1. **Результаты тестирования:**

Входные данные:

|  |  |  |
| --- | --- | --- |
|  | Фигура 1 | Фигура 2 |
| 1 | 0 0 21 21 SEGMENT BLACK | 11 31 31 11 SEGMENT RED |
| 2 | 200 0 0 100 SEGMENT GREEN | 100 0 0 200 SEGMENT BLUE |
| 3 | 70 50 50 0 CIRCLE BLUE | 15 30 140 90 SEGMENT GREEN |
| 4 | 270 300 150 0 CIRCLE RED | 45 190 370 70 SEGMENT BLUE |
| 5 | 170 300 50 0 CIRCLE GREEN | 45 90 370 270 RECTANGLE BLUE |
| 6 | 70 30 150 100 RECTANGLE BLACK | 145 90 370 270 RECTANGLE RED |
| 7 | 55 430 250 415 SEGMENT BLACK | 70 65 290 300 RECTANGLE GREEN |
| 8 | 170 120 75 0 CIRCLE BLUE | 130 310 110 0 CIRCLE RED |
| 9 | 470 320 75 40 SEGMENT BLUE | 330 60 460 190 RECTANGLE RED |
| 10 | 370 320 80 0 CIRCLE WHITE | 115 230 295 360 RECTANGLE RED |

Вывод:

|  |  |  |
| --- | --- | --- |
| № | Рисунок | Пересечение |
| 1 |  | true |
| 2 |  | true |
| 3 |  | true |
| 4 |  | false |
| 5 |  | true |
| 6 |  | true |
| 7 |  | false |
| 8 |  | false |
| 9 |  | false |
| 10 |  | true |

1. **Описание методов класса GraphicPic.**

* **Конструктор GraphicPic(String path):**

Создает пустой экземпляр класса LinkedList с конструктором класса Figure. Принимает в аргумент список по множеству фигур, заданному в текстовом файле, на основании этих данных создает экземпляр класса Figure и помещает его в список list.

* **Метод void show():**

Создает экземпляр класса JFrame, устанавливает кнопку выхода, создает экземпляр класса FigurePanel, добавляет его во frame, устанавливает размер окна и видимость.

* **Метод void insert(Figure f):**

Вставляет экземпляр класса Figure в список.

* **Метод void delete(Type i):**

Проходится по списку фигур, и удаляет элемент с типом i.

* **Метод GraphicPic commonWith(Figure r):**

Строит новый список, состоящий из фигур, которые имеют общие точки с данной фигурой.

* **Метод GraphicPic hasSquareBiggerThanS(double s):**

Строит новый список из фигур, которые имеют площадь больше, чем s.

* **Метод GraphicPic getByColor(Color [] colors):**

Строит новый список, содержащий только фигуры цветов из массива colors.

* **Метод save(String path):**

Создает экземпляр класса PrintWriter и записывает в него элементы списка фигур.

1. **Описание дополнительных классов.**
2. **Класс Figure:**

Данный класс предназначен для создания готовых фигур. Он содержит enum Type: прямоугольник, отрезок или круг (RECTANGLE, CIRCLE, SEGMENT); координаты фигуры; enum Color с цветами фигур(WHITE, BLACK, RED, GREEN, BLUE);конструктор класса; метод getSquare(), возвращающий площадь данной фигуры(для CIRCLE это значение равно 0); метод intersectWith(Figure another), возращаюший true или false, если данная фигура имеет или не имеет пересечение с фигурой, заданной в аргументе, соответственно.

1. **Класс FigurePanel:**

Данный класс предназначен для отображения фигур на экран. Он содержит переменную класса LinkedList с параметром класса Figure; 4 перегруженных конструктора класса; метод paint(Graphics g), рисующий фигуры в списке list.

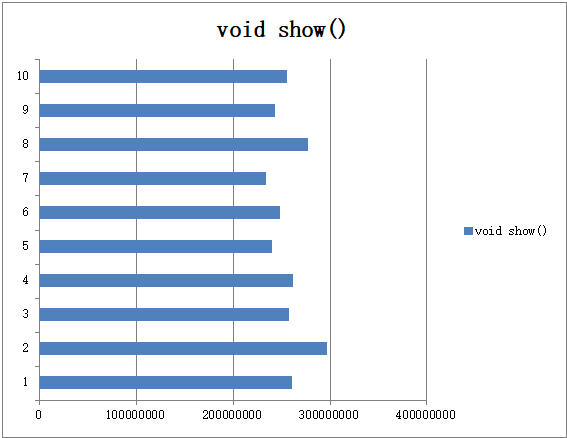
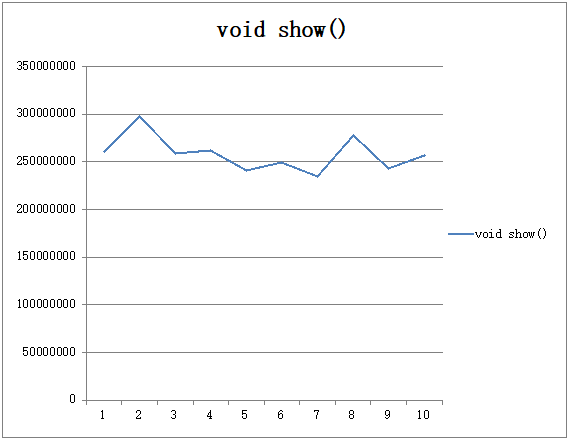
1. **Класс LinkedList:**

Данный класс представляет собой реализацию связанного списка и предназначен для хранения в себе объектов класса Figure.

1. **Анализ скорости работы методов класса GraphicPic.**

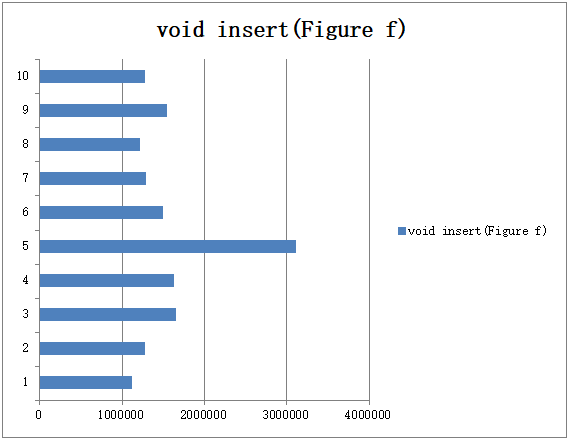
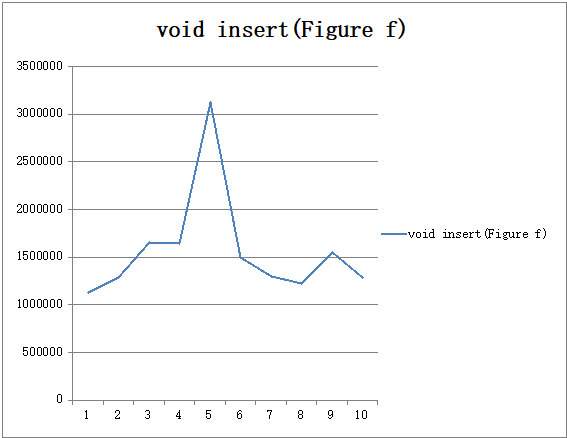
* **Метод void show():**

|  |  |
| --- | --- |
| № | Время выполнения (нано сек) |
| 1 | 260883400 |
| 2 | 297193300 |
| 3 | 258398800 |
| 4 | 262320800 |
| 5 | 240939700 |
| 6 | 248998100 |
| 7 | 234749800 |
| 8 | 277863300 |
| 9 | 243431800 |
| 10 | 256437200 |



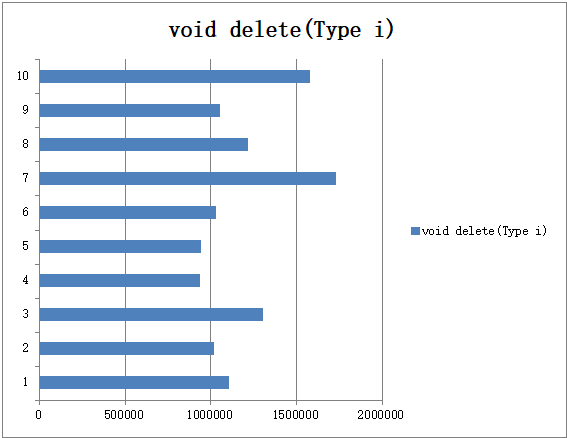
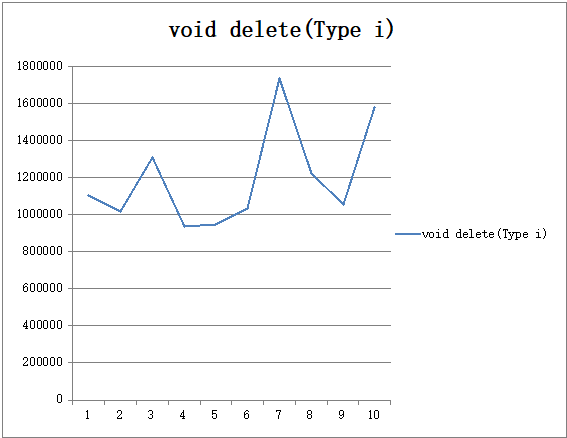
* **Метод void insert(Figure f):**

|  |  |
| --- | --- |
| № | Время выполнения (нано сек) |
| 1 | 1125400 |
| 2 | 1281700 |
| 3 | 1655400 |
| 4 | 1639800 |
| 5 | 3120500 |
| 6 | 1500800 |
| 7 | 1296800 |
| 8 | 1220600 |
| 9 | 1553300 |
| 10 | 1285300 |



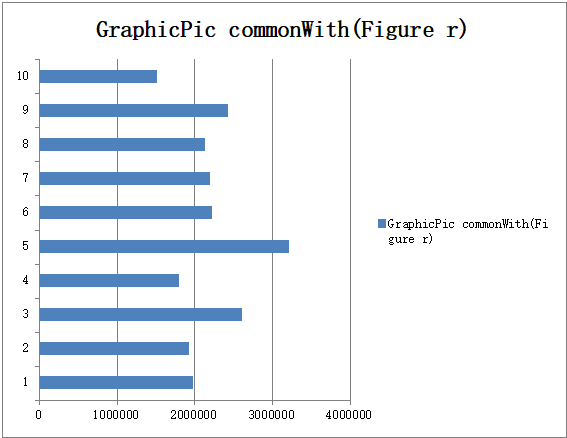
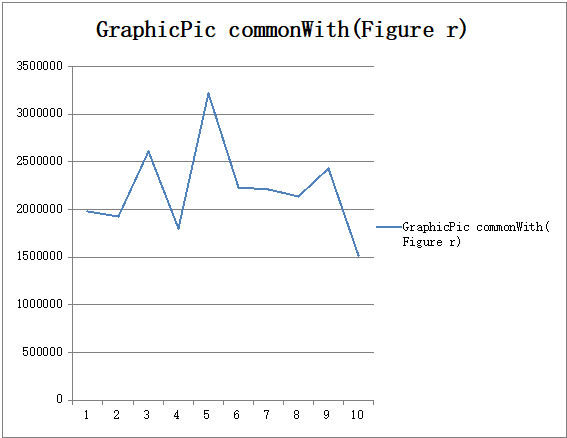
* **Метод void delete(Type i):**

|  |  |
| --- | --- |
| № | Время выполнения (нано сек) |
| 1 | 1105900 |
| 2 | 1019800 |
| 3 | 1309500 |
| 4 | 938700 |
| 5 | 949700 |
| 6 | 945700 |
| 7 | 1032200 |
| 8 | 1734900 |
| 9 | 1056700 |
| 10 | 1582000 |



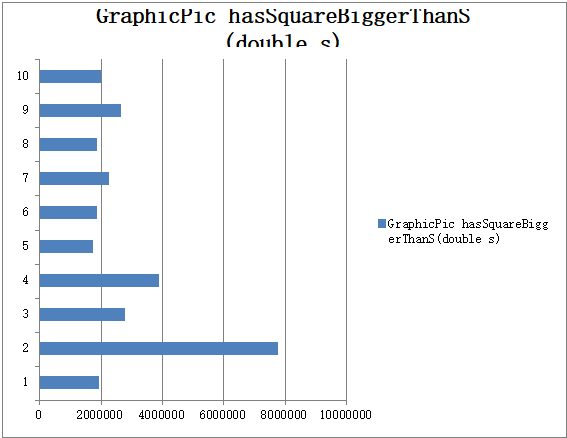
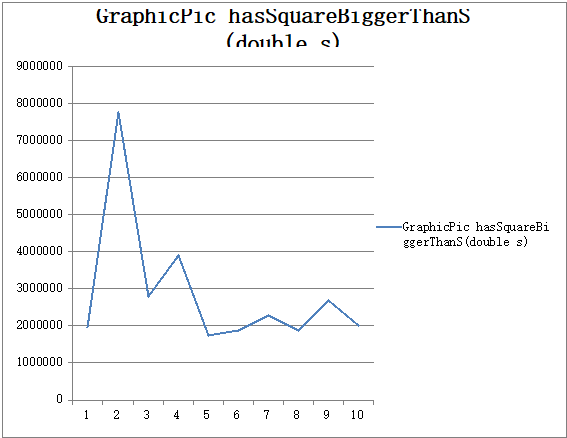
* **Метод GraphicPic commonWith(Figure r):**

|  |  |
| --- | --- |
| № | Время выполнения (нано сек) |
| 1 | 1981000 |
| 2 | 1928100 |
| 3 | 2611300 |
| 4 | 1802800 |
| 5 | 3213700 |
| 6 | 2227100 |
| 7 | 2205700 |
| 8 | 2141300 |
| 9 | 2426700 |
| 10 | 1514600 |



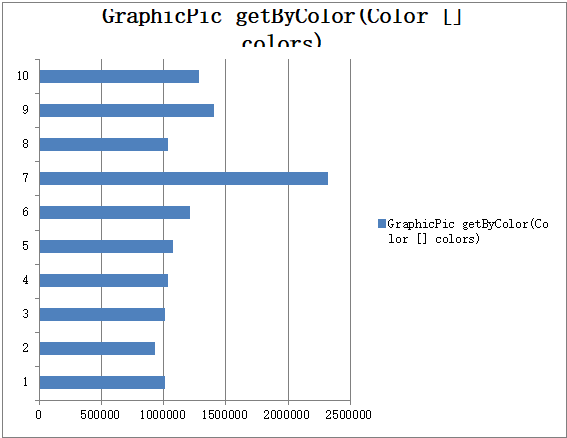
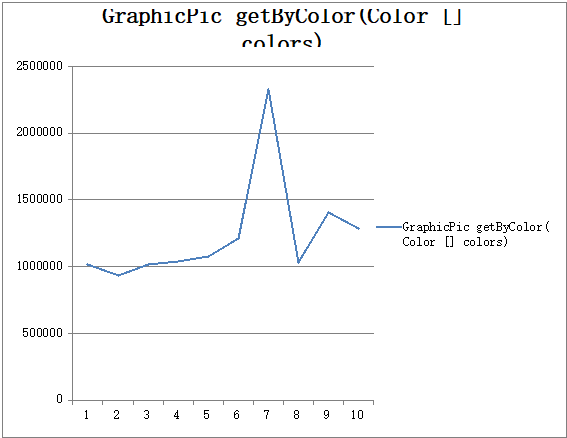
* **Метод GraphicPic hasSquareBiggerThanS(double s):**

|  |  |
| --- | --- |
| № | Время выполнения (нано сек) |
| 1 | 1959100 |
| 2 | 7771600 |
| 3 | 2804700 |
| 4 | 3891700 |
| 5 | 1742600 |
| 6 | 1888600 |
| 7 | 2271300 |
| 8 | 1870700 |
| 9 | 2677800 |
| 10 | 2002200 |



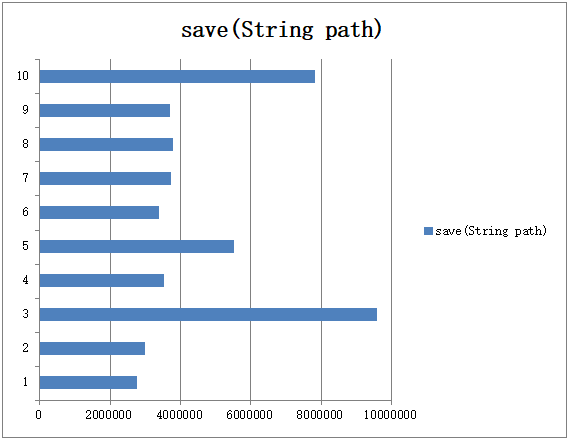
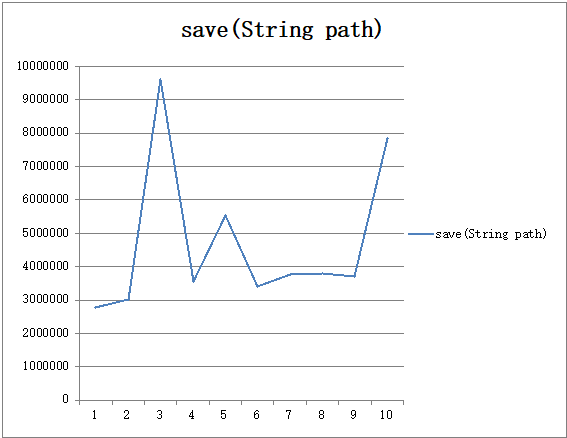
* **Метод GraphicPic getByColor(Color [] colors):**

|  |  |
| --- | --- |
| № | Время выполнения (нано сек) |
| 1 | 1012700 |
| 2 | 936100 |
| 3 | 1013200 |
| 4 | 1037200 |
| 5 | 1075000 |
| 6 | 1214900 |
| 7 | 2328300 |
| 8 | 1033600 |
| 9 | 1403900 |
| 10 | 1285500 |



* **Метод save(String path):**

|  |  |
| --- | --- |
| № | Время выполнения (нано сек) |
| 1 | 2788000 |
| 2 | 3007200 |
| 3 | 9601700 |
| 4 | 3540700 |
| 5 | 5532400 |
| 6 | 3406200 |
| 7 | 3753400 |
| 8 | 3804300 |
| 9 | 3717700 |
| 10 | 7836100 |



**Расчет t-критерия Стьютенда:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название | M | σ | N |
| 1 | void show() | 258 121 620 | 18514242,3 | 10 |
| 2 | void insert(Figure f) | 1 567 960‬ | 575403,25 | 10 |
| 3 | void delete(Type i) | 1 167 510 | 272260,04 | 10 |
| 4 | GraphicPic commonWith(Figure r) | 2 205 230 | 471583,5 | 10 |
| 5 | GraphicPic hasSquareBiggerThanS(double s) | 2 888 030‬ | 1834288,9 | 10 |
| 6 | GraphicPic getByColor(Color [] colors) | 1 234 040 | 410963,32 | 10 |
| 7 | save(String path) | 4 698 770 | 2279305,8 | 10 |

М - среднее арифметическое, σ - стандартное отклонение, N - количество.

Воспользуемся формулой t-критерий Стьюдента для нахождения эмпирического значения:



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номера методов | Значение формулы | Степень свободы | Уровень значимости | Вывод о различии |
| 2 и 3 | 1,98932548797112 | 18 | Больше 0,05 | Нет |
| 2 и 4 | 2,708772602929848‬ | 18 | Больше 0,01 | Есть |
| 2 и 5 | 2,171442491892358‬ | 18 | Больше 0,01 | Есть |
| 3 и 4 | 6,026369198841193‬ | 18 | Меньше 0,001 | Есть |
| 3 и 5 | 2,933999079146784‬ | 18 | Больше 0,001 | Есть |
| 4 и 6 | 4,909749692140272‬ | 18 | Меньше 0,001 | Есть |
| 4 и 7 | 3,387753728230045‬ | 18 | Больше 0,001 | Есть |

1. **Плюсы и минусы программы.**

Плюсы:

* Прямой вывод всех фигур на экран
* Возможность изображения нескольких фигур в одном окне
* Ускоренный поиск фигур в коллекции с помощью реализации связанного списка
* Информация о пересечении двух разных фигур и их площадях
* Возможность построения нового списка фигур на основании исходного
* Сохранение информации о фигурах в файле

Минусы:

* Невозможность ввода фигур с отрицательными значениями координат
* Ограниченный выбор цветов
* Файл с информациями о фигурах перезаписывается при новом запуске программы

1. **Особенности данной реализации**

Данный класс представляет собой графический рисунок, содержащий список объектов: прямоугольников, отрезков и окружностей. Объекты фигур создаются на основании 6 параметров: координаты первой точки, координаты второй точки, тип фигуры и ее цвет (для круга это координата ее центра, радиус, тип и цвет). Все фигуры добавляются в связанный список в отдельном классе и особым методом рисуются на экран. Параметры входных данных фигуры сохраняются в файле.

Реализация этой программы может использоваться при необходимости отображения фигур на графике для решения геометрических задач, при поиске пересечения между ними и при их размещении на оси. Также возможна польза при создании простых изображений.

Зависимость времени(наносек.) от размера входных данных

1)Вставка:

Тут представлено время при записи n фигур в список.

по 2000 замеров для каждого размера

|  |  |  |
| --- | --- | --- |
| Размер | Среднее | Среднеквадр. отклонение |
| 100 | 1757 | 1355 |
| 200 | 3471 | 757 |
| 300 | 3372 | 1866 |
| 400 | 2616 | 148 |
| 500 | 3228 | 119 |
| 600 | 4036 | 659 |
| 700 | 4705 | 932 |
| 800 | 5429 | 858 |
| 900 | 5563 | 908 |
| 1000 | 6740 | 2336 |

Сложность: O(n)

2)Удаление:

Тут представлено время удаления какого-либо типа фигур при n фигурах в списке.

|  |  |  |
| --- | --- | --- |
| Размер | Среднее | Среднеквадр. отклонение |
| 100 | 2952 | 279 |
| 200 | 9781 | 755 |
| 300 | 21613 | 2565 |
| 400 | 38653 | 5224 |
| 500 | 74033 | 11265 |
| 600 | 136471 | 26919 |
| 700 | 192850 | 30291 |
| 800 | 269024 | 32331 |
| 900 | 363913 | 38506 |
| 1000 | 457976 | 44858 |

Сложность: O(n2)

3)Проверка пересечений:

Тут представлено время, за которое составляется список из фигур, пересекающихся с данной. В списке n фигур.

Сложность: O(n)