

Пояснительная записка ко второму домашнему заданию по ABC

Подюков Иван Владимирович, БПИ207, вариант 259

Номер задачи – 7, номер дополнительной функции – 19

Задача:

Обобщенный артефакт, используемый в задании	Базовые альтернативы (уникальные параметры, задающие отличительные признаки альтернатив)	Общие для всех альтернатив переменные	Общие для всех альтернатив функции
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)

Дополнительная функция:

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

Структура программы:

int	4
double	8
char	1

class Fiction producer: std::string	24 24[0]
class Cartoon type_of_cartoon: int	4 4[0]
class Science length_of_movie_in_minutes: int	4 4[0]
class Movie year: int name: std::string	28 4[0] 24[4]
class Container len: int *cont: Movie[10000]	280004 4[0] 280000[4]

main(..)	
argc: int	4[0]
argv: char*	8[4]
c: Container	280004[12]
ifst: std::ifstream	4[280016]
size: int	4[280020]
ofst: std::ofstream	4[280024]

Stack
c.In
c.Out
c.DeleteElementsWithQuotientLessThanAverage
c.Out

Heap
argv[0]
argv[1]="f"
argv[2] = "test1.txt"
argv[3] = "answer1.txt"

Общая информация о программе:

Число интерфейсных модулей (заголовочных файлов) – 6

Число модулей реализации – 6

Общий размер исходных текстов – 15,4 КБ (в прошлом задании было 15,9 КБ), 496 строчек кода.

Размер исполняемого кода – 70,4 КБ (в прошлом задании было 57,1 КБ)

Тесты	Время выполнения программы для различных тестовых наборов данных
test1.txt	0.0010345
test2.txt	0.0011554
test3.txt	0.0010337
test4.txt	0.0020002
test5.txt	0.0034175

Сравнение объектно-ориентированного и процедурного подходов на основе приведённых данных:

Размер исполняемого кода больше при объектно-ориентированном подходе, а общий размер исходных текстов наоборот, меньше. На таких же тестах программа при объектно-ориентированном подходе работает быстрее, чем при процедурном примерно в 1,5 раза. Но при генерации 2000 объектов данная программа тратит 0.0409959 секунд, а предыдущая около 0.018 секунд. Памяти тоже тратиться больше. В целом можно сказать, что использование объектно-ориентированного подхода гораздо более эффективно и удобно, чем использование процедурного.