоровья

Для обеспечения образования инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) реализация дисциплины может осуществляться в адаптированном виде, исходя из индивидуальных психофизических особенностей и по личному заявлению обучающегося, в части создания специальных условий.

В специальные условия могут входить: предоставление отдельной аудитории, необходимых технических средств, присутствие ассистента, оказывающего необходимую техническую помощь, выбор формы предоставления инструкции по порядку проведения текущего контроля и промежуточной аттестации, использование специальных технических средств, предоставление перерыва для приема пищи, лекарств и др.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

10 Карта компетенций дисциплины

Программирование на Python (направление подготовки – 35.03.06 «Агроинженерия»)					
Цель дисциплины	Цель – изучение конструкций языка программирования Python, формирование навыков программирования на языке Python для решения профессиональных задач.				
Задачи дисциплины	<ul style="list-style-type: none"> • изучение основных конструкций языка программирования Python; • формирование умений самостоятельного создания и использования различных структур данных; • формирование навыков автоматизация задач по сбору и обработке данных; • развитие умений разработки эффективных алгоритмов и программ на основе языка программирования Python. 				
В процессе освоения данной дисциплины студент формирует и демонстрирует следующие					
Компетенции		Планируемые результаты обучения (индикаторы достижения компетенции)	Технологии формирования	Форма оценочного средства	Ступени уровней освоения компетенции
Индекс	Формулировка				
Универсальные компетенции					
УК-13	Способен участвовать в процессе создания систем искусственного интеллекта, на различных этапах жизненного цикла в качестве эксперта и ключевого пользователя	ИПК-13.1. Участвует в коллективной работе по созданию систем искусственного интеллекта в качестве эксперта ИПК-13.2. Проводит тестирование и опытную эксплуатацию систем искусственного интеллекта	Лекции Лабораторные занятия Самостоятельная работа	Устный опрос Индивидуальное задание	<p>Пороговый уровень</p> <p>Частично знает методы и средства взаимодействия с инженерами в процессе создания систем искусственного интеллекта в части применения языка программирования Python; основные критерии качества систем искусственного интеллекта.</p> <p>В основном умеет взаимодействовать с инженерами в процессе создания систем искусственного интеллекта и, принимает участие в координации работ по созданию, внедрению и сопровождению систем искусственного интеллекта и в тестировании работоспособности и качества функционирования систем искусственного интеллекта, и в проверке выполнения требований к системе на основе применения языка программирования Python</p> <p>Продвинутый уровень</p> <p>Знает основные методы и средства взаимодействия с инженерами в процессе создания систем искусственного интеллекта в части применения языка программирования Python; основные критерии качества</p>

				<p>систем искусственного интеллекта. Умеет взаимодействовать с инженерами в процессе создания систем искусственного интеллекта и, принимает участие в координации работ по созданию, внедрению и сопровождению систем искусственного интеллекта и в тестировании работоспособности и качества функционирования систем искусственного интеллекта, и в проверке выполнения требований к системе на основе применения языка программирования Python</p> <p style="text-align: center;">Высокий уровень</p> <p>Демонстрирует глубокое знание и понимание: методов и средств взаимодействия с инженерами в процессе создания систем искусственного интеллекта в части применения языка программирования Python; основных критериев качества систем искусственного интеллекта. Умеет полностью верно и самостоятельно взаимодействовать с инженерами в процессе создания систем искусственного интеллекта и, принимает участие в координации работ по созданию, внедрению и сопровождению систем искусственного интеллекта и в тестировании работоспособности и качества функционирования систем искусственного интеллекта, и в проверке выполнения требований к системе на основе применения языка программирования Python</p>
--	--	--	--	---