Лабораторная работа №4

**Тема**: Работа с файлами, классами, сериализаторами, регулярными выражениями и стандартными библиотеками.

**Цель**: освоить базовый синтаксис языка Python, приобрести навыки работы с файлами, классами, сериализаторами, регулярными выражениями и стандартными библиотеками и закрепить их на примере разработки интерактивных приложений.

**Cсылки**:

1. <https://docs.python.org/>
2. <https://pythonworld.ru/>
3. <https://smartiqa.ru/courses/python/lesson-1>
4. <https://pythonru.com/uroki/vvedenie-uroki-po-python-dlja-nachinajushhih>
5. <https://pythontutor.ru/>
6. <https://pythonchik.ru/osnovy>
7. <https://younglinux.info/python/course>
8. <https://www.w3schools.com/python/default.asp>

Для защиты ЛР необходимо оформить Отчет со скринами кода и результатов его выполнения

**Требования к выполнению**

1. Программа должна быть снабжена комментариями на английском языке, в которых необходимо указать краткое предназначение программы, номер лабораторной работы и название, версию программы, Ф.И.О. разработчика и дату разработки.
2. Программа должна быть снабжена дружелюбным и интуитивно понятным интерфейсом
3. Выполнить документирование кода для получения справки по каждой функции
4. Для выполнения каждого задания разработать **пользовательские классы и обеспечить их наследование**. Реализовать примеры использования:

* статических и динамических атрибутов класса,
* полиморфизма,
* специальных(магических методов),
* super(),
* геттеров и сеттеров,
* свойств класса,
* примесей

1. При разработке программ рекомендуется придерживаться принципа: за решение одной конкретной задачи должна отвечать одна функция.
2. Все функции необходимо сгруппировать в модулях, согласно их логике их работы.
3. Разработанные основные функции, размещенные в отдельных модулях, нужно подключить в другом модуле, где будет происходить тестирование данных функций.
4. В программах предусмотреть возможность повторного выполнения без выхода из программы и защиту от ввода некорректных пользовательских данных. Для этих целей рекомендуется разработать отдельные функции.
5. Обеспечить обработку конкретных классов исключений

**Индивидуальные задания по вариантам**

**Задание 1.** Исходные данные представляют собой словарь. Необходимо поместить их в файл, используя сериализатор. Организовать считывание данных, поиск, сортировку в соответствии с индивидуальным заданием. Обязательно использовать классы. Реализуйте два варианта: 1)формат файлов CSV; 2)модуль pickle

|  |  |
| --- | --- |
| Вар-т | Условие |
| 29 | Реализуйте школьный класс (фамилия и инициалы, дата рождения, месяц рождения, год рождения). Напечатайте список учеников, рожденных в месяце, введенном с клавиатуры. |

**Задание 2.** В соответствии с заданием своего варианта составить программу для анализа текста. Считать из исходного файла текст. Используя регулярные выражения получить искомую информацию (см. условие), вывести ее на экран и сохранить в другой файл. Заархивировать файл с результатом с помощью модуля zipfile и обеспечить получение информации о файле в архиве.

Также выполнить общее задание – определить и сохранить в файл с результатами:

* количество предложений в тексте;
* количество предложений в тексте каждого вида отдельно (повествовательные, вопросительные и побудительные);
* среднюю длину предложения в символах (считаются только слова);
* среднюю длину слова в тексте в символах;
* количество смайликов в заданном тексте. Смайликом будем считать последовательность символов, удовлетворяющую условиям:

первым символом является либо «;» (точка с запятой) либо «:» (двоеточие) ровно один раз;

далее может идти символ «-» (минус) сколько угодно раз (в том числе символ минус может идти ноль раз);

в конце обязательно идет некоторое количество (не меньше одной) одинаковых скобок из следующего набора: «(», «)», «[», «]»;

внутри смайлика не может встречаться никаких других символов. Например, эта последовательность является смайликом: «;---------[[[[[[[[». Эти последовательности смайликами не являются: «]», «;--»,«:»,«)».

|  |  |
| --- | --- |
| Вар-т | Условие |
| 29 | Вывести все слова, включающие сочетание букв нижнего регистра и цифр  Определить, является ли заданная строка IP-адресом, записанным в десятичном виде. Примеры правильных выражений: 127.0.0.1, 255.255.255.0. Примеры неправильных выражений: 1300.6.7.8, abc.def.gha.bcd.  определить количество строчных букв;  найти первое слово, содержащее букву 'v' и его номер;  вывести строку, исключив из нее слова, начинающиеся с 's' |

**Задание 3.**  В соответствии с заданием своего варианта доработать программу из ЛР3, использовав класс и обеспечить:

а) определение дополнительных параметров среднее арифметическое элементов последовательности, медиана, мода, дисперсия, СКО последовательности;

б) с помощью библиотеки matplotlib нарисовать графики разных цветов в одной координатной оси:

* график по полученным данным разложения функции в ряд, представленным в таблице,
* график соответствующей функции, представленной с помощью модуля math. Обеспечить отображение координатных осей, легенды, текста и аннотации.



Здесь x – значение аргумента, F(x) – значение функции, n – количество просуммированных членов ряда, Math F(x) – значение функции, вычисленное с помощью модуля math.

в) сохранить графики в файл

|  |  |
| --- | --- |
| **Вар-т** | **Условие** |
| 29 |  |

**Задание 4.** В соответствии с заданием своего варианта разработать базовые классы и классы наследники.

Требования по использованию классов:

Абстрактный класс «Геометрическая фигура» содержит абстрактный метод для вычисления площади фигуры (<https://docs.python.org/3/library/abc.html> )

Класс «Цвет фигуры» содержит свойство для описания цвета геометрической фигуры (<https://docs.python.org/3/library/functions.html#property> )

Класс «Прямоугольник» (Круг, Ромб, Квадрат, Треугольник и т.д.) наследуется от класса «Геометрическая фигура». Класс должен содержать конструктор по параметрам «ширина», «высота» (для другого типа фигуры соответствующие параметры, например, для круга задаем «радиус») и «цвет». В конструкторе создается объект класса «Цвет фигуры» для хранения цвета. Класс должен переопределять метод, вычисляющий площадь фигуры <https://docs.python.org/3/library/math.html> .

Для класса «Прямоугольник»(тип фигуры в инд. задании)

определить метод, который возвращает в виде строки основные параметры фигуры, ее цвет и площадь. Использовать метод format (<https://pyformat.info/> )

название фигуры должно задаваться в виде поля данных класса и возвращаться методом класса.

В корневом каталоге проекта создайте файл main.py для тестирования классов. Используйте конструкцию, описанную в <https://docs.python.org/3/library/__main__.html>

Пример объекта: Прямоугольник синего цвета шириной 5 и высотой 8.

Программа должна содержать следующие базовые функции:

1. ввод значений параметров пользователем;

2) проверка корректности вводимых данных;

3) построение, закрашивание фигуры в выбранный цвет, введенный с клавиатуры, и подпись фигуры текстом, введенным с клавиатуры;

4) вывод фигуры на экран и в файл.

|  |  |
| --- | --- |
| Вар | Условие |
| 29 | Построить правильный n-угольник со стороной a. |

**Задание 5.** В соответствии с заданием своего варианта исследовать возможности библиотека NumPy при работе с массивами и математическими и статическими операциями. Сформировать целочисленную матрицу А[n,m] с помощью генератора случайных чисел (random).

а) Библиотека NumPy.

1. Создание массива. Функции array() и values().

2. Функции создания массива заданного вида.

3. Индексирование массивов NumPy. Индекс и срез.

4. Операции с массивами. Универсальные (поэлементные) функции.

б) Математические и статистические операции.

1. Функция mean()

2. Функция median()

3. Функция corrcoef()

4. Дисперсия var().

5. Стандартное отклонение std()

|  |  |
| --- | --- |
| Вар | Условие |
| 29 | Вычислить сумму элементов матрицы, расположенных ниже главной диагонали.  Вычислить стандартное отклонение для элементов главной диагонали  матрицы. Ответ округлите до сотых. Вычисление стандартного отклонения  выполнить двумя способами: через стандартную функцию и через  программирование формулы |

Дополнительное задание для тех, кто стремится к большему (не обязательное):

Используя любой из наборов данных (datasets) на ресурсе: <https://www.kaggle.com/datasets> исследовать основные возможности библиотеки Pandas. См. Анализ Данных Python методичка.pdf

*Задание а. Библиотека Pandas. Структуры Series и DataFrame*

1. Библиотека Pandas. Импорт библиотеки.

2. Структура Series.

3. Создание Series .

4. Функция display.

5. Доступ к элементам Series с использованием .loc или .iloc.

6. Объект DataFrame. Создание.

*Задание б. Основные операции*

2. Получение информации о датафрейме (по каждому параметру)

5. Индексация и извлечение данных: статистические методы (во сколько раз среднее значение выбранного показателя с максимальным значением другого параметра (например, самый молодой, самый быстрый и т.д.) выше среднего выбранного показателя с минимальным значением )

Пример :

Определите, во сколько раз средняя сила удара (ShotPower) самых агрессивных

игроков (игроков с максимальным значением показателя "Агрессивность"

(Aggression)) выше средней силы удара игроков с минимальной агрессией.

Ответ округлите до сотых.

Какова средняя скорость (SprintSpeed) футболистов, зарплата (Wage) которых ниже среднего? Ответ округлите до сотых.

**Контрольные вопросы**

1. Работа с файлами в Python
2. Открытие и закрытие файлов
3. Текстовые файлы
4. Файлы CSV
5. Бинарные файлы
6. Основные свойства и методы файлов.
7. Сериализация и десериализация. CSV- и JSON-файлы
8. Модуль shelve
9. Модуль OS и работа с файловой системой
10. Программа подсчета слов
11. Запись и чтение архивных zip-файлов
12. Работа с датами и временем в Python
13. Модуль datetime
14. Операции с датами
15. Классы и объекты
16. Инкапсуляция, атрибуты и свойства
17. Наследование
18. Переопределение функционала базового класса
19. Атрибуты классов и статические методы
20. Класс object. Строковое представление объекта
21. Обработка ошибок и исключений в Python
22. Конструкция try...except...finally
23. except и обработка разных типов исключений
24. Генерация исключений и создание своих типов исключений
25. Модули. Пакеты. Классификация.
26. Подключение и использование модулей и пакетов. Специальные атрибуты.
27. Стандартная библиотека Python.
28. Регулярные выражения.
29. Визуализация данных. Инструменты визуализации. Matplotlib
30. NumPy.
31. Работа с табличными данными. Pandas