

АВС. Домашнее задание 3

Выполнил Завалинский Даниил, БПИ204.

Объекты – фильмы, сортировка включением по убыванию. Вариант 133 (7, 10)

Запуск программы:

```
python main.py <input_file_name> <output_file_name>
```

Для генерации случайного теста:

```
python main.py random <number_of_movies> <input_file_name> <output_file_name>
```

Структура проекта:

Файл Movie.py – класс для абстрактного класса-фильма

Documentary.py, AnimationFilm.py, FictionFilm.py – классы для вариаций фильма

main.py – содержит описания функций чтения объектов из файла в контейнер, записи из контейнера в файл, функцию сортировки объектов в контейнере и функцию main.

Файлы input{i}.txt – содержат наборы тестовых данных, output.txt – соответствующие результаты выполнения программы.

Организация структуры программы:

	поля		функции		
	название	тип	название	аргументы	что делает
	name	string	__init__	self, input_string - string	Конструктор
	year	int	function	self	Возвращает признак общий для всех альтернатив
Movie Базовый класс для фильмов			to_string	self	Абстрактная ф- я для строкового представления объекта

	поля		функции		
	название	тип	название	аргументы	что делает
	name	string	__init__	self, input_string - string	Конструктор
	year	int	to_string	self	Переопределенная ф-я для строкового представления объекта
FictionFilm Класс для описания игрового фильма	director	String			

	поля		функции		
	название	тип	название	аргументы	что делает
	name	string	__init__	self, input_string - string	Конструктор
	year	int	to_string	self	Переопределенная ф-я для строкового представления объекта
Documentary Класс для описания документального фильма	length	int			

	поля		функции		
	название	тип	название	аргументы	что делает
	name	string	__init__	self, input_string - string	Конструктор
	year	int	to_string	self	Переопределенная ф-я для строкового представления объекта
AnimationFilm Класс для описания мультфильма *также содержит класс- перечисление типов анимации	animation_type	int			

Данные классы содержатся в пакете movies, который весит 2,76 кб

	функции		
	название	аргументы	что делает
	generate_random_test	ostream	Записывает в файл сгенерированные случайным образом объекты
	read_input	input_string	считывает объекты из файла в контейнер, возвращает контейнер
	write_result	output_stream, container	записывает содержимое контейнера в файл
Другие функции (в utils.py)	insertion_sort	container	сортирует объекты нужным образом

- Пакет utils весом 2,33 кб

Организация памяти блока main:

1,32 кб

переменная	тип	описание
inputFileName outputFileName	string	имя файлов с входными и выходными данными
input_file output_file	_IO.TextIOWrapper	файлы с входными данными и для записи полученного контейнера
file_content	string	содержимое файла
cont	list(Movie)	контейнер

Общее количество модулей – 3. Суммарный вес модулей – 6,41 кб.

Время исполнения программы:

кол-во объектов	1-10	100	1000	10000
подход с динамической типизацией	менее 1 мс	0,015625с	0,1875с	16,8с
Функциональный подход	0,002с	~0,01с	0,028с	1,24с
ООП подход	менее 0,002с	менее 0,01 с	0,02с	0,95с

В сравнении с ООП и функциональным подходом, стиль с динамической типизацией несколько более медленный, поскольку значительная часть времени уходит на проверку типов.