



**СОФИЙСКИ УНИВЕРСИТЕТ "СВ. КЛИМЕНТ ОХРИДСКИ"**

**ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА**

## **Документация на проект** по Обектно ориентирано програмиране

НА ТЕМА:

### **Проект 2: SVG**

Изготвил:  
Росица Деянова  
Фак. № 71924  
Информационни системи, 2 курс

# Съдържание

Глава 1: Увод.....	3
Описание и идея на проекта: .....	3
Цел и задачи на разработката.....	3
Глава 2: Проектиране и Реализация .....	3
String.h.....	3
Vector.h.....	4
Class Shape.....	4
Методи: .....	4
Класове, които наследяват class Json: .....	5
Functions.h.....	5
Функции .....	5
Main.cpp .....	6
Потребителски интерфейс.....	6
Глава 5: Заключение .....	6
GITHUB линк.....	6

# Глава 1: Увод

## Описание и идея на проекта:

SVG е маркиращ език, базиран на XML, за описване на двумерна векторна графика с възможност за включване и на растерни изображения. Идеята на проекта е програма, която чете и Parse-ва един SVG текстов файл и запамятава обектите (key-value) в паметта на програмата с цел използването им на по-късен етап.

## Цел и задачи на разработката

### Цели:

- Постигане на четим код, отговарящ на принципите на ООП.
- Програмата трябва да може да отваря и чете текстови файлове с SVG синтаксис.
- След отварянето на такъв файл, програмата трябва да Parse-не обектите от файла в паметта на програмата.
- Програмата трябва да предоставя лесен и удобен начин за работа с тези обекти: принтиране, добавяне, премахване и т.н.

# Глава 2: Проектиране и Реализация

След отваряне на файла, програмата ще прочита текста ред по ред, по време на което ако забележи ред започващ с нужните ни тагове "`<rect`", "`<circle`" или "`<line`" ще раздели реда спрямо интервали и ще го запише във вектор от елементи String.

За тази цел използвам класовете String (String.h) и Vector (Vector.h).

## String.h

Този клас се съдържа променливата `char* data`. И методите, използвани за проекта:

- `int length()const;` - Връща дължината на String
- `int find(char c, int lastToken);` - обхожда String до lastToken и връща положението на първия срещнат char c
- `void empty();` - изпразва стойността на String
- `String substr(size_t startPos, size_t length);` - Връща подстринг с начало startPos и дадена дължина
- `Vector<String> tokenize(char c, bool isReadingFromFile);` - разделя String спрямо char c и ги пълни във вектор от String
- `int toInt(const char* txt);` - Парсва данните от char\* към int

- `char* getString();` - връща съдържанието на String

## Vector.h

Този клас съдържа променливите `T* data`, `size_t size`, `size_t capacity`. И методите използвани за проекта:

- `size_t getSize()const;` - Връща размера на вектора
- `Vector<T>& push_back(const T& New);` - Записва нова данна в края
- `Vector<T>& eraseAt(size_t n);` - Изтрива данна от конкретна позиция
- `Vector<T>& empty();` - изпразва вектора, трийейки всичките му член данни

## Class Shape

Методи:

- `char* getName()`  
Функция, връщаща името на фигурата. Името може да бъде rectangle, circle или line.
- `virtual void print();`  
Виртуална функция, която се използва при принтиране на контейнера в конзолата. Извикват се `print()` функции на класовете, които наследяват Shape и се принтират съответните променливи от тези класове.
- `virtual void writeOut(std::ostream& out)`  
Виртуална функция, принтираща контейнера във файл, добавяйки му нужния header и footer. Извикват се `writeOut()` функции на класовете, които наследяват Shape и се запазват променливите им в подадения файл.
- `void setX(int x);`  
Приема `int` параметър и го присвоява на променливата `x`.
- `void setY(int y);`  
Приема `int` параметър и го присвоява на променливата `y`.
- `void setColor(char* color);`  
Приема `char*` параметър и го присвоява на променливата `color`.
- `int getX();`  
Връща променливата `x`.
- `int getY();`  
Връща променливата `y`.
- `char* getColor();`  
Връща променливата `color`.

Класове, които наследяват class Json:

- Rectangle
- Circle
- Line

Всеки клас отговаря на съответния тип фигура, която приложението поддържа, заедно с нейните параметри и методи.

## Functions.h

### Функции

Functions.h е помощен файл, създаден с цел по-добро организиране на функциите.

В себе си той съдържа:

- **String getType(String& word)**  
Разпознава и връща типа данни, които се подават в String word. Пример: x="52", функцията ни ще върне "x". Защото стойността 52 трябва да бъде присвоена на променливата x.
- **String getInfo(String& word, const String& type)**  
Разпознава и връща стойността, която трябва да бъде присвоена. Пример: x="52", функцията ни ще върне "52".
- **Rectangle\* constructRectangle(Vector<String>& words)**  
Получава вектор и чрез гореспоменатите функции отделя стойностите от типовете им данни и ги присвоява, връщайки готов обект Rectangle.
- **Circle\* constructCircle(Vector<String>& words)**  
Получава вектор и чрез гореспоменатите функции отделя стойностите от типовете им данни и ги присвоява, връщайки готов обект Circle.
- **Line\* constructLine(Vector<String>& words)**  
Получава вектор и чрез гореспоменатите функции отделя стойностите от типовете им данни и ги присвоява, връщайки готов обект Line.
- **Shape\* recogniseShape(Vector<String>& words)**  
Получава вектор и разпознава нужния за конструкция обект, след което извиква constructLine, constructCircle или constructRectangle.
- **Vector<Shape\*> openFile(const char\* input\_file\_name, Vector<Shape\*> shapes)**  
Отваря файл, прочита го, използвайки String::tokenize го дели спрямо спейсове, и връща вектор. Извиква recogniseShape.

- `void save(Vector<Shape*>& shapes, const char* filename)`  
Запазва данните във файл, използвайки виртуалната функция `writeOut`.
- `String checkTranslation(String word)`  
Отделя и връща типа на транслация (`vertical/horizontal`).

## Main.cpp

Потребителски интерфейс

Командите са разделени на 2 вида

- **Команден ред**

Това е главното меню на програмата, чрез който се работи с файловете.

Реализирани са следните функционалности:

`void Open(); void Close(); void Save(); void SaveAs(); void Help(); void exit();`

- **Проектна функционалност**

Това е функционалното меню на програмата, чрез който потребителят работи с данните от файловете:

- `print`
- `create`
- `erase`
- `translate`

## Глава 5: Заключение

GITHUB линк

- <https://github.com/RosiDeyanova/SVG>