Задача: Фильмы, Вариант 147 (7, 11)
Автор: Тасбауова Даяна

ВТОРОЕ ДОМАШНЕЕ ЗАДАНИЕ ПО ПРЕДМЕТУ "АРХИТЕКТУРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ"

УСЛОВИЕ ЗАДАЧИ

Общие для всех альтернатив переменные

- Название фильма – строка символов.
- Год выхода – целое

Общие для всех альтернатив функции

- Частное от деления года создания на количество символов в названии (действительное число)

Функционал

- После размещения данных в контейнер необходимо осуществить их обработку в соответствии с вариантом задания.
  Обработанные данные после этого заносятся в отдельный файл результатов.
- Упорядочить элементы контейнера по возрастанию используя сортировку Сортировка с помощью прямого выбора (Straight Selection). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив.

Базовые альтернативы

1. Игровой
   - наличие, строки символов – режиссёр
2. Мультфильм
   - способ создания – перечислимый тип = рисованный, кукольный, пластилиновый
3. Научный
   - длительность в минутах – целое

ВРЕМЯ РАБОТЫ ПРОГРАММЫ НА РАЗНЫХ РАЗМЕРАХ ВХОДНЫХ ДАННЫХ:

КОЛИЧЕСТВО ФИЛЬМОВ	ВРЕМЯ РАБОТЫ, SECONDS	ПОТРЕБЛЯЕМАЯ ПАМЯТЬ, КВ
7	< 0.002	~2750
100	< 0.01	~2934
1000	0.02	~3450
5000	0.16	~4226
10000	0.95	~4564

РАЗНИЦА МЕЖДУ ПРОЦЕДУРНОЙ И ОБЪЕКТНО-ОРИЕНТИРОВАННОЙ РЕАЛИЗАЦИЕЙ

У процедурной реализации код потребляет в 1,21 раза больше памяти и в 1,27 раза больше времени выполнения. Следовательно, ООП подход лучше по времени и памяти чем процедурный.

МЕТРИКИ, ОПРЕДЕЛЯЮЩИЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ:

МЕТРИКА	ЗНАЧЕНИЕ
Число интерфейсных модулей	6

Число модулей с реализацией	5 + 1 (методы рандома)
Общий размер исходных текстов программы	14.451 KB
Размер исполняемого файла [GCC, Linux] *	31.352 KB

*Версии подробнее:

```
$ g++ --version
g++ (Ubuntu 9.3.0-17ubuntu1~20.04) 9.3.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

$ uname -a
Linux Dayana-T9241NG2 5.10.16.3-microsoft-standard-WSL2 #1 SMP Tue June 1 12:45:20 UTC 2021 x86_64
x86_64 x86_64 GNU/Linux

$ lsb_release -a
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.2 LTS
Release:        20.04
Codename:       focal
```

БАЗОВЫЕ КЛАССЫ

ТАБЛИЦА КЛАССОВ БАЗОВЫЕ КЛАССЫ	CONTAINER [80004 БАЙТ]
Поля	Static: Отсутствует Local: int length [4 байта] Movie *container[80000 байт]
Методы	Static: Отсутствует Local: void In(FILE *file); void InRandom(int size); void Out(FILE *file); void Sort();
Enum	-
Базовые Классы	Movies [0 байт]
Поля	Static: Отсутствует Local: Отсутствует
Методы	Static: Movie *StaticIn(FILE *file); Movie *StaticInRnd(); Local: ~Movies(); void In(FILE *file); void InRnd(); void Out(FILE *file); double Quotinent();
Enum	-

ПРОИЗВОДНЫЕ КЛАССЫ

ТАБЛИЦА КЛАССОВ ПРОИЗВОДНЫЕ КЛАССЫ	CARTOON : MOVIES [17 БАЙТ + 0 БАЙТ]
Поля	Static: - Local: const char *name [5 байт]; int year [4 байт]; type_ [4 байт];
Методы	Static: - Local: ~Cartoon(); void In(FILE *file); void InRnd(); void Out(FILE *file); double Quotinent();
Enum	Enum MAX_LENGTH -> 4 байта
Производный Класс	Documentary : Movie [13 байт + 0 байт]

Поля	Static: - Local: const char *name [5 байт]; int year [4 байт]; duration [4 байт];
Методы	Static: - Local: ~Documentary(); void In(FILE *file); void InRnd(); void Out(FILE *file); double Quotinent();
Enum	-
Производный Класс	Fiction : Movie [14 байт + 0 байт]
Поля	Static: - Local: const char *name [5 байт]; int year [4 байт]; const char *director [5 байт];
Методы	Static: - Local: ~Fiction(); void In(FILE *file); void InRnd(); void Out(FILE *file); double Quotinent();
Enum	-

ГЛОБАЛЬНАЯ ПАМЯТЬ

ГЛОБАЛЬНАЯ ПАМЯТЬ	БАЗОВОГО КЛАССА CONTAINER	ПРОИЗВОДНОГО КЛАССА MOVIE
Поля	Все статические поля	Все статические поля
Локальные методы	void In(Container *self, FILE *file); void InRnd(Container *self, int size); void Out(Container *self, FILE *file); void Sort(Container *self);	~Movie(); void In(FILE *file); void InRandom(); void Out(FILE *file);

ГЛОБАЛЬНАЯ ПАМЯТЬ(ГП)	БАЗОВОГО КЛАССА(БК)	ПРОИЗВОДНЫХ КЛАССОВ (ПК) CARTOON	FICTIONAL	DOCUMENTARY
Таблица Виртуальных методов (ТВМ)	~Movie(Movie *self); void In(FILE *file); void InRnd(Movie *self); void Out(Movie *self, FILE *file);	~InheritedClassed(Movie *self); void In(Movie *self, FILE *file); void InRandom(Movie *self); void Out(Movie *self, FILE *file);		

ПРОГРАММНАЯ ПАМЯТЬ

ПРОГРАММНАЯ ПАМЯТЬ (ПП)	CONTAINER	MOVIE	ПРОИЗВОДНЫЕ КЛАССЫ CARTOON, FICTIONAL, DOCUMENTARY
Статические	-	char *name [4 байт] int year [4 байт] char, file : Languages *StaticIn(FILE *file) -> 13 байт self : Languages *StaticInRnd() -> 16 байт	
Локальные	file : void In(FILE *file) -> 8 байт size : void InRandom(int size) -> 4 байта file : void Out(FILE *file) -> 8 байт self : void Sort() -> 8 байт	-	-
Виртуальные	-	self : ~Movie() -> 8 байт file : void In(FILE *file) -> 8 байт self : void InRnd() -> 8 байт file : void Out(FILE *file) -> 8 байт double, self : double Quotient() -> 16 байт	self : ~Movie() -> 8 байт file : void In(FILE *file) -> 24 или 25 байт self : void InRnd() -> 24 или 25 байт file : void Out(FILE *file) -> 24 или 25 байт + Quotient() размер double, name, year ,self : double Quotient() -> 24 байт

КУЧА

КУЧА
Container *container -> 80000 байт
Объекты ПК в зависимости от количества (от 0 до 10000)
ПК1 : Cartoon || Fictional || Documentary
ПК2 : Cartoon || Fictional || Documentary
.....
ПК(n - 1) : Cartoon || Fictional || Documentary
ПКn : Cartoon || Fictional || Documentary

СБОРКА ПОЛУЧЕННОЙ ПРОГРАММЫ.

```
cmake -B <build_dir> -DCMAKE_BUILD_TYPE=Release
cmake --build <build_dir> --target main
```

Исполняемый файл программы будет расположен в папке `bin`.

ТЕСТИРОВАНИЕ

Исходные данные для тестирования содержатся в каталоге `tests`.

Файл с результатами прогонов тестов `tests/test_result.txt`.

```
# Parse input file
./bin/main tests/test1.txt tests/test1.out tests/test1_sorted.out.txt
# Parse input random
./bin/main -n 10 tests/data/random.out.txt tests/random_sorted.out.txt
```