

**УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ  
УЧЕБНОЙ ДИСЦИПЛИНЫ  
ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON**

Направление подготовки: 35.03.06 Агроинженерия

Направленность (профиль): Искусственный интеллект, цифровая среда образовательной организации

Квалификация (степень) выпускника: Бакалавр

## 1. Компетенции обучающегося, формируемые в результате освоения дисциплины

Дисциплина направлена на формирование следующих компетенций выпускника:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции
УК-11. Способен планировать и организовывать свою деятельность в цифровом пространстве с учетом правовых и этических норм взаимодействия человека и искусственного интеллекта и требований информационной безопасности	ИУК-11.1. Выбирает современные технологии и системы искусственного интеллекта для решения задач в профессиональной деятельности ИУК-11.3. Применяет и адаптирует правовые и этические нормы и национальные и международные стандарты в области искусственного интеллекта и смежных областях для решения задач в профессиональной деятельности в условиях изменений социально-экономических условий
ОПК-8. Способен применять естественные, общественные, когнитивные науки и инженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования для решения профессиональных задач с использованием систем искусственного интеллекта	ИОПК-8.1. Применяет естественные, общественные, когнитивные науки и инженерные знания, методы математического анализа и моделирования для решения профессиональных задач с использованием систем искусственного интеллекта
ОПК-9. Способен решать задачи в профессиональной деятельности на основе информационной и библиографической культуры, цифровых технологий и систем искусственного интеллекта	ИОПК-9.1. Выбирает, применяет и адаптирует методы исследования для решения задач профессиональной деятельности с использованием систем искусственного интеллекта
ПК-12. Способен классифицировать и идентифицировать задачи искусственного интеллекта, выбирать адекватные методы и инструментальные средства решения задач искусственного	ИПК-12.1. Классифицирует и идентифицирует задачи систем искусственного интеллекта в зависимости от особенностей проблемной и предметной областей ИПК-12.2. Выбирает методы и инструментальные средства искусственного интеллекта для решения задач в зависимости от особенностей проблемной области ИПК-12.3. Собирает исходную информацию и формирует требования к решению задач с использованием методов искусственного

<b>Код и наименование компетенции</b>	<b>Код и наименование индикатора достижения компетенции</b>
интеллекта	интеллекта
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

## 2. Разделы дисциплины и вид занятий

№ п/п	Раздел дисциплины	Лекции	ЛЗ	СРС	Всего
1.	Развитие искусственного интеллекта	3	4	16	23
2.	Правовые и этические нормы в области искусственного интеллекта	4	4	14	22
3.	Задачи агротехнической отрасли, решаемые на основе методов искусственного интеллекта	4	4	20	28
4.	Основы программирования систем искусственного интеллекта	6	5	20	31
<b>Всего</b>		<b>17</b>	<b>17</b>	<b>70</b>	<b>104</b>

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

## 3. Образовательные технологии

Объем аудиторных занятий всего – 34 часа, в том числе лекции – 17 часов, лабораторные работы – 17 часов.

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

Семестр	Вид занятия	Наименование темы	Используемые интерактивные образовательные технологии	Количество часов
3	Лекция	Развитие искусственного интеллекта	Лекция-визуализация	3
3	Лекция	Правовые и этические нормы в области искусственного интеллекта	Лекция-визуализация	4
3	Лекция	Задачи агротехнической отрасли, решаемые на основе методов искусственного интеллекта	Лекция-визуализация	4
3	Лекция	Основы программирования систем искусственного интеллекта	Лекция-визуализация	6
3	ЛЗ	Классификация систем искусственного интеллекта. Методы и инструментальные средства решения задач искусственного интеллекта	Лабораторная работа	2
3	ЛЗ	Правовые нормы в области искусственного интеллекта. Этика и искусственный интеллект	Лабораторная работа	2
3	ЛЗ	Применение методов искусственного интеллекта для решения задач агротехнической отрасли	Лабораторная работа	4

Семестр	Вид занятия	Наименование темы	Используемые интерактивные образовательные технологии	Количество часов
3	ЛЗ	Базовые конструкции языка Python	Лабораторная работа	5
3	ЛЗ	Пакет для быстрой обработки матриц и векторов NumPy	Лабораторная работа	2
3	ЛЗ	Пакет для построения и отображения графиков Matplotlib	Лабораторная работа	2
<b>Итого:</b>				<b>34</b>

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

##### 4.1. 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.

##### 4.2 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.
6. Написать функцию, которая принимает целочисленный список, содержащий три элемента, и возвращает сумму этих элементов.
7. Написать функцию, которая принимает целочисленный список, содержащий три элемента, и изменяет данный список путем изменения всех его элементов на максимальный крайний элемент списка. Например, для входящего списка {1,2,3} изменённым будет {3,3,3}.
8. Написать функцию, которая принимает целочисленный список, состоящий из n элементов, и возвращает True, если он содержит числа 2 или 3.
9. Написать функцию, которая принимает целочисленный список, состоящий из n элементов, и возвращает количество четных чисел в списке.
10. Написать функцию, которая принимает целочисленный список, состоящий из n элементов, и возвращает сумму элементов списка. Однако стоит исключить из подсчета число 13 и числа, которые следуют после него. Например, для входящего списка {1,2,3,13,4} сумма будет равна 6.
11. Написать функцию, которая принимает целочисленный список, состоящий из n элементов, и возвращает True, если в каком-то месте списка по порядку содержатся элементы 1,2,3 или комбинации их перестановок.
12. Написать функцию, которая принимает два словаря и возвращает новый словарь, в котором значения одинаковых ключей из входящих словарей являются суммой, а разные ключи просто добавляются в новый словарь. Например, для входящих словарей {'a':200, 'b':50} и {'a':100, 'c':500} выходным словарем будет {'a': 300, 'b':50, 'c':500}.
13. Написать функцию, которая принимает два списка и возвращает словарь, в котором ключами выступают элементы первого списка, а значениями ключей - элементы второго. Требуется предусмотреть ситуации, когда один из списков или оба будут пустыми. В ситуации, когда один список по длине больше другого, последние элементы большего по длине списка, не учитывать.
14. Написать функцию, которая принимает список, состоящий из n словарей, где в каждом словаре присутствуют ключи status (True или False) и name (строка). Вернуть список, который будет содержать все значения ключей name в тех словарях, где значение status является True. В случае отсутствия таких вернуть пустой список.
15. Написать функцию, которая принимает на вход список, состоящий из n различных элементов. Вернуть словарь, в котором ключами являются элементы входящего списка, а их значениями - число повторений этих элементов во входящем списке.

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  $n$  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.
6. Написать функцию, которая принимает список, состоящий из  $n$  элементов, и возвращает количество целых чисел и чисел с плавающей точкой.
7. Написать функцию, которая принимает целочисленный список, состоящий из  $n$  элементов, и записывает в файл новый список, состоящий из тех элементов входящего списка, значения квадрата которых не превышают 30.
7. Написать функцию, которая принимает целочисленный список, состоящий из  $n$  элементов, и записывает в файл новый список, состоящий из тех элементов входящего списка, значения квадрата которых не превышают 30.
8. Написать функцию, которая принимает целочисленный список, состоящий из  $n$  элементов, и возвращает список, состоящий из новых списков, элементами которых являются числа от 0 до значений элемента во входящем списке включительно. Например, входящий список имеет вид [1,2,3] на выходе будет список списков [[0 1], [0,1,2], [0,1,2,3]].
9. Написать функцию, которая принимает целочисленный список, содержащий  $n$  элементов, и целое число, и выводит на экран сгенерированные списки, содержащие  $n$  элементов с шагом итерации, равном элементу входящего списка.
10. Написать функцию, которая принимает на вход две строки и возвращает количество совпадений двух символов в строках. Например, на входе две строки «xadasw» и «xad» совпадением будет считаться группа символов «xa» и «ad».

#### 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  $c$  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.
6. Написать функцию, которая принимает словарь и с помощью генераторного выражения создает и возвращает новый список, содержащий значения ключей входящего словаря.
7. Написать функцию, которая принимает два одинаковых по длине списка и с помощью генераторного выражения создает и возвращает новый словарь, в котором ключами являются элементы первого списка, а значениями ключей - элементы второго.
8. Написать функцию, которая принимает список и с помощью генераторного выражения создает и возвращает словарь, где в качестве ключей будут номера позиций элементов входящего словаря, а значениями - сами элементы.
9. Написать функцию, которая принимает целочисленный список, состоящий из  $n$  элементов, и с помощью генераторного выражения создает новый массив элементов из тех элементов входящего массива, квадрат числа которых не превышает 30.
10. Написать функцию, которая принимает список, состоящий из  $n$  элементов, и целое число  $x$  и с помощью генераторной функции выводит на консоль только те элементы входящего списка, которые больше числа  $x$ .
10. Написать функцию, которая принимает список, состоящий из  $n$  элементов, и целое число  $x$  и с помощью генераторной функции выводит на консоль только те элементы входящего списка, которые больше числа  $x$ .
11. Написать функцию, которая принимает список, состоящий из  $n$  элементов, и два целых числа  $a$  и  $b$  с помощью генераторной функции выводит на консоль элементы входящего списка от  $a$  до  $b$  включительно. Требуется предусмотреть ситуацию, когда значения  $a$  или  $b$  больше длины списка.
12. Написать функцию, которая принимает список списков и с помощью генераторного выражения создает и возвращает новый список, который содержит все элементы входящих списков.
13. Написать функцию, которая принимает список и с помощью генераторного выражения создает и возвращает новый список, содержащий только уникальные элементы входящего списка.
14. Написать функцию, которая принимает список словарей и с помощью генераторного выражения создает и возвращает новый словарь, который содержит все элементы входящих словарей. Ключи в словарях уникальны.
15. Написать функцию, которая принимает строку и с помощью генераторного выражения создает список, в котором все символы входящей строки смещены на один символ вправо по алфавиту, кроме символов от «ы» до «я». Требуется вернуть строку из созданного списка.

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.
6. Написать функцию, которая принимает два объекта `datetime` и возвращает число по модулю дней между ними.
7. Написать функцию, которая принимает объект `datetime` и возвращает третью среду' месяца.
8. Написать функцию, которая принимает целое число `x` и возвращает список из 20 объектов `datetime` с интервалами между днями равными `x`. За первоначальное значение `datetime` следует принять текущую дату.
9. Написать функцию, которая принимает объект `datetime` и возвращает временную метку (`timestamp`) из данного объекта.
10. Написать функцию, которая принимает строку, содержащую GMT, и возвращает смещенное значение текущей даты и времени.
11. Написать функцию, которая принимает путь к директории и возвращает список с наименованиями файлов, которые хранятся в полученной директории.
12. Написать функцию, которая принимает путь к директории и наименование файла и возвращает полный путь к полученному файлу.
13. Написать функцию, которая принимает список путей и возвращает число путей, которые являются директорией.
14. Написать функцию, которая принимает путь к директории и удаляет в ней все файлы, а вместо них создает директории с тем же наименованием, что и удаленные файлы без значения расширения.
15. Написать функцию относительно задания 8. которая принимает созданные 20 объектов `datetime` и создает для значений года, месяца и дня вложенные директории. Например, значение 2019-09-02 создаст три директории. Корневой будет 2019. внутри неё 09 и т.д. Стоит отметить, что значение другого объекта 2019-07-02 создаст в ранее созданной директории 2019 директорию с наименованием 07.

#### 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.
6. Написать функцию, которая принимает путь к HTML файлу и html тег («p», «h1», «article» и др.) и возвращает количество повторений полученного тега в файле с учетом того, что требуется вернуть только количество тегов, который имеет открывающую и закрывающую часть.
7. Написать функцию, которая принимает путь к директории. путь к файлу и записывает в файл информацию обо всех файлах в директории (формат файла, размер, дата создания). Информация о файлах должна быть пронумерована, а каждый параметр должен быть помечен. Например, «1. Test; Расширение файла: ru; Дата создания: 2019-12-12...» и т.д.
8. Написать функцию, которая принимает путь к двум файлам и возвращает количество одинаковых предложений в файлах.
9. Написать функцию, которая принимает путь к файлу и параметр x, который может являться строкой или списком, и возвращает частоту повторений параметра x в строке. В случае, когда параметром x является список, следует вернуть словарь, в котором в качестве ключей будут искомые строки, а их значениями частота повторений.
10. Написать функцию, которая принимает путь к HTML и путь к CSS файлам и возвращает словарь, в котором ключами выступают тети, идентификаторы или классы в файле CSS. а значениями список списков, где первым элементом внутреннего списка будет наименование тега, которые попадают под стили. указанные в файле CSS, а вторым - номер строк, в которых они находятся. Например. `{'#inline-text' [[ 'h7', 29], ['p', 50]]}`.

### 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 именуется 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.
6. Написать функцию, которая принимает наименование таблицы, имя поля и возвращает все записи по полученному полю из указанной таблицы.
7. Написать функцию, которая переводит структуру всех таблиц существующей базы данных, в формат Json.
8. Написать функцию, которая принимает путь к файлу Json, который содержит описание таблиц и их полей, и создает новые таблицы, указанные в полученном файле.
9. Написать функцию, которая принимает путь к файлу с расширением txt. который содержит данные для заполнения таблицы, и по указанным в файле данным создает таблицу или таблицы и заполняет их данными.
10. Написать функцию, которая строит ER модель базы данных. Можно сгенерировать схему в формате HTML или XML, либо воспользоваться сторонними приложениями и библиотеками.

#### 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<sup>1</sup> в верхнем регистре.
5. Написать класс UserLanguagePreference, который в конструкторе принимает список языков в виде строки, которые использует пользователей 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.
6. Написать класс Shape, который является родительским для класса Square, который содержит конструктор, принимающий длину. Оба класса содержат метод area() для расчета площади. Причем класс Shape имеет площадь равную нулю.
7. Написать класс Point, который в конструкторе принимает координаты точки и содержит методы show(), move() и dist(), где первый метод возвращает координаты точки, второй принимает значения, на которые нужно сместить координаты точки, и последний выводит расстояние по следующей формуле:  $d = \sqrt{(x_1 - x_0)^2 + (y_1 - x_0)^2}$ .

8. Написать класс ORMMapper, который в конструкторе принимает имя другого существующего класса, который содержит только поля и с помощью метода convertTo\_db() создает на основе структуры полученного класса таблицу в базе данных.
9. Написать класс Publisher и несколько различных классов с постфиксом Subscriber. Реализовать между данными классами отношения по паттерну Observer.
10. Написать класс ChortDrawer, который принимает список списков, содержащих координаты точек и содержит методы add\_point(x, y) для добавления новой точки в список, remove\_point(x, y) для удаления точек с указанными координатами. Класс должен правильно использовать инкапсуляцию/

## 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.
6. Написать функцию, которая принимает путь к изображению и поворачивает его на 90 градусов, сохраняя его по тому же пути.
7. Написать функцию, которая принимает путь к директории с изображениями и список расширений {png, jpg, ico и т.д.) и создает директории по полученному пути с наименованием расширения, в которые записывает конвертированные исходные изображения по полученным форматам.
8. Написать функцию, которая принимает путь к архиву и путь для сохранения и распаковывает из него все изображения. сохраняя их по указанному пути.
9. Написать функцию, которая принимает путь к изображению и список пикселей и заменяет полученные пиксели прозрачным цветом, сохраняя изображение в той же директории.
10. Написать функцию, которая принимает путь к изображению. искомый цвет пикселя и цвет, на который требуется его заменить, и заменяет полученный цвет на картинке цветом замены. сохраняя новое изображение в той же директории.
11. Написать класс, который содержит функцию, принимающую путь к изображению в формате gif и делает его раскадровку, сохраняя изображения по тому же пути, что и gif. Вторая функция будет принимать директорию с изображениями и записывать их в gif.
12. Написать класс, который содержит функцию, выполняющую скриншот экрана и функцию, которая сохраняет изображение, находящееся в буфере, если такое существует.
13. Написать функцию, которая принимает путь к файлу с изображением и два списка координат пикселей и выполняет перемещение пикселей двух списков.

14. Написать функцию, которая принимает путь к изображениям и склеивает их в одно изображение.

#### 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, которые 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.
6. Написать GUI приложение, которое представляет собой библиотеку с поиском книг по автору и наименованию, добавлению новых книг и чтению книг в форматах epub, fb2 и pdf.
7. Написать GUI приложение, которое позволяет рисовать в определенном поле с помощью курсора мыши. В приложении также можно выбрать цвет и толщину кисти.
8. Написать GUI приложение, которое представляет собой несложный мессенджер, работающий с другим пользователем по IP адресу.
9. Написать GUI приложение, которое представляет собой книгу заметок, позволяющую создавать новые заметки, редактировать и удалять существующие. Приложение позволяет создавать группы заметок и выполнять сортировку по дате.
10. Написать GUI приложение для проверки регулярных выражений. В двух полях пользователю требуется ввести текст и регулярное выражение. В поле текста после применения регулярного выражения требуется выделить найденную часть, если регулярное выражение написано с ошибкой, необходимо уведомить об этом пользователя.
11. Написать GUI приложение для работы с базой данных. Приложение позволяет выполнять соединение с БД, а также выполнять запросы и получать результаты ответов.

#### 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.
6. Написать функцию, которая принимает матрицу и возвращает её определитель.
7. Написать функцию, которая принимает матрицу и возвращает сумму элементов её главной диагонали.
8. Написать функцию, которая генерирует шесть случайных чисел между 30 и 100.
9. Написать функцию, которая генерирует матрицу 5x5 и заполняет её случайными значениями и возвращает минимальное и максимальное значения полученной матрицы.
10. Написать функцию, которая принимает список целых чисел и возвращает список с перемешанными числами.
11. Написать функцию, которая строит гистограмму популярных языков программирования. Данные можно взять из последних данных ТЮВИ.
12. Написать функцию, которая читает таблицу из файла .csv, содержащую шапку и процентную долю каждого столбца и строит по ней круговую диаграмму.
13. Написать функцию, которая создает случайную точечную диаграмму со случайным размером шариков.
14. Написать функцию, которая строит точечную диаграмму повышения уровня моря за последние 100 лет.

### 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*.

#### 4.3 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?
  - a) 100 100
  - b) 100 150
  - c) 50 100
  - d) 100 50

```

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)

```

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.

#### 4.4 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><b>Частично знает</b> 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><b>В основном умеет</b> взаимодействовать с инженерами в процессе создания систем искусственного интеллекта, 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</p>	<p><b>Знает</b> 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><b>Умеет</b> взаимодействовать с инженерами в процессе создания систем искусственного интеллекта, 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</p>	<p><b>Демонстрирует глубокое знание и понимание:</b> 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><b>Умеет полностью верно и самостоятельно</b> взаимодействовать с инженерами в процессе создания систем искусственного интеллекта, 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</p>

Индикаторы достижения компетенций	Критерии оценивания компетенций		
	Пороговый уровень	Продвинутый уровень	Высокий уровень
	языка программирования Python	языка программирования Python	проверке выполнения требований к системе на основе применения языка программирования Python

## 5 Учебно-методическое и информационное обеспечение дисциплины

### 5.1 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).

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

Учебно-методические материалы для выполнения лабораторных работ.

### 5.3 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/>

### 5.4 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](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/>