**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР.**

**отчет**

**по курсовой работе**

**по дисциплине «Алгоритмы и структуры данных»**

**Тема: «Реализация алгоритма генерирования команд отрисовывания объектов двумерной графики.».**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1301 |  | Ищенко Д.О. |
| Преподаватель |  | Давидчук А.Н. |

Санкт-Петербург

2022

**ЗАДАНИЕ**

**на курсовую работу**

|  |  |  |
| --- | --- | --- |
| Студентка Ищенко Д.О. | | |
| Группа 1301 | | |
| Тема работы: Реализация алгоритма генерирования команд отрисовывания объектов двумерной графики. | | |
| Исходные данные:  Файл двумерной графики в формате svg. | | |
| Содержание пояснительной записки:  Формулировка задания, теоретические сведения, файловая организация проекта, описание алгоритмов и структур данных программы, пример работы программы, текст программы.  Предполагаемый объем пояснительной записки:  Не менее 5 страниц. | | |
| Дата выдачи задания: 14.11.2022 | | |
| Дата сдачи реферата: 19.12.2021 | | |
| Дата защиты реферата: 19.12.2021 | | |
| Студентка |  | Ищенко Д.О. |
| Преподаватель |  | Давидчук А.Н. |

**Аннотация**

**Summary**

Оглавление

[Формулировка задания. 5](#_Toc28096)

[Теоретические сведения. 6](#_Toc23664)

[Файловая организация проекта. 7](#_Toc19901)

[Описание алгоритмов и структур данных программы. 7](#_Toc6063)

[Пример работы программы. 9](#_Toc7803)

[Текст программы. 9](#_Toc24991)

# Формулировка задания.

Реализовать алгоритм генерирования команд отрисовывания объектов двумерной графики, заданных параметрически, таким образом чтобы минимизировать расстояния перемещения поднятой “кисти”.

В качестве языка команд использовать gcode, в качестве входных данных двумерной графики - файлы svg.

**Контрольный пример.**

Входные данные: svg - файл

<svg version="1.1"  
 width="195" height="195"  
 xmlns="http://www.w3.org/2000/svg">  
  
 <line x1="60" y1="80" x2="60" y2="20" stroke="red" />  
 <line x1="100" y1="80" x2="100" y2="20" stroke="red" />  
 <line x1="40" y1="130" x2="30" y2="100" stroke="red" />  
 <line x1="120" y1="130" x2="130" y2="100" stroke="red" />  
 <line x1="115" y1="130" x2="45" y2="130" stroke="red" />  
  
  
</svg>

Выходные данные: Gcode - файл

G28

G1 X70.300000Z3 F5000.0

G1 X70.300000 Y60.300000 Z0 F5000.0

;SEGMENT

G1 Z5 F3000

; move to point (60:20)

G1 X175 Y215 F3000

G1 Z0 F3000

G1 X175 Y155 F3000; draw

;RECT

G1 Z5 F3000

; move to point (120:100)

G1 X115 Y135 F3000

G1 Z0 F3000

G1 X115 Y125 F3000; draw

G1 X175 Y125 F3000; draw

G1 X175 Y135 F3000; draw

G1 X115 Y135 F3000; draw

;SEGMENT

G1 Z5 F3000

; move to point (100:80)

G1 X135 Y155 F3000

G1 Z0 F3000

G1 X135 Y215 F3000; draw

G1 Z5 F3000

G1 X70 Y60 F5000.0

G1 X0 Y0

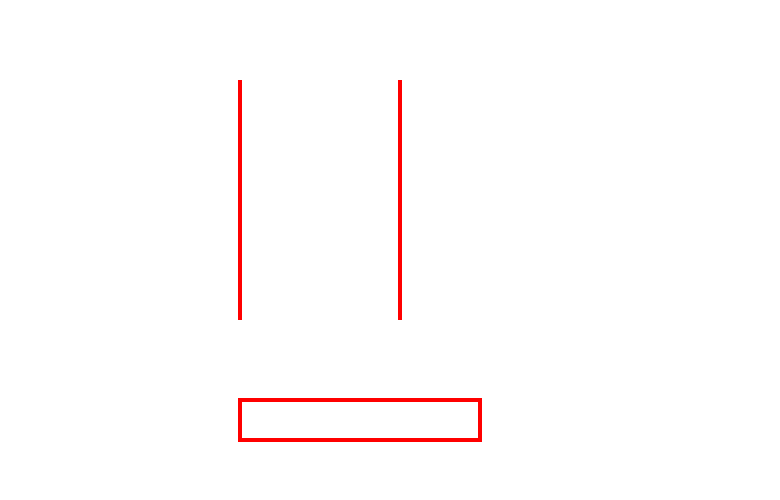
****

рис. 1

Рис. 1 является графческим представлением входного svg-файла.

# Теоретические сведения.

**Структура документа SVG.**

Ниже приведены основные сведения о тэгах формата svg, с которыми работает алгоритм.

* <circle cx= “50” cy=”30” r=”80”/>
* <ellipse cx="100" cy="50" rx="100" ry="50" />
* <line x1="0" y1="80" x2="100" y2="20" stroke="black" />

Требования к svg файлу:

\* содержит ровно один, закрытый тег svg с ненулевыми параметрами width и height.

\* в этом теге есть хотябы один элемент из поддерживаемых (см выше)

\* размер svg элемента не превышает размеры области печати в мм (165х175).

**GCode**

*G-код — язык программирования устройств с числовым программным управлением (ЧПУ).*

Ниже описаны основные команды языка gcode, с которыми работает алгоритм.

G28 ; переместиться в начальную позицию

G0 F1500 ; утсановить скорость перемещения g0 - холостое перемещение

G1 X0.1 Y20 Z0.3 ; g1 - линейное перемещение

# Файловая организация проекта.

# Описание алгоритмов и структур данных программы.

## Структуры данных.

## Алгоритмы.

Алгоритм **setFiguresDrawingOrder** находит кратчайший путь через все фигуры. Он работает следующим образом:

1) построить двунаправленный взвешенный полный граф на n вершинах, где весами являются расстояния от всех фигур до всех.

2) найти в нем путь кратчайший проходящий через все вершины

### Алгоритмы getNearestPoint(figVarians, figVariants).

Для случая двух окружностей задача сводится к нахождению точек пересечения отрезка на центрах окружностей и второй окружности.

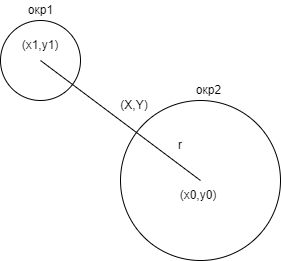
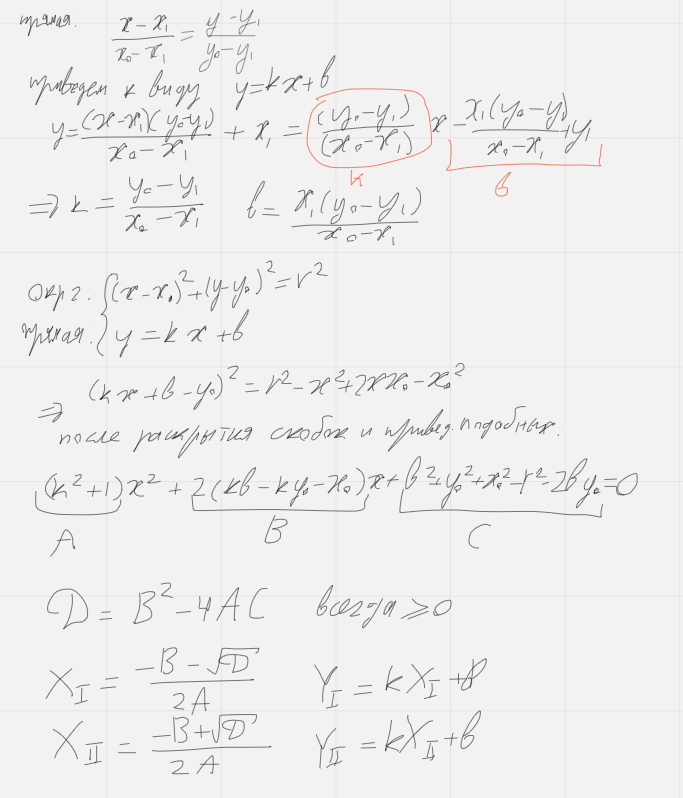


рис. 2

Дано: x1, y1, x0, y0, r

Найти: X,Y

Решение:



Далее по алгоритму проверки принадлежности точки отрезку (pointBelongs для класса Segment) выбирается нужная пара значений X,Y

# Пример работы программы.

# Текст программы.

[Ссылка на github](https://github.com/Nekttuman/Algosee)