Министерство науки и образования РФ

Федеральное государственное бюджетное учреждение

высшего образования

**«Тверской государственный технический университет»**

(ТвГТУ)

Кафедра программного обеспечения

**Отчет по лабораторной работе №1**

по дисциплине: «Новые технологии в разработке ПС»

Тема: «Работа с библиотекой Pandas»

|  |
| --- |
| Выполнил:  студент группы  М.ПИН.РИС – 22.11  Удальцов З.С. |
| Проверила:  ассистент кафедры ПО  Корнеева Е.И. |

Тверь 2022

**Задачи**

**Общие задачи**

1. Загрузите данные из репозитория (по ссылке, либо локально). Прочитайте csv файл и распечатайте первые 8-15 строк.
2. Переведите данные в формат DataFrame. Убедитесь, что перевод выполнен корректно.
3. Сохраните в файл вывода (для notebook на экране) количество строк, столбцов и тип данных для каждого столбца.
4. Напишите программу для расчета среднего, максимального и минимального значений по выбранному количественному столбцу DataFrame. Допишите результат в файл вывода.
5. Создать столбец, в котором будет отображаться год наблюдения НЛО (вынуть год из столбца 1).
6. Создайте в таблице строку с текущим годом записи в любом штате/области США/Канады.

**Вариант 8**

1. Получите список времени наблюдения НЛО и разделите их на утро, день, вечер, ночь.
2. Посчитайте частоту отчетов об НЛО по годам и странам

**Задание**

Напишите программу, которая будет эмулировать калькулятор. Калькулятор принимает на вход выражение в текстовой форме. После нажатия на клавишу «=», введенная строка преобразуется в обратную польскую нотацию ОПН. Затем происходит вычисление выражения, записанного в ОПН.

**Алгоритм решения**

**Вариант 8.1**

|  |
| --- |
| Импортируем pandas as pd    times = []  times\_mor = []  times\_day = []  times\_ev = []  times\_night = []  data = pd.read\_csv('f:/magistratura/kurs\_1/sem\_1/nsr/lab\_1/ufo\_sighting\_data.csv')    Для i in data['date\_time']:  time = i.split(' ')[1]  times.append(time)    Для i in times:  t = i.split(':')  minut = int(t[0]) \* 60 + int(t[1])  if(minut > 0 and minut <= 360):  times\_night.append(i)  Иначе если (minut > 360 and minut <= 720):  times\_mor.append(i)  Иначе если (minut > 720 and minut <= 1080):  times\_day.append(i)  else:  times\_ev.append(i)      print("morning: ")  Для i in times\_mor:  print(i)    print("day: ")  Для i in times\_day:  print(i)    print("evening: ")  Для i in times\_ev:  print(i)    print("night: ")  Для i in times\_night:  print(i) |

**Вариант 8.2**

|  |
| --- |
| Импортируем pandas as pd    years = {}  countries = {}    data = pd.read\_csv('f:/magistratura/kurs\_1/sem\_1/nsr/lab\_1/ufo\_sighting\_data.csv')    Для i in data['date\_time']:  date = i.split(' ')[0]  year = date.split('/')[2]  count = 1    Для y in years.keys():  Если year == y:  count = years[y] + 1    years[year] = count    print("years: ")  print(years)    Для i in data['country']:  count = 1    Для c in countries.keys():  Если i == c:  count = countries[c] + 1    countries[i] = count    print("countries: ")  print(countries) |

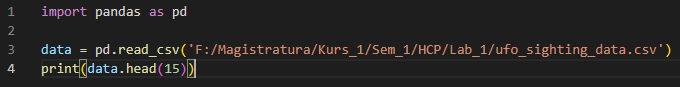
**Калькулятор**

|  |
| --- |
| Функция rpn(s):  lex = parse(s)  s2 = []  r = []  oper = ["+", "-", "\*", "/", "(", ")"]    Для a in lex:  Если a == "(":  s2 = [a] + s2  Иначе если a in oper:  Если s2 == []:  s2 = [a]  Иначе если a == ")":  while(true):  q = s2[0]  s2 = s2[1:]  Если q == "(":  выйти  r += [q]  Иначе если prty(s2[0]) < prty(a):  s2 = [a] + s2  else:  while(true):  Если s2 == []:  выйти  q = s2[0]  Если prty(q) < prty(a):  выйти  r += [q]  s2 = s2[1:]  s2 = [a] + s2  else:  r += [a]    while(s2 != []):  q = s2[0]  r += [q]  s2 = s2[1:]  вернуть r    Функция prty(o):  Если o == "+" or o == "-":  вернуть 1  Иначе если o == "\*" or o == "/":  вернуть 2  Иначе если o == "(":  вернуть 0    Функция parse(s):  delims = ["+", "-", "\*", "/", "(", ")"]  lex = []  tmp = ""  Для a in s:  Если a != " ":  Если a in delims:  Если tmp != "":  lex += [tmp]  lex += [a]  tmp = ""  else:  tmp += a  Если tmp != "":  lex += [tmp]    вернуть lex    Функция rpn\_calc(formula):  s = []  Для lex in formula:  Если lex[0].isdigit():  s.append(float(lex))  else:  a2 = s.pop()  a1 = s.pop()  Если lex == '+':  s.append(a1 + a2)  Если lex == '-':  s.append(a1 - a2)  Если lex == '\*':  s.append(a1 \* a2)  Если lex =='/':  s.append(a1 / a2)  вернуть s.pop()    Функция getrpn(a):  вернуть rpn(a)    Функция solve(a):  вернуть rpn\_calc(rpn(a)) |

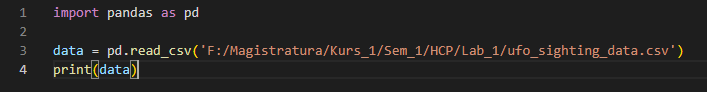
|  |
| --- |
| из decimal Импортируем divisionbyzero  из tkinter Импортируем \*  из tkinter Импортируем messagebox  из rpn Импортируем \*    Функция clicked():  try:  res = solve(txt.get())  lblres.configure(text = "решение: {}".format(res))  except zerodivisionerror:  messagebox.showerror('ошибка', 'деление на ноль')  except:  messagebox.showerror('ошибка', 'неверный ввод')    window = tk()  window.title("калькулятор")  window.geometry('480x240')    lbl = label(window, text = "пример: ")  lbl.grid(column = 0, row = 0)    txt = entry(window, width = 50)  txt.grid(column = 1, row = 0)  txt.focus()    lblres = label(window, text = "решение: ")  lblres.grid(column = 0, row = 1)    btn = button(window, text = "решение", command = clicked, height = 10, width = 50)  btn.grid(column = 1, row = 2)  window.mainloop() |

**Код**

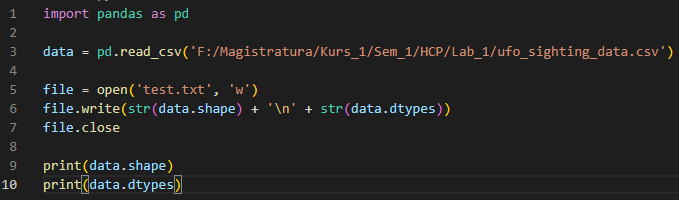
**Задача 1**



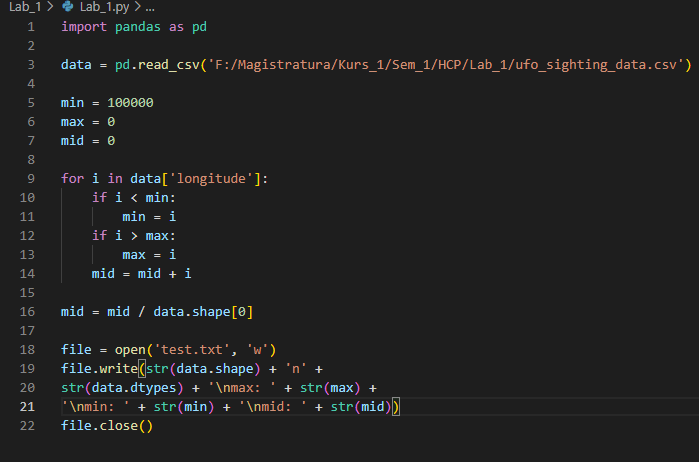
**Задача 2**



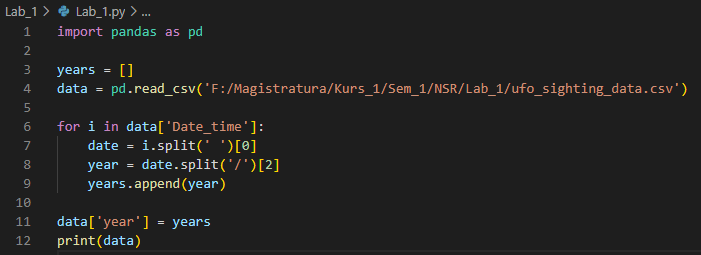
**Задача 3**



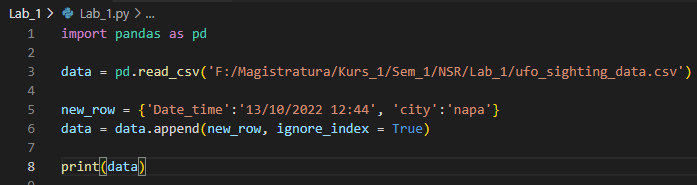
**Задача 4**



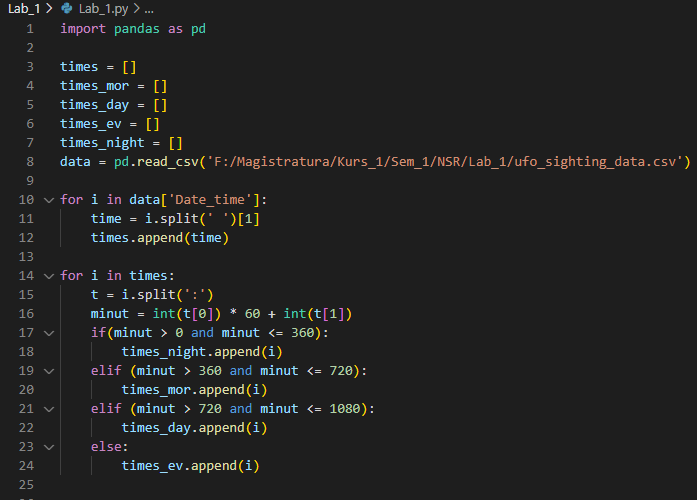
**Задача 5**

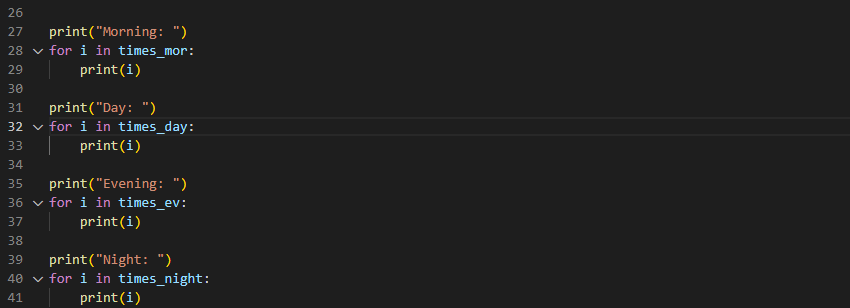


**Задача 6**

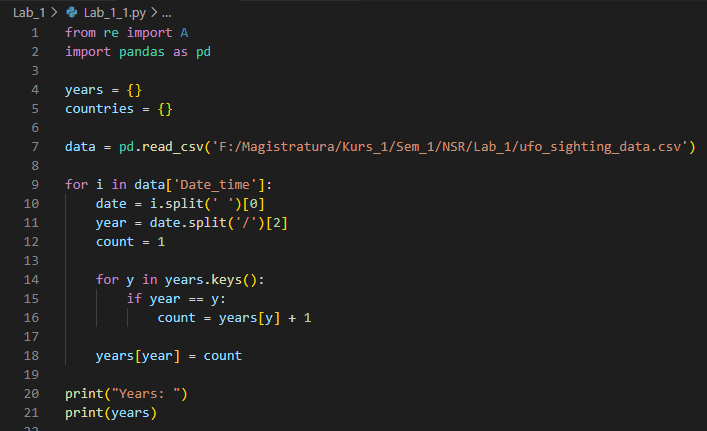


**Вариант 8.1**



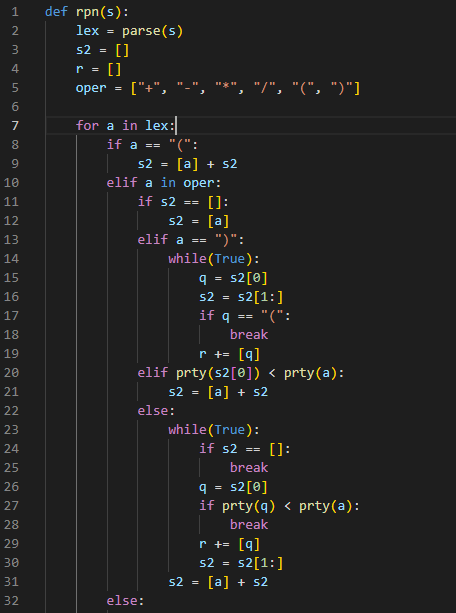


**Задача 8.2**

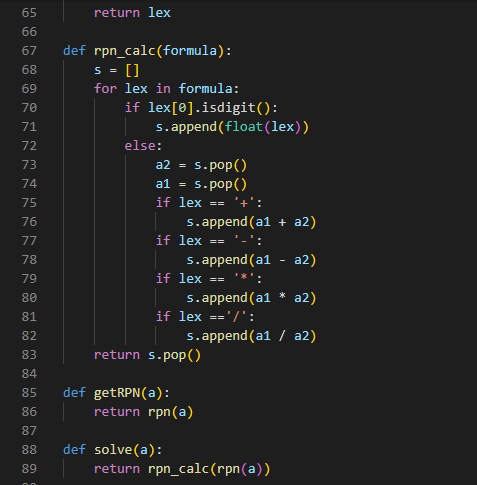




**Калькулятор**





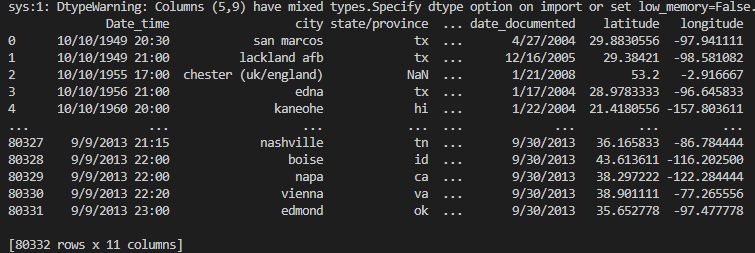




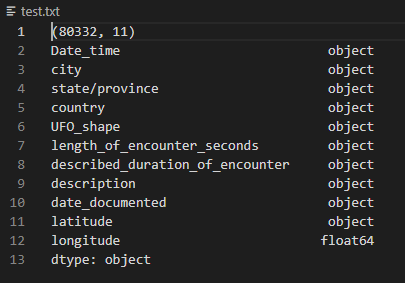
**Тесты**

Считывание из файла данных и отображение их с помощью pandas.

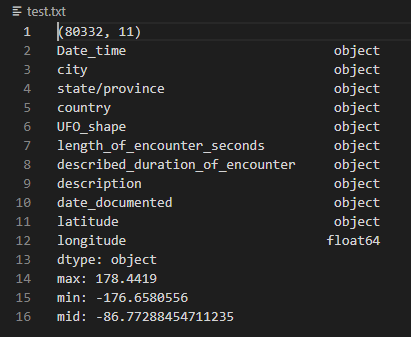




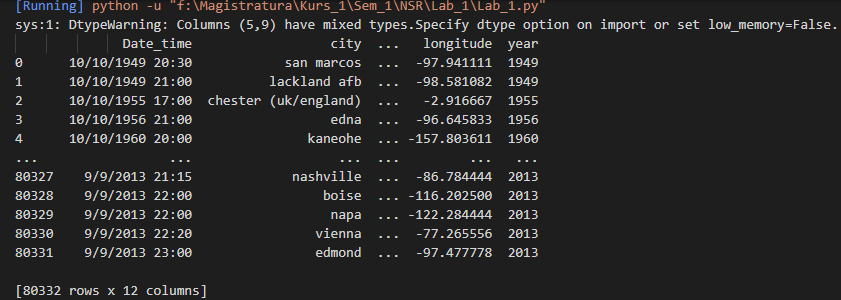
Определение типов данных у столбцов DataFrame и запись их в файл.



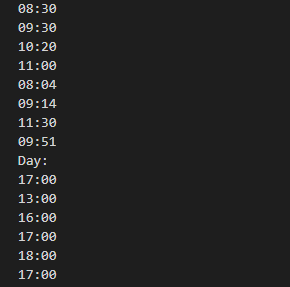
Определение максимального, минимального и среднего значения для столбца.



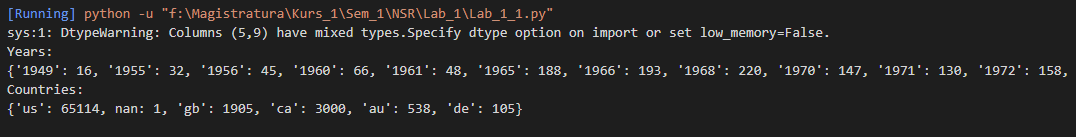
Создание дополнительного столбца в DataFrame.



Получение времени наблюдения НЛО и разделение их по времени дня.

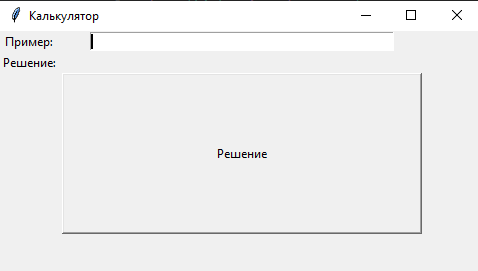


Получение количества сообщений по годам и странам.

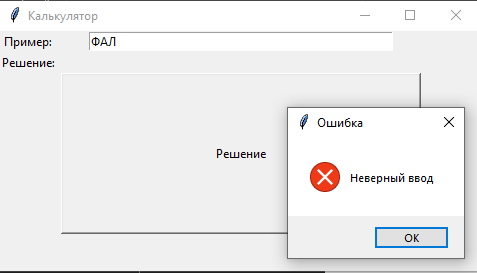


**Калькулятор**

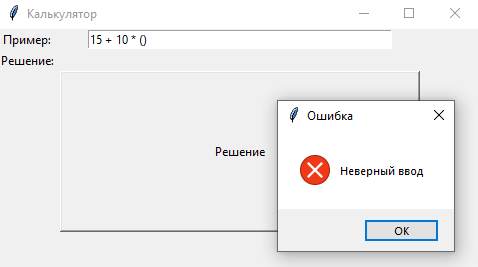
Главная форма



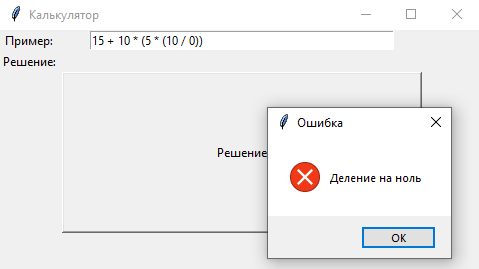
Неверный ввод



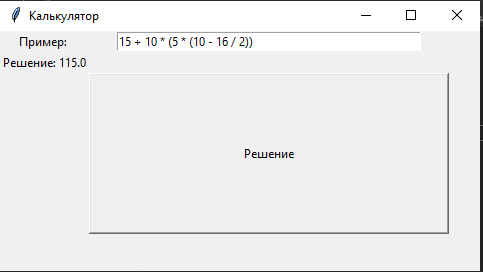
Неверный ввод пустых скобок



Ошибка при делении на ноль



Решение примера



**Вывод**

В данной работе была изучена библиотека pandas. С помощью данной библиотеки была проведена работа с данными.

Также в данной работе был разработан калькулятор с обратной полькой записью. Для калькулятора был реализован интерфейс с помощью tkinter.