ГУАП  
КАФЕДРА № 41

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доцент |  |  |  | А.В. Аграновский |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| РАБОТА С ДАННЫМИ В ФОРМАТЕ CSV |
| по курсу: ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | М023 |  |  |  | Д.А. Трегуб |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

ПОСТАНОВКА ЗАДАЧИ

|  |  |  |  |
| --- | --- | --- | --- |
| №  варианта | Название файла набора данных | Способ  отображения  стран | Дополнительное  задание |
| 1 | AccessToComputersFromHome.csv | A | A |

Задание № 1

Требования к программе:  
Цель программы: Чтение файла csv, обработка находящихся в нём данных, создание на их основе графиков, с помощью выбора случайных значений, а также провести дополнительную обработку данных, представляющую собой нахождение средних значений и среднеквадратичных отклонений для всех стран.. На вход подаётся 1 параметр: количество отображаемых на графиках стран. Выходными данными является окно с графиками, обработанные данные в файле DataOut.csv, а также сообщение "successful" или сообщение по ошибке.

МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Математического обоснования требует третье задание. Для нахождение среднего значения необходимо иcпользовать формулу:

Для нахождения среднеквадратичных отклонений необходимо использовать формулу:

ОПИСАНИЕ РАЗРАБОТАННОЙ ПРОГРАММЫ

Таблица 1 - Описание переменных

|  |  |  |
| --- | --- | --- |
| Имя переменной | Тип переменной | Назначение |
| n | Int | Количество отображаемых стран |
| LOCATION | list | Список стран из файл |
| TIME | list | Список временных отметок |
| VALUE | List | Список значений |
| Uniq\_LOCATION | List | Список стран без повторений |
| reader | Reader/DictReader | Построчный итератор для чтения файла csv |
| headers | List | Список для хранения заголовков |
| loc\_help | List | Вспомогательный список с нумерацией стран |
| Num\_LOCATION | List | Список с номером страны без повторения |
| fig | Figure | Окно для отображения графков |
| ax | Axes | Контейнер для хранения отдельных графиков |
| first | Int | Первое вхождение страны в список LOCATION |
| last | Int | Последнее вхождение страны в список LOCATION |
| this\_time | List | Хранения значений времени для конкретной страны |
| this\_value | List | Хранения значений для конкретной страны |
| arr1 | Np.array | Массив на основе this\_time для построения графика |
| arr2 | Np.array | Массив на основе this\_value для построения графика |
| average | List | Список со средними значениями |
| cwadro | List | Список со среднеквадратичными отклонениями |
| colva | Int | Количество стран для вычисление среднего значения |
| first1 | Int | Копия первого вхождение страны в список LOCATION для вычисления среднеквадратичного отклонения |
| data\_help | List | Массив с данными для записи в файл |
| writer | Writer | Построчный итератор для записи данных в файла csv |
| fileOut | File | Файл для записи |
| file | File | Файл для чтения |

На вход подаётся функции подаётся n – количество отображаемых стран.

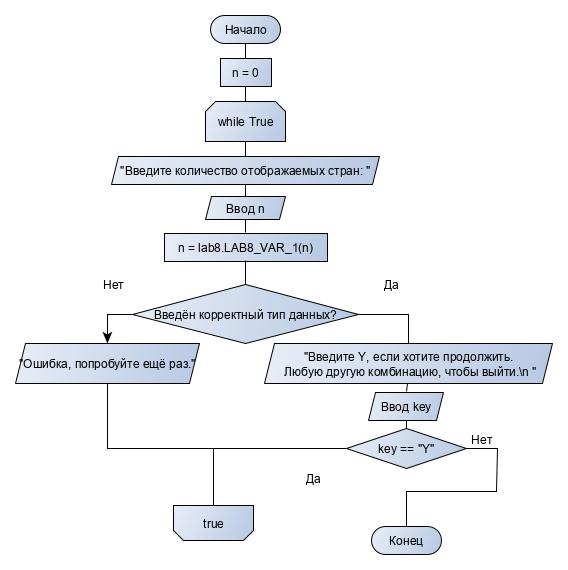


Рисунок 1 – функция main

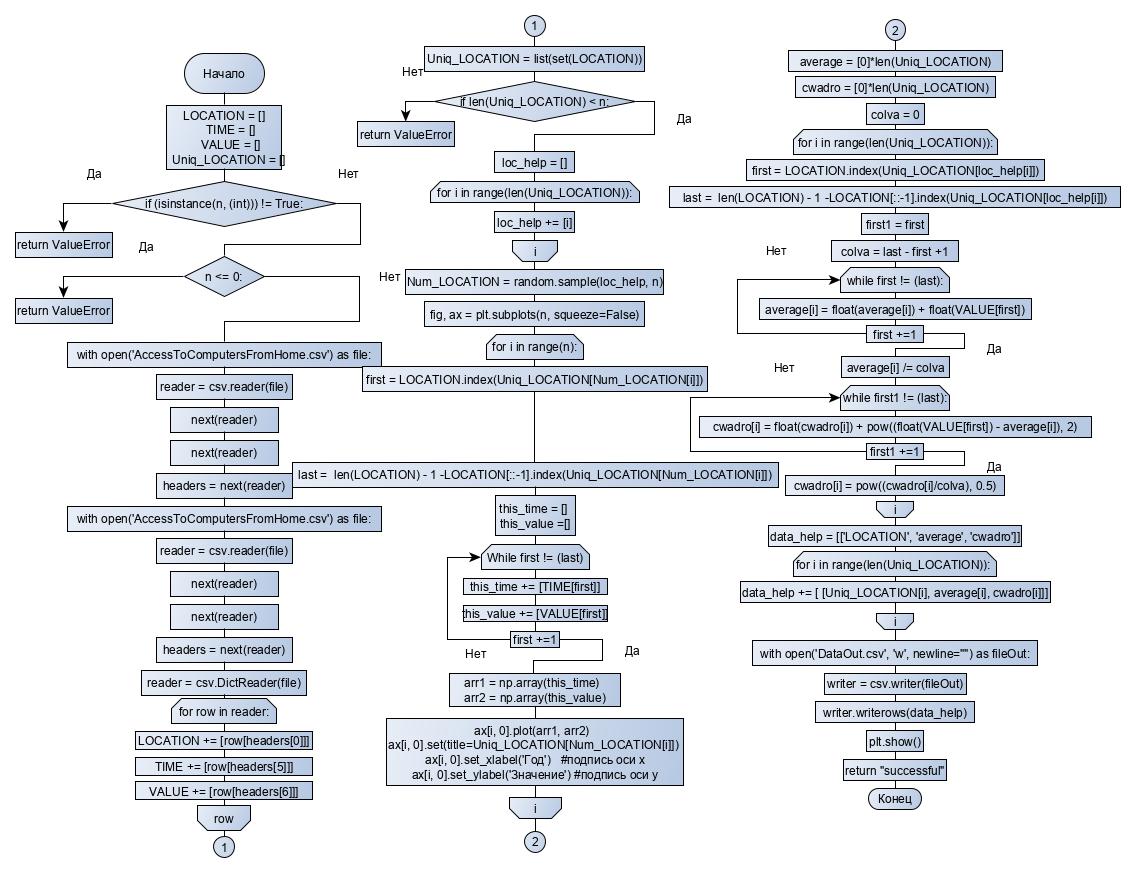


Рисунок 2 - функция LAB8\_VAR\_1

ОПИСАНИЕ ТЕСТОВОГО НАБОРА

Таблица 2 - Тестовый набор

|  |  |  |  |
| --- | --- | --- | --- |
| Описание тестового случая | Входные данные | Ожидаемый результат | Результат теста |
| Классический случай | 2 | " successful" | OK |
| Отрицательное количество стран | -1 | ValueError | OK |
| Несоветские типов данных (данные – строка) | one | ValueError | OK |
| Несоветские типов данных (данные – список) | [1] | ValueError | OK |
| Несоветские типов данных (данные – числа с плавающей точкой) | 1.5 | ValueError | OK |
| Ошибка с превышение количества стран для анализа | 42 | ValueError | OK |

ПРИМЕР РАБОТЫ ПРОГРАММЫ



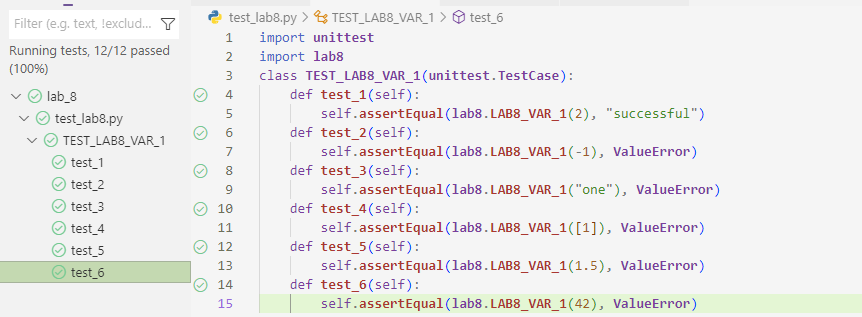
Рисунок 3 - Результат работы тестов до написания программы

Рисунок 4 - Результат работы тестов после написания программы

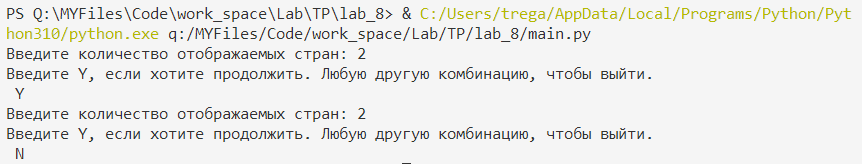


Рисунок 5 - Результат работы программы(терминал)

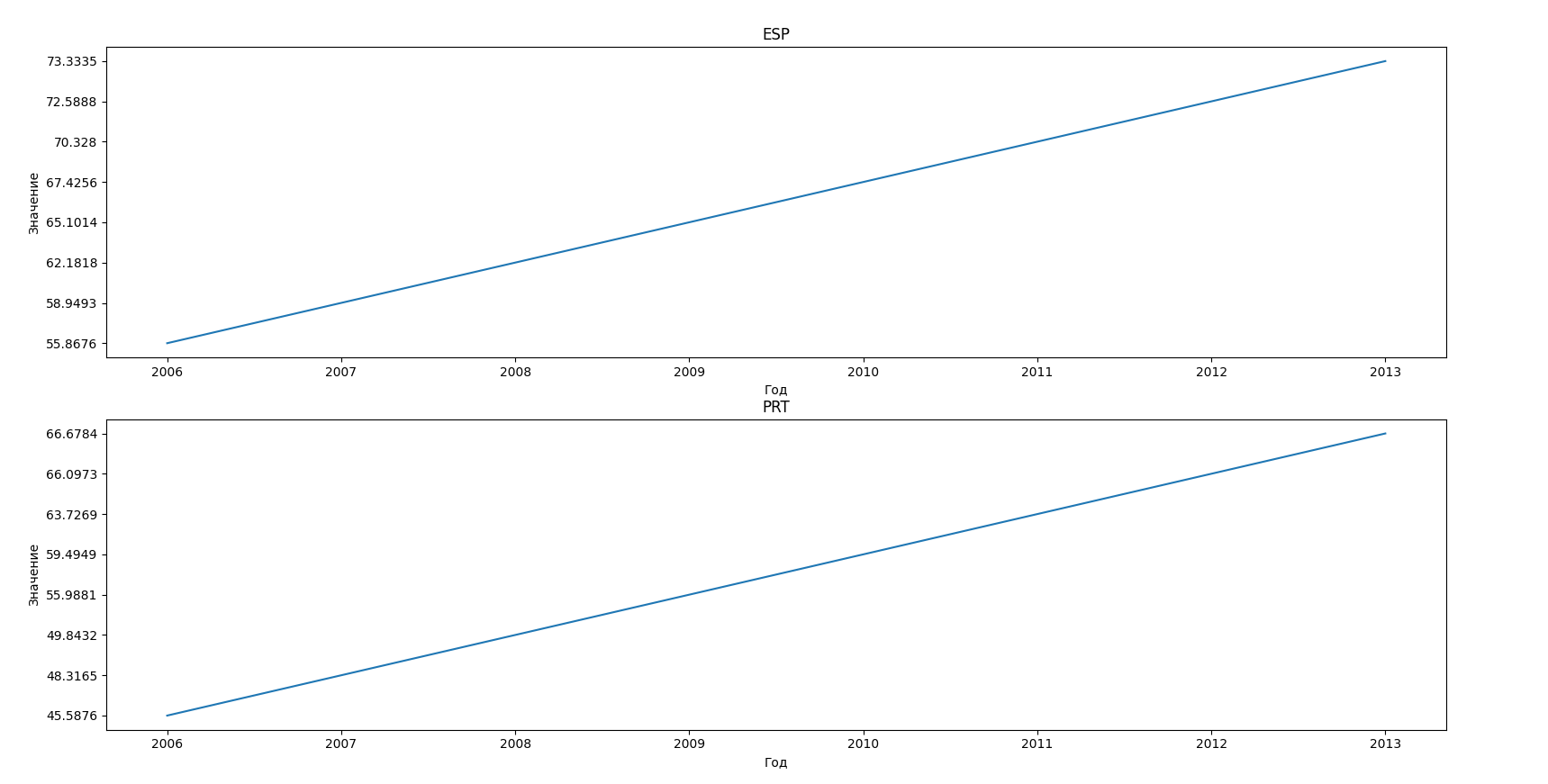


Рисунок 6 - Результат работы программы(графики)

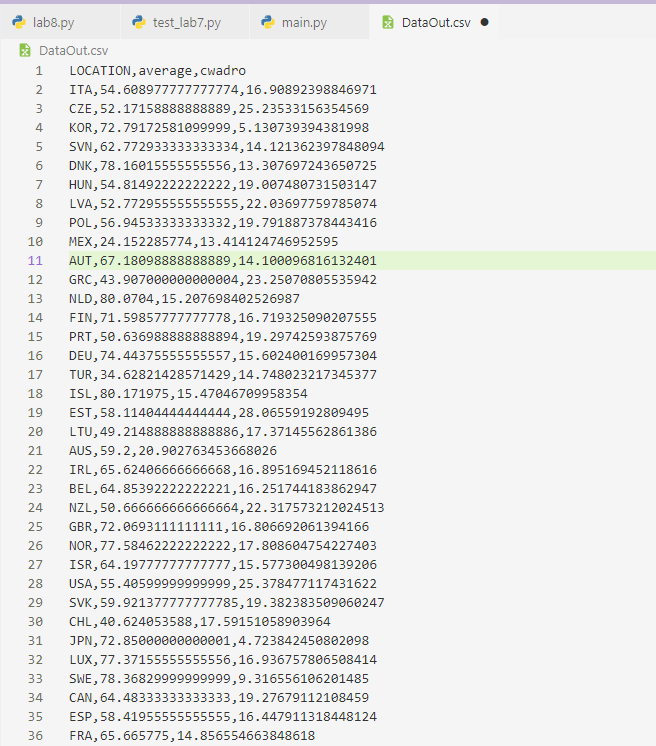


Рисунок 7 - Результат работы программы(запись в файл)

ВЫВОД

Были улучшены навыки работы с python, получены навыки работы с библиотекой csv для чтения файла, обработки данных из него, и записи в файл, была выполнена поставленная задача и дополнительное задание по обработке данных. Составленная программа была успешно протестирована.

ПРИЛОЖЕНИЯ

Приложение 1

Код программы

Код тестов:

import **unittest**

import **lab8**

class **TEST\_LAB8\_VAR\_1**(**unittest**.**TestCase**):

    def **test\_1**(self):

        self.**assertEqual**(**lab8**.**LAB8\_VAR\_1**(2), "successful")

    def **test\_2**(self):

        self.**assertEqual**(**lab8**.**LAB8\_VAR\_1**(-1), **ValueError**)

    def **test\_3**(self):

        self.**assertEqual**(**lab8**.**LAB8\_VAR\_1**("one"), **ValueError**)

    def **test\_4**(self):

        self.**assertEqual**(**lab8**.**LAB8\_VAR\_1**([1]), **ValueError**)

    def **test\_5**(self):

        self.**assertEqual**(**lab8**.**LAB8\_VAR\_1**(1.5), **ValueError**)

    def **test\_6**(self):

        self.**assertEqual**(**lab8**.**LAB8\_VAR\_1**(42), **ValueError**)

Код функции LAB8\_VAR\_1:

import **csv**

import **matplotlib**.**pyplot** as **plt**

import **numpy** as **np**

import **random**

def **LAB8\_VAR\_1**(n):

*#иницилизация списков*

    LOCATION = []

    TIME = []

    VALUE = []

    Uniq\_LOCATION = []

    if (**isinstance**(n, (**int**))) != True: return **ValueError** *#Проверка соответсвия типам*

    elif n <= 0: return **ValueError**                       *#Проверка для соответсвия условиям*

*#Получение заголовков из файла*

    with **open**('AccessToComputersFromHome.csv') as file: *#Открытие файла*

        reader = **csv**.**reader**(file) *#Создания объекта reader для чтения файла*

**next**(reader)              *#Прохождение строки с коментарием*

**next**(reader)              *#Прохождение пустой строки*

        headers = **next**(reader)    *#Получение заголовка*

*#Чтение файла и сохранение данных в файл*

    with **open**('AccessToComputersFromHome.csv') as file: *#Открытие файла*

        reader = **csv**.**reader**(file)    *#Создания объекта reader для чтения файла*

**next**(reader)                 *#Прохождение строки с коментарием*

**next**(reader)                 *#Прохождение пустой строки*

        reader = **csv**.**DictReader**(file)*#Создание объекта DictReader для чтения по определёным позициям*

        for row in reader:           *#Запись данных в списки по определённым заголовкам*

            LOCATION += [row[headers[0]]]

            TIME += [row[headers[5]]]

            VALUE += [row[headers[6]]]

*#Выбор случайных стран*

    Uniq\_LOCATION = **list**(**set**(LOCATION)) *#Создание списка для хранения стран в единственном экземпляре*

    if **len**(Uniq\_LOCATION) < n: return **ValueError** *#Провека для соответвия условиям*

    loc\_help = [] *#Вспомогательный список для нумерации стран*

    for i in **range**(**len**(Uniq\_LOCATION)): loc\_help += [i]

    Num\_LOCATION = **random**.sample(loc\_help, n) *#выбор случайных стран в количестве введёнх пользователем*

*#Создание графиков*

    fig, ax = **plt**.**subplots**(n, squeeze=False)

*#Цикл по выбранным странам*

    for i in **range**(n):

        first = LOCATION.**index**(Uniq\_LOCATION[Num\_LOCATION[i]])                          *#Получение первого вхождение страны*

        last =  **len**(LOCATION) - 1 -LOCATION[::-1].**index**(Uniq\_LOCATION[Num\_LOCATION[i]]) *#Получение последнего вхождения страны*

        this\_time = []  *#список значений времени для конкретной страны*

        this\_value =[]  *#список значений для конкретной страны*

        while first != (last):  *#цикл для записи конкретных значений*

                this\_time += [TIME[first]]

                this\_value += [VALUE[first]]

                first +=1

        arr1 = **np**.**array**(this\_time)  *#Создания массивов для создания графика*

        arr2 = **np**.**array**(this\_value)

        ax[i, 0].plot(arr1, arr2)  *#Отображение данных на графике*

        ax[i, 0].set(title=Uniq\_LOCATION[Num\_LOCATION[i]])    *#Размещение заголовока над "Axes"*

        ax[i, 0].set\_xlabel('Год')   *#подпись оси x*

        ax[i, 0].set\_ylabel('Значение') *#подпись оси y*

*#Дополнительное задание*

    average = [0]\***len**(Uniq\_LOCATION) *#Список со средними значениями*

    cwadro = [0]\***len**(Uniq\_LOCATION)  *#Список со среднекваратичными отклонениями*

    colva = 0                       *#Переменная для хранения количества упоминаний конкретной странн*

    for i in **range**(**len**(Uniq\_LOCATION)):

        first = LOCATION.**index**(Uniq\_LOCATION[loc\_help[i]])  *#Получение первого вхождение страны*

        last =  **len**(LOCATION) - 1 -LOCATION[::-1].**index**(Uniq\_LOCATION[loc\_help[i]]) *#Получение последнего вхождения страны*

        first1 = first  *#Копия первого вхождения страны*

        colva = last - first +1 *#Вычисление количества стран*

        while first != (last):  *#цикл для вычисления суммы значений*

            average[i] = **float**(average[i]) + **float**(VALUE[first])

            first +=1

        average[i] /= colva *#Вычисление среднего значения*

        while first1 != (last): *#цикл для вычисления квадрата среднеквадратичного отклонения значеня*

            cwadro[i] = **float**(cwadro[i]) + **pow**((**float**(VALUE[first]) - average[i]), 2)

            first1 +=1

        cwadro[i] = **pow**((cwadro[i]/colva), 0.5) *#среднеквадратичное отлонение*

*#Вывод в файл*

    data\_help = [['LOCATION', 'average', 'cwadro']] *#Создание списка для записи в файл*

    for i in **range**(**len**(Uniq\_LOCATION)): data\_help += [ [Uniq\_LOCATION[i], average[i], cwadro[i]]]

    with **open**('DataOut.csv', 'w', newline="") as fileOut: *#Открытие файла для записи*

        writer = **csv**.**writer**(fileOut)    *#Создание объекта writer*

        writer.**writerows**(data\_help)     *#Запись по строкам в файл*

**plt**.**show**() *#Отрисовка графика*

    return "successful" *#Возвращение сообщения о завершение функции*

Код функции main:

import **lab8**

*#Иницилизация переменных*

n = 0

key = ""

result = ""

*#Начало бесконечного цикла*

while True:

    try:

        n = **int**(**input**("Введите количество отображаемых стран: "))    *#Ввод амплитуды*

        result = **lab8**.**LAB8\_VAR\_1**(n)       *#Вызов функции*

    except **ValueError**:

**print**("Ошибка, попробуйте ещё раз.")

    key = **str**(**input**("Введите Y, если хотите продолжить. Любую другую комбинацию, чтобы выйти.\n "))*#Ввод комбинации для продолжения или завершения программы*

    if(key == "Y"):

        continue *#Продолжение программы*

    else:

            break *#Завершение программы*

*#Конец бесконечного цикла*