

Описание (вариант 2, 23)

Реализовать контейнер для хранения альтернатив и их параметров.

Обобщённый артефакт: плоские геометрические фигуры, размещаемые в координатной сетке.

Базовые альтернативы и их параметры:

- Прямоугольник:
 - целочисленная координата по оси X левого верхнего угла
 - целочисленная координата по оси Y левого верхнего угла
 - целочисленная координата по оси X правого нижнего угла
 - целочисленная координата по оси Y правого нижнего угла
- Треугольник:
 - целочисленная координата по оси X первого угла
 - целочисленная координата по оси Y первого угла
 - целочисленная координата по оси X второго угла
 - целочисленная координата по оси Y второго угла
 - целочисленная координата по оси X третьего угла
 - целочисленная координата по оси Y третьего угла
- Круг:
 - целочисленная координата по оси X центра
 - целочисленная координата по оси Y центра
 - целочисленный радиус

Общая для всех альтернатив переменная – цвет: красный, оранжевый, жёлтый, зелёный, голубой, синий, фиолетовый.

Общая для всех альтернатив функция: вычисление периметра.

Обработка массива результатов общих функций: перемещение периметров, которые превышают среднее арифметическое значение, в начало.

Входные и выходные данные

Форматы входной команды:

- '-f <файл с входными данными> <файл для вывода информации о фигурах и их периметрах> <файл для вывода обработанного массива периметров>'
Например, '-f input.txt output.txt output_sorted.txt'.
- '-n <количество случайно генерируемых фигур> <файл для вывода информации о фигурах и их периметрах> <файл для вывода обработанного массива периметров>'
Например, '-n 20 output.txt output_sorted.txt'.

Формат описания фигур в файле входных данных:

- Прямоугольник:

1 <цвет>

<целочисленная координата по оси X левого верхнего угла> <целочисленная координата по оси Y левого верхнего угла> <целочисленная координата по оси X правого нижнего угла> <целочисленная координата по оси Y правого нижнего угла>

Например,

1 1

0 0 1 1

- прямоугольник красного цвета с левым нижним углом в вершине [0, 0] и правым верхним – в [1, 1].

- Треугольник:

2 <цвет>

<целочисленная координата по оси X первого угла> <целочисленная координата по оси Y первого угла> <целочисленная координата по оси X второго угла> <целочисленная координата по оси Y второго угла> <целочисленная координата по оси X третьего угла> <целочисленная координата по оси Y третьего угла>

Например,

2 2

0 0 1 1 0

- треугольник оранжевого цвета с координатами вершин: [0, 0], [0, 1], [1, 0].

- Круг:

3 <цвет>

<целочисленная координата по оси X центра> <целочисленная координата по оси Y центра> <целочисленный радиус>

Например,

3 7

0 0 1

- круг фиолетового цвета с центром в точке [0, 0] и радиусом 1.

Выходные файлы:

- вывод информации о фигурах и их периметрах:

Filled container:

Container contains <количество фигур> elements.

- для прямоугольника:

<порядковый номер>: It is Rectangle. Left top point: x = <целочисленная координата по оси X левого верхнего угла>, y = <целочисленная координата по оси Y левого верхнего угла>. Right bottom point: x = <целочисленная координата по оси X правого нижнего угла>, y = <целочисленная координата по оси Y правого нижнего угла>.

- для треугольника:

<порядковый номер>: It is Triangle. A-point: x = <целочисленная координата по оси X первого угла>, y = <целочисленная координата по оси Y первого угла>. B-point: x = <целочисленная координата по оси X второго угла>, y = <целочисленная координата по оси Y второго угла>. C-point: x = <целочисленная координата по оси X третьего угла>, y = <целочисленная координата по оси Y третьего угла>.

- для круга:

<порядковый номер>: It is Circle. Center: x = <целочисленная координата по оси X центра>, y = <целочисленная координата по оси Y центра>. Radius: r = <целочисленный радиус>.

Perimeter = <периметр>. Color: <цвет>

- вывод обработанного массива периметров:

Arithmetic mean: <среднее значение>

Perimeteres processed:

<...периметры всех фигур, отсортированные в соответствии с требованием...>

Метрики

Состав: 7 файлов реализации.

Размер файлов: 8.85КБ.

Время исполнения:

- 20 фигур – 2млс
- 100 фигур – 4млс
- 500 фигур – 19млс
- 2000 фигур – 48млс
- 10000 фигур – 230млс

Сравнение

Программа, разработанная при динамической типизации выполняется намного медленнее своих аналогов при статической типизации, однако размер файлов с кодом уменьшился.