Маринченко Игорь БПИ207

**Домашнее задание №3**

Использование динамически типизированного универсального языка программирования.

1. **Описание задания:**

|  |  |  |  |
| --- | --- | --- | --- |
| 7. Фильмы | 1. Игровой (режиссер - строка символов)  2. Мультфильм (способ создания – перечислимый тип = рисованный, кукольный, пластилиновый…)  3. Документальный фильм (длительность в минутах – целое) | 1. Название фильма –строка символов.  2. Год выхода -целое | Частное от деления года выхода фильма на количество символов в названии (действительное число) |

20. Удалить из контейнера те элементы, для которых значение, полученное с использованием функции, общей для всех альтернатив, больше чем среднее арифметическое для всех элементов контейнера, полученное с использованием этой же функции

1. **Структурная схема программы:**

|  |
| --- |
| Типы |
| Class Film  \_\_film\_type (хранит в себе тип фильма в виде строки)  \_\_title (хранит в себе название фильма в виде строки)  \_\_year (хранит в себе год выпуска фильма в виде строки)  \_\_special\_argument (хранит в себе кортеж из двух элементов:  1) название переменной преимущественной для заданного типа фильма в виде строки  2) её значение в виде строки) |
| Class Container  \_\_films\_list (список фильмов)  \_\_function\_average (среднее значение функции для элементов списка) |

1. **Основные характеристики программы**

Размер исходных текстов – 5кб.

Число модулей реализации – 4.

Размер исполняемого файла:

1) Основной файл программы main.py 2кб

2) Сформированный main.exe исполняемый файл 6582кб

**Формат входных данных:**

В начале идет число (1,2 или 3), которое определяет тип фильма, затем:

1. Художественный фильм:

* Во второй строчке идет название фильма.
* В третьей год выпуска.
* В четвертой имя режиссёра.

1. Мультипликационный фильм:

* Во второй строчке идет название фильма.
* В третьей год выпуска.
* В четвёртой тип анимации:

1: рисованный

2: кукольный

3: пластилиновый

1. Документальный фильм:

* Во второй строчке идет название фильма.
* В третьей год выпуска.
* В четвертой длина фильма.

Файлы test1-test5 содержат примеры вводных данных.

Файлы outfile[n]\_1 содержат выходные данные после формирования контейнера, а outfile[n]\_2 данные после удаления лишних элементов.

|  |  |  |
| --- | --- | --- |
| Номер тестового набора | Количество элементов | Время работы, миллисекунды |
| 1 | 3 | 0,001053 |
| 2 | 5 | 0,001396 |
| 3 | 7 | 0,001848 |
| 6 | 2000 | 0.040425 |
| 7 | 10000 | 0.294857 |

**4. Вывод**

Динамическая типизация языка позволила сократить число необходимых классов фильмов до одного, реализовав отличительную черту фильма каждого типа в отдельной переменной. Код стал намного более читабельным и ёмким. Та же программа написанная на C++ потребовала более чем в 2 раза большего количества строк кода. Также нельзя не заметить, что время работы практически не изменилось, возможно при работе с другими объемами данных или со сложными вычислениями разница бы стала критичной, но в случае с маленькими программами она минимальна.