

Анализ и обработка данных с использованием структуры данных Словарь “Фильмотека”

Шувалова В. Д.
студентка группы ПМбд-31

Суть программы “Фильмотека”

Данная программа реализует:

- хранение данных о фильмах
- ввод запросов от пользователя через консоль
- обработку запросов и анализ данных
- вывод ответов в консоль

Готовый продукт - исполняемый файл **Cinematheque.exe**

Либо можно запускать файл с кодом **main.py** из папки проекта **CinemathequeProject**. В ней также лежит файл с данными **data.json**.

Работа с программой с
точки зрения пользователя

Главное меню

Начало работы программы "Фильмотека"

Выберите пункт:

1. Вывести фильмы, удовлетворяющие заданным(-ому) условиям(-ю)
2. Вывести фильмы, отсортированные по заданному критерию
3. Вывести информацию по фильму
4. Вывести список всех фильмов в "Фильмотеке"
5. Завершить программу

Введите номер пункта: *a*

Вы ввели что-то, отличное от целого числа. Повторите ввод.

Введите номер пункта: *10*

Вы ввели целое число, отличное от 1, 2, 3, 4 или 5. Повторите ввод.

Введите номер пункта:

Вывод всех фильмов, продолжение работы программы

Введите номер пункта: 4

1. Бегущий по лезвию 2049

2. Бедные-несчастные

3. Виды доброты

4. Высшее общество

5. Круэлла

6. Ла-Ла Ленд

7. Маяк

8. Микки 17

9. Солнцестояние

10. Субстанция

Желаете продолжить? +/- :

Желаете продолжить? +/- : -

Завершение работы программы "Фильмотека"

Желаете продолжить? +/- : да

Завершение работы программы "Фильмотека"

Желаете продолжить? +/- : +

Выберите пункт:

1. Вывести фильмы, удовлетворяющ

2. Вывести фильмы, отсортированн

3. Вывести информацию по фильму

Вывод информации о фильме

Введите название фильма: *Ла-Ла Ленд*

Ла-Ла Ленд:

год производства - 2016

страны производства - США, Гонконг

длительность - 2:08

режиссер - Дэмьен Шазелл

жанры - мюзикл, драма, мелодрама, комедия

рейтинг IMDb - 8.0

бюджет в \$ - 300000000

актеры - Райан Гослинг, Эмма Стоун, Джон Ледженд, Дж.К. Симмонс

Желаете продолжить? +/-:

Введите название фильма: *ла-ла ленд*

Ла-Ла Ленд:

год производства - 2016

Введите название фильма: *Интерстеллар*

Такого фильма нет в "Фильмотеке"

Желаете продолжить? +/-:

Сортировка фильмов по критерию

Введите номер пункта: 2

Выберите параметр, по которому буд

1. год производства
2. длительность
3. рейтинг IMDb
4. бюджет в \$

Введите номер параметра: 3

Выберите критерий сортировки:

1. по возрастанию
2. по убыванию

Выберите номер критерия: 2



1. Ла-Ла Ленд - 8.0
 2. Бегущий по лезвию 2049 - 8.0
 3. Бедные-несчастные - 7.8
 4. Маяк - 7.4
 5. Субстанция - 7.3
 6. Круэлла - 7.3
 7. Микки 17 - 7.1
 8. Солнцестояние - 7.1
 9. Виды доброты - 6.5
 10. Высшее общество - 5.7
- Желаете продолжить? +/-:

Пример простого запроса фильмов по одному условию

Введите номер пункта: **1**

Выберите параметр, для которог

1. год производства
2. страны производства
3. длительность
4. режиссер
5. жанры
6. рейтинг IMDb
7. бюджет в \$
8. актеры

Введите номер параметра: **8**

По данному параметру будут ото

Введите имя актера: **Роберт Паттинсон**

Хотите ввести ещё одно условие? +/-: -

1. Высшее общество
2. Маяк
3. Микки 17

Желаете продолжить? +/-:

Запрос фильмов по нескольким условиям

1

Введите номер пункта: 1

Выберите параметр, для которого будет задано условие:

1. год производства
2. страны производства
3. длительность
4. режиссер
5. жанры
6. рейтинг IMDb
7. бюджет в \$
8. актеры

Введите номер параметра: 1

Запрос фильмов по нескольким условиям

2

После того, как вы введёте значение

Введите год производства: 2020

3

Введите отношение между параметром и значением: =/>/</>=/<= (без символа слеша): >=

Хотите ввести ещё одно условие? +/-: +

Запрос фильмов по нескольким условиям

4

Хотите ввести ещё одно условие? +/- : +

Выберите параметр, для которого будет задано условие:

1. год производства
2. страны производства
3. длительность
4. режиссер
5. жанры
6. рейтинг IMDb
7. бюджет в \$
8. актеры

Введите номер параметра: 1

Запрос фильмов по нескольким условиям

5

После того, как вы введёте значение
Введите год производства: 2024

6

Введите отношение между параметром и значением: =/>/</>=/<= (без символа слеша): <
Хотите ввести ещё одно условие? +/-: -

Запрос фильмов по нескольким условиям

7

Вы ввели несколько условий. Введите отношение между ними:

И - найти фильмы, удовлетворяющие всем условиям

ИЛИ - найти фильмы, удовлетворяющие хотя бы одному условию

Отношение И/ИЛИ: И

1. Бедные-несчастные

2. Круэлла

Желаете продолжить? +/-:

Отношение И/ИЛИ: И

1. Бедные-несчастные

2. Круэлла

Желаете продолжить? +/-:

Отношение И/ИЛИ: and

Некорректный ввод. Повторите ввод.

Отношение И/ИЛИ:

Структура кода

Хранение данных

data.json x

```
1  {
2    "Субстанция": {
3      "год производства": 2024,
4      "страны производства": ["Великобритания", "Франция"],
5      "длительность": "2:21",
6      "режиссер": "Корали Фаржа",
7      "жанры": ["ужасы", "драма"],
8      "рейтинг IMDb": 7.3,
9      "бюджет в $": 17500000,
10     "актеры": ["Деми Мур", "Маргарет Куэлли", "Деннис Куэйд"]
11   },
12   "Виды доброты": {
13     "год производства": 2024,
```

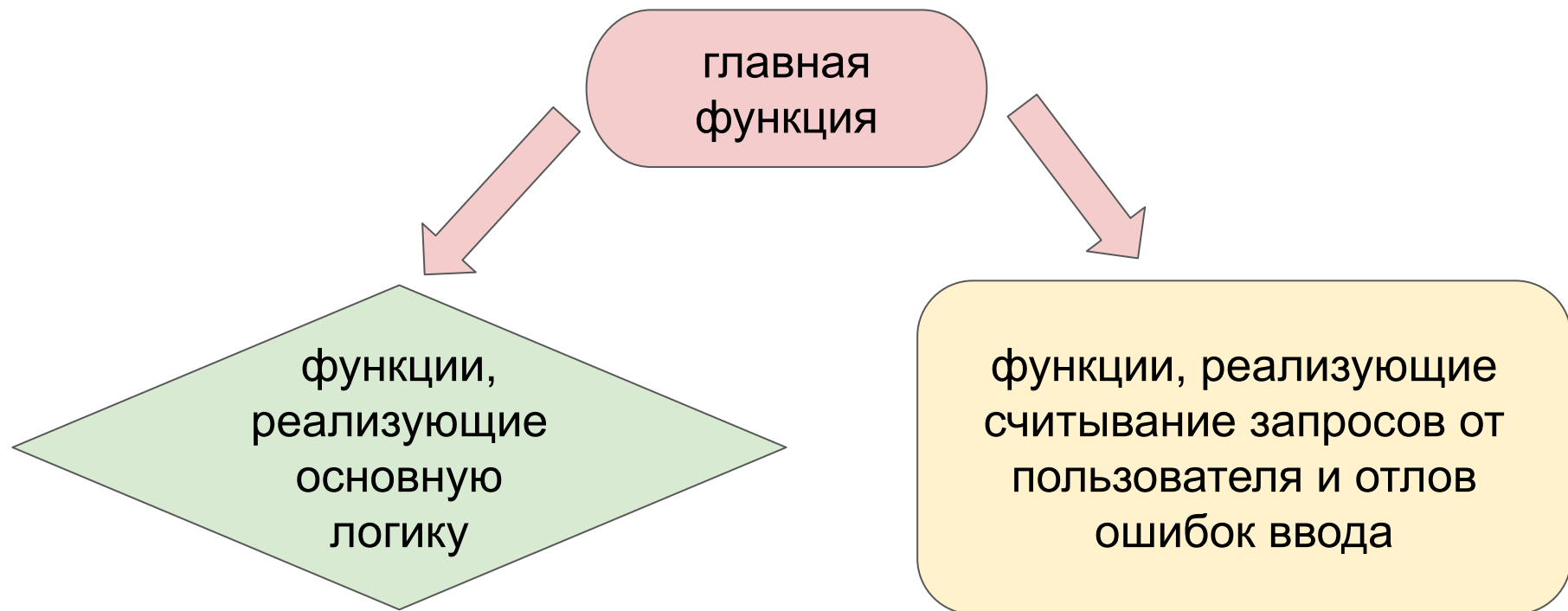
Считывание данных и структура словаря

main.py ×

```
1 import json
2
3 with open(file="data.json", mode="r", encoding="utf-8") as file_in:
4     films_dictionary = json.load(file_in)
```

Благодаря формату файла JSON **структура словаря `films_dictionary`** идентична структуре, хранящейся в `data.json`.

Функции программы можно разделить на 3 группы



Главная функция - main()

```
247 def main():
248     print("Начало работы программы \"Фильмотека\"")
249 > while True: ...
293     print("Завершение работы программы \"Фильмотека\"")
294
295
296 main()
```

Она является **точкой входа в программу**: вызывает все остальные функции.

Функции, реализующие основную логику:

request_films_complying_one_condition()

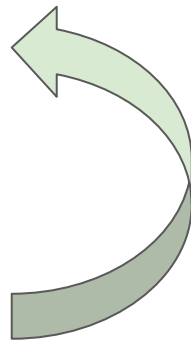
запрашивает множество фильмов, удовлетворяющих одному условию

request_films_complying_many_conditions()

запрашивает множество фильмов, удовлетворяющих многим условиям

request_sort_films_by_criteria()

запрашивает список фильмов и значений их параметров, отсортированных по заданному критерию



Функция `request_films_complying_one_condition()`

```
def request_films_complying_one_condition(par, val, rel):
```

```
    """
```

```
    Запрашивает фильмы, удовлетворяющие одному условию
```

```
    :param par: параметр ("год производства"/"страны производства"/.../"актеры")
```

```
    :param val: значение параметра
```

```
    :param rel: отношение между параметром и значением: "="/">"/"<"/">="/"<="
```

```
    :return: мн-во фильмов films
```

```
    Пример запроса №1: par="год производства", val=2020, rel=">" - фильмы после 2020
```

```
    Пример запроса №2: par="актеры", val="Роберт Паттинсон", rel=None - фильмы, в ко
```

```
    """
```

Функция `request_films_complying_many_conditions()`

```
def request_films_complying_many_conditions(conds, cond_rel):  
    """  
    Запрашивает фильмы, удовлетворяющие многим условиям  
    :param conds: список условий [(par1, val1, rel1), (par2, val2, rel2), ...]  
    :param cond_rel: отношение между условиями: "И" - должны выполняться все условия,  
    :return: мн-во фильмов films  
    """  
  
    if cond_rel == "И":  
        func = lambda x, y: x.intersection(y)  
    else:  
        func = lambda x, y: x.union(y)
```

Функция request_sort_films_by_criteria()

```
def request_sort_films_by_criteria(par, criteria):  
    """  
    Запрашивает список фильмов и значения их параметров, отсортированных по заданному критерию  
    :param par: параметр ("год производства"/"длительность"/"рейтинг IMDb"/"бюджет в $")  
    :param criteria: критерий: "increase" - возрастание, "decrease" - убывание  
    :return: sorted_films_data  
    """  
  
    films_data = [(film_name, film_info[par]) for film_name, film_info in films_dictionary.items()]  
  
    sorted_films_data = sorted(films_data, key=cmp, reverse=reverse_par)  
    return sorted_films_data
```

Функции, реализующие считывание запросов от пользователя и отлов ошибок ввода:

get_main_option()

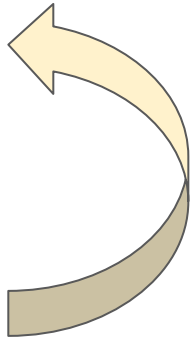
запрашивает пункт главного меню

get_one_condition()

запрашивает информацию для отбора фильмов по одному условию

get_many_conditions()

запрашивает информацию для отбора фильмов по нескольким условиям



Функции, реализующие считывание запросов от пользователя и отлов ошибок ввода:

get_condition_relation()

запрашивает отношение между условиями (И/ИЛИ)

get_sorting_criteria()

запрашивает информацию для сортировки фильмов по критерию