

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

Архитектура вычислительных систем.
Пояснительная записка.
Практическое задание 4
Вариант 183 (Задача 1. Функция 14).

Исполнитель
Студент БПИ205
Никитин Никита Евгеньевич

Описание полученного задания

Разработать программу, состоящую из контейнера, способного хранить базовые альтернативы обобщенного артефакта.

Обобщенный артефакт — Плоская геометрическая фигура, размещаемая в координатной сетке.

Базовые альтернативы:

1. Круг (целочисленные координата центра окружности, радиус)
2. Прямоугольник (целочисленные координаты левого верхнего и правого нижнего углов)
3. Треугольник (целочисленные координаты трех углов)

Общая переменная для всех альтернатив - Цвет фигуры (перечислимый тип) = {красный, оранжевый, желтый, зеленый, голубой, синий, фиолетовый}

Общая для всех альтернатив функция: Вычисление площади фигуры (действительное число)

Контейнер должен уметь заполняться из файла и случайно (во входном файле нужно указать кол-во элементов для случайной генерации), выводить содержимое в файл(в удобном для пользовательского восприятия пользователем), сортироваться по общей для всех альтернатив функции методом Сортировки Шелла.

Порядок действий для запуска программы описан в `readme.md`

Общие характеристики программы

Число модулей реализации — 6

Общий размер исходных кодов — 1740

Время работы программы на тестовых данных

input1.txt — 3 объекта

real 0m0,003s

user 0m0,000s

sys 0m0,002s

input2.txt — 6 объектов

real 0m0,06s

user 0m0,00s

sys 0m0,003s

input3.txt — 10 объектов

real 0m0,003s

user 0m0,002s

sys 0m0,000s

input4.txt — 14 объектов

real 0m0,003s

user 0m0,002s

sys 0m0,000s

input5.txt — 20 объектов

real 0m0,003s

user 0m0,002s

sys 0m0,000s

Время работы при случайной генерации 1000 элементов

real 0m0,013s

user 0m0,006s

sys 0m0,003s

Время работы при случайной генерации 10000 элементов

real 0m0,036s

user 0m0,025s

sys 0m0,014s

Сравнение с предыдущим методом реализации:

Таблица сравнения размера и времени исполнения программ

	Статическая типизация процедурный подход	Статическая типизация объектно-ориентированный подход	Динамическая типизация	Язык Ассемблера
Размер исходных кодов программы, строки	444	429	248	1740
Считывание из файла 3 объектов, сек	0,003	0,004	0,072	0,003
Считывание из файла 20 объектов, сек	0,005	0,005	0,079	0,003
Случайная генерация 1000 объектов, сек	0,010	0,019	0,169	0,013
Случайная генерация 10000 объектов, сек	0,059	0,081	1,289	0,036

Программа, написанная на языке ассемблера, работает значительно быстрее любого другого варианта. Причина — прямая работа с регистрами, близость к железу. Однако минус — колоссальная сложность разработки по на нем. Поэтому резонно реализовывать на ассемблере особо ресурсоемкие задачи, но не в больших количествах, поскольку стоимость часа работы высококлассного программиста, помноженная на объем работы, не всегда дает эффект для бизнеса. Поэтому множество компаний переходит на языки с большим кол-вом синтаксического сахара: Kotlin, Python, JS. Это позволяет быстро писать много кода. Не самого быстрого, но это уже зависит от приоритетов компании.