# Лабораторная работа 3.3. Работа с массивами данных. Вычислительная сложность алгоритмов

**Цель работы**: научиться понимать природу и разновидности ошибок, возникающих в программном обеспечении, а также исследовать подходы к их поиску, отладке и обработке. Освоить обработку исключений в Python и научиться писать надёжный код с минимизацией ошибок.

**Задачи работы**:

1. Изучить основные известные примеры ошибок в программном обеспечении и их последствия.

2. Разобраться в типах ошибок: синтаксических, логических и ошибок времени выполнения.

3. Освоить методы поиска ошибок и отладки программ.

4. Изучить подходы к обработке ошибок, включая использование конструкции **try-except** в Python.

5. Научиться возбуждать исключения вручную с использованием **raise** и работать с утверждениями (**assert**).

6. Научиться правильно использовать исключения и утверждения в коде, оценить их преимущества и недостатки.

7. Решить практические задачи по обработке ошибок и написанию кода с учётом возможных исключительных ситуаций.

**Примечание**

При выполнении заданий используйте заготовки решений: -> [Репозиторий](https://github.com/BosenkoTM/Python-Programming-A-101/tree/main/labs/lab_03-02).

* <https://github.com/BosenkoTM/Python-Programming-A-101/tree/main/labs/lab_03-03>
* Изучить теоретические основы раздела[Ошибки и исключения](https://www.yuripetrov.ru/edu/python/ch_07.html)

<https://www.yuripetrov.ru/edu/python/ch_07_01.html>

**Цель работы**: научиться работать с файлами, выполнять их чтение, запись и манипуляцию, а также освоить методы сериализации и десериализации данных с использованием популярных форматов хранения, таких как CSV и JSON.

**Задачи работы**:

* Изучить структуру файловой системы, свойства файлов, виды путей к файлам (абсолютный и относительный).
* Освоить операции с файлами в Python: открытие, чтение, запись, закрытие.
* Разобраться с файловыми объектами и их основными свойствами и методами.
* Изучить методы работы с файлами: чтение файла целиком, построчное чтение, запись строк в файл.
* Познакомиться с концепцией сериализации и десериализации данных.
* Изучить популярные форматы сериализации: CSV и JSON.
* Научиться использовать модуль **pickle** для сериализации данных в Python.
* Решить практические задачи по обработке файлов и сериализации данных.

**Теоретические основы**

**3.4.1** Файлы и файловая система

**3.4.1.1** Свойства файла

**Файл** — это именованный блок данных на диске. Основные свойства файла:

- Имя файла (например, data.txt).

- Размер файла (в байтах).

- Тип файла (текстовый, бинарный, изображение и т.д.).

- Права доступа (чтение, запись, выполнение).

**3.4.1.2** Путь к файлу: абсолютный и относительный

- Абсолютный путь: полный путь от корня файловой системы до файла. Например:

C:\Users\username\Documents\data.txt

- Относительный путь: путь относительно текущей директории программы. Например:

./data.txt

**3.4.1.3** Операции с файлами

- Создание: создание нового файла.

- Чтение: получение содержимого файла.

- Запись: добавление данных в файл.

- Удаление: удаление файла из файловой системы.

**3.4.1.4** Виды файлов

Текстовые файлы: хранят данные в читаемом для человека виде (например, .txt, .csv, .json).

Бинарные файлы: хранят данные в закодированном виде (например, .bin, .png, .exe).

**3.4.2** Файловый объект в Python

Файловый объект предоставляет интерфейс для работы с файлами. Его можно получить с помощью функции **open().**

Синтаксис:

file = open('filename.txt', 'mode')

- Режимы работы с файлами:

- 'r' — чтение (read).

- 'w' — запись (write), создаёт файл, если его нет.

- 'a' — добавление (append).

- 'b' — бинарный режим (например, 'rb', 'wb').

Пример:

file = open('example.txt', 'r')

content = file.read()

file.close()

**3.4.3** Работа с файлами в Python

**3.4.3.1** Основные свойства и методы

Методы работы с файлами:

- **read()** — чтение файла целиком.

- **readline()** — чтение одной строки.

- **readlines()** — чтение всех строк в виде списка.

- **write()** — запись строки в файл.

- **writelines()** — запись списка строк.

**3.4.3.2** Простое чтение и запись

Чтение:

with open('example.txt', 'r') as file:

content = file.read()

print(content)

Запись:

with open('example.txt', 'w') as file:

file.write("Hello, world!")

**3.4.3.3** Чтение файла целиком

Для чтения больших файлов можно использовать цикл:

python

with open('large\_file.txt', 'r') as file:

for line in file:

print(line.strip())

**3.4.4** Сериализация и десериализация

**3.4.4.1** Модуль pickle

**pickle** — модуль для сериализации Python-объектов в бинарный формат.

Пример:

import pickle

# Сериализация

data = {'name': 'Alice', 'age': 25}

with open('data.pkl', 'wb') as file:

pickle.dump(data, file)

# Десериализация

with open('data.pkl', 'rb') as file:

loaded\_data = pickle.load(file)

print(loaded\_data)

**3.4.4.2** Популярные форматы файлов

**3.4.4.2.1** CSV

CSV (Comma-Separated Values) используется для хранения таблиц. Работа с ним осуществляется с помощью модуля csv.

Пример:

import csv

# Запись

data = [['Name', 'Age'], ['Alice', 25], ['Bob', 30]]

with open('data.csv', 'w', newline='') as file:

writer = csv.writer(file)

writer.writerows(data)

# Чтение

with open('data.csv', 'r') as file:

reader = csv.reader(file)

for row in reader:

print(row)

**3.4.4.2.2** JSON

**JSON (JavaScript Object Notation)** используется для хранения структурированных данных. Модуль json упрощает работу с этим форматом.

Пример:

import json

# Запись

data = {'name': 'Alice', 'age': 25}

with open('data.json', 'w') as file:

json.dump(data, file)

# Чтение

with open('data.json', 'r') as file:

loaded\_data = json.load(file)

print(loaded\_data)

**Ход работы**

## [Тема Простые текстовые файлы](https://www.yuripetrov.ru/edu/python/ch_08_02.html#id17)

**№ 3.4.1**

С клавиатуры в одной строке вводится произвольное количество вещественных чисел. Запишите их в файл, расположив каждое число на отдельной строке.

**№ 3.4.2**

Дан файл, полученный на выходе [задачи № 3.4.1](https://www.yuripetrov.ru/edu/python/ch_08_02.html#ch-08-02-01):

* загрузите список чисел;
* вычислите их сумму и максимум и допишите их в файл.

Выполнив программу несколько раз, убедитесь, что новые значения учитываются при подсчете.

Если файл прочитать не удается, программа должна прекратить чтение и сообщить об этом пользователю.

**Примечание**

Пример

|  |  |
| --- | --- |
| Исходный файл | Файл после обработки |
| 1  2  3  4  5 | 1  2  3  4  5  15  5 |

**№ 3.4.3**

Измените [задачу № 3.4.2](https://www.yuripetrov.ru/edu/python/ch_08_02.html#ch-08-02-02) так, чтобы вне зависимости от ошибок чтения файла, программа выполняла подсчет суммы и максимума.

**Примечание**

Пример

|  |  |
| --- | --- |
| Исходный файл | Файл после обработки |
| 1  asd  3  -  1.1  5 | 1  asd  3  -  1.1  5  9  5 |

**№ 3.4.4**

В файле записано стихотворение. Выведите его на экран, а также укажите, каких слов в нем больше: начинающихся на гласную или на согласную букву (регистр не учитывается)?

**№ 3.4.5**

Информация о занятости мест в зрительном зале кинотеатра хранится в текстовом файле:

0 0 1 1

0 0 0 1 1

1 1 0 1

где строка обозначает ряд, столбец - место (0 - свободно, 1 - занято).

Напишите программу, которая позволит пользователю увидеть количество свободных мест, а также, введя номер ряда и места, получить информацию - свободно оно или нет.

**№ 3.4.6**

Дан файл с результатами голосования на выборах в виде:

1 3 14 10 2 3 -1 ...

где номер (через пробел) определяет голос за партию из списка:

1. Партия №1.

2. Партия №2.

3. Партия №3.

4. Партия №4.

5. Партия №5.

-1. Испорченный бланк.

Если при чтении файла какое-либо значение не распознается (не является целым числом) или партии с таким номером не существует, считается, что бланк испорчен (-1).

Подведите итоги выборов, выведя на экран список партий в соответствии с убыванием количества полученных голосов и их процентным соотношением:

1. Партия №2 | 1111 | 58.21%

2. Партия №4 | 999 | 38.14%

...

## [Тема Сериализация и десериализация](https://www.yuripetrov.ru/edu/python/ch_08_02.html#id18)

**№ 3.4.7**

На сайте [Всемирного банка](http://worldbank.org/) (WB) в разделе Data доступна экономическая статистика о [валовом внутреннем продукте](https://ru.ruwiki.ru/wiki/%D0%92%D0%B0%D0%BB%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B2%D0%BD%D1%83%D1%82%D1%80%D0%B5%D0%BD%D0%BD%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82) (ВВП) на душу населения в долларах США [1](https://www.yuripetrov.ru/edu/python/ch_08_02.html#id14).

Используя подготовленный CSV-файл, реализуйте:

* загрузку данных;
* поиск государства по названию, а также государства с максимальным, минимальным ВВП на душу населения;
* сохранение данных в новый CSV-файл с фильтром по определенному условию (например, топ-10 государств по объему ВВП на душу населения).

**Совет**

В исходном файле для ряда государств значения ВВП на душу населения не известны - эти государства должны быть пропущены при загрузке

**№ 3.4.8**

[Федеральная служба государственной статистики](http://www.gks.ru/) приводит данные о заболеваемости населения по основным классам болезней [2](https://www.yuripetrov.ru/edu/python/ch_08_02.html#id15).

Подготовленный CSV-файл содержит данные по детям (0-14 лет) и взрослым (15 лет и старше) за период 2000-2015 гг. в отношении на 1000 человек:

Класс,2000,**2001**,2002,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015,Возраст

болезни глаза и его придаточного аппарата,»31,**9»**,»32,4»,»33,5»,33,»34,2»,»33,6»,»35,7»,»34,8»,34,»33,5»,33,»33,3»,»35,2»,35,»34,7»,»33,3»,Взрослые

болезни глаза и его придаточного аппарата,»4,**67»**,»4,81»,»5,32»,»5,13»,»5,43»,»5,58»,»5,5»,»5,68»,»5,7»,»5,68»,»5,81»,»5,87»,»6,12»,»6,03»,6,»5,91»,Дети

…

Предоставив пользователю выбрать 2 различных года для сравнения, сформируйте JSON-файл, который содержит абсолютные значения (для второго года) и изменения по количеству заболеваний в процентах (округление до 2-х знаков после запятой). Значения должны быть сгруппированы по категории Возраст и идти по убыванию.

Фрагмент выходного файла для 2014 и 2015 гг.:

{

"Второй год": {

"Взрослые": [

{

"болезни органов дыхания": 337.9

},

{

"травмы, отравления и некоторые другие последствия воздействия внешних причин": 90.4

},

{

"осложнения беременности, родов и послеродового периода": 73.6

},

...

],

"Дети": [

{

"болезни органов дыхания": 115.76

},

{

"травмы, отравления и некоторые другие последствия воздействия внешних причин": 10.43

},

{

"болезни кожи и подкожной клетчатки": 7.64

},

...

]

},

"Изменения": {

"Взрослые": [

{

"болезни эндокринной системы, расстройства питания и нарушения обмена веществ": 18.75

},

{

"болезни системы кровообращения": 8.33

},

{

"болезни органов дыхания": 1.35

}

...

],

"Дети": [

{

"травмы, отравления и некоторые другие последствия воздействия внешних причин": 0.19

},

{

"новообразования": 0.0

},

{

"болезни органов дыхания": -1.1

}

...

]

}

}

...

**№ 3.4.9**

На официальном сайте [международной федерации хоккея на льду](https://ru.ruwiki.ru/wiki/%D0%9C%D0%B5%D0%B6%D0%B4%D1%83%D0%BD%D0%B0%D1%80%D0%BE%D0%B4%D0%BD%D0%B0%D1%8F_%D1%84%D0%B5%D0%B4%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F_%D1%85%D0%BE%D0%BA%D0%BA%D0%B5%D1%8F_%D0%BD%D0%B0_%D0%BB%D1%8C%D0%B4%D1%83) доступна [информация об участниках прошедших чемпионатов мира (ЧМ)](http://www.iihf.com/iihf-home/history/past-tournaments/).

Используя подготовленный JSON-файл с данными игроков сборной России на ЧМ 2016 г. [3](https://www.yuripetrov.ru/edu/python/ch_08_02.html#id16):

[

{

"номер":13,

"фамилия":"Дацюк",

"имя":"Павел",

"амплуа":"Нападающий",

"рука":"Левая",

"дата\_рождения":"1978-07-20",

"клуб":"Детройт Ред Уингз",

"клуб\_страна":"США",

"возраст":37.782340862423,

"рост":180,

"вес":88

},

...

реализуйте:

* загрузку данных;
* сохранение в CSV-файл группировки по выбранному параметру в формате (на примере имени):
* "Имя","Количество"
* "Сергей","5"
* "Александр","2"
* "Алексей","2"
* ...

**№ 3.4.10**

Дан телефонный справочник в формате JSON:

[

{

"имя":"...",

"телефоны":[

{

"описание":"...",

"номер":"..."

},

{

"описание":"...",

"номер":"..."

}

]

},

...

Программа должна позволять (предоставлять функции):

* загружать информацию из справочника;
* выполнять поиск контактов по номеру телефона;
* выполнять поиск контактов по имени;
* добавлять контакт;
* удалять контакты по имени;
* удалять номер телефона из контактов;
* сохранять справочник в файл.

[[1](https://www.yuripetrov.ru/edu/python/ch_08_02.html#id8)] GDP per capita (current US$) | Data. URL: <https://data.worldbank.org/indicator/NY.GDP.PCAP.CD>.

[[2](https://www.yuripetrov.ru/edu/python/ch_08_02.html#id10)] Здравоохранение::Федеральная служба государственной статистики. URL: <http://www.gks.ru/wps/wcm/connect/rosstat_main/rosstat/ru/statistics/population/healthcare/>

[[3](https://www.yuripetrov.ru/edu/python/ch_08_02.html#id13)] Ice hockey players at IIHF world championships, 2001-2016. The dataset contains 6292 records of 3333 players“ height and weight parameters at 16 world championships. URL: <https://figshare.com/articles/Height_of_ice_hockey_players/3394735/2>