

Отчет

Отчет по домашнему заданию №2.

Архитектура вычислительных систем.

Номер варианта: 161

Номер задания: 7 (Фильмы)

Номер функции: 12 (Bubble-sort)

Выполнил Ахундов Алексей Назимович, БПИ205.

Задание:

Основываясь на данных таблице по номеру задания и номеру функции, разработать программу в объектно-ориентированном стиле, поддерживающую ввод, вывод, складирование и сортировку внутри контейнера объектов (фильмов), имеющих альтернативы (игровое кино, мультфильм, документальное кино)

Сортировка осуществляется по ключу, определяемому функцией на основе общих для всех альтернатив полей (год выхода, деленный на количество символов в названии фильма)

Подход к решению:

Создается общий класс фильма, в нем декларируются общие поля – год выхода фильма и его название. От общего класса наследуются альтернативы: игровой фильм, мультфильм и документальное кино, в которых инициализируются специфичные альтернативам поля, а также определяются отдельные методы: вывода альтернативы и заполнения как из файла, так и стохастическими данными.

Данные попадают в контейнер, где контролируется их время жизни (удаление после вывода), а также производится сортировка по ключу, объявленному в главном классе фильма.

Архитектура

Архитектура описана в файле, располагающемся по относительному пути от корня решения: /docs/Архитектура.pdf

Основные характеристики программы

Число интерфейсных модулей: 7
Число модулей реализации: 7 (+main.cpp – драйвер)
Общий размер исходных текстов: 48,0 КБ
Полученный размер исполняемого кода: 72,0 КБ (-O0)

Время выполнения программы:
(Intel Core i7-8550U @ 1.8GHz – 8CPUs)

Детерминированные тесты:

Время исполнения		
Тест	Стиль	
	Процедурный	Объектно-ориентированный
1	0s	0s
2	0s	0s
incorrect_0	0s	0s
incorrect_1	0s	0s
incorrect_2	0s	0s

Тесты стохастическими данными:

Время исполнения		
Тест	Стиль	
	Процедурный	Объектно-ориентированный
1 (1 000)	0.0156s	0.0156s
2 (5 000)	0.4375s	0.3437s
3 (10 000)	1.8125s	1.4218s

Сравнительный анализ

Программа, написанная в объектно-ориентированном стиле, показала лучшие результаты по времени относительно процедурного стиля из-за отсутствия затрат для определения конкретной структуры альтернативы по информации о типе общей структуры, при этом увеличился расход памяти в связи с тем, что приходится поддерживать виртуальную таблицу функций, чтобы вызывать свойственные конкретной альтернативе методы (ввод, вывод, заполнение стохастическими данными).