



**СУ “Св. Климент Охридски”,
ФМИ – Софтуерно инженерство
Курсов проект по Обектно-ориентирано
програмиране**

Работа със SVG файлове

Кирил Яворов Русев, факултетен № 61887

Съдържание

| | |
|--------------------------------------|---|
| 1. Архитектура на приложението | 2 |
| 2. Основни класове | 4 |
| 3. Използван чужд код | 6 |

1. Архитектура на приложението

Приложението може да създава, обработва и запамята файлове във SVG формат. То поддържа три основни фигури - кръг, правоъгълник и линия.

При стартиране на приложението се извежда текст **"Enter help for more information."**.

След това на нов ред се извежда **">"** и потребителят трябва да въведе команда, която да се изпълни. Докато не се въведе команда **open**, могат да се изпълняват само командите **help** и **exit**.

- **help**
Извежда на екрана помощен списък с действията на всички останали команди.
- **exit**
Излиза от програмата, като се изпълнява и командата **close**, ако не е била въведена преди това.

При въвеждане на команда **open** може да се укаже име на файл. Той се отваря, информацията в него се прочита и след това се затваря. Ако файл с това име не съществува или не е указано име на файл, се създава нов файл. Прочита се само информацията за фигурите, които приложението поддържа. Всички останали фигури се игнорират и при запазване на файла изчезват. Не може да има повече от един отворен файл едновременно.

При отворен файл, приложението може да изпълнява следните команди, които променят фигурите:

- **print**
Извежда на екрана всички фигури.
- **create**
Създава нова фигура, като потребителят трябва да укаже всичката необходима информация за нея.
- **erase**
Изтрива фигура, като потребителят трябва да укаже нейният индекс.

- **translate**

Транслира една или всички фигури. Ако потребителят иска да транслира само една фигура, той трябва да укаже нейният индекс. В противен случай се транслират всички фигури. Също така трябва да бъде посочено и с колко ще бъде транслирана(транслирани) фигурата(фигурите) по вертикал и по хоризонтал.

- **within**

Извежда на екрана всички фигури, които изцяло се съдържат в даден регион. Потребителят трябва да укаже какъв да бъде регионът - кръг или правоъгълник, както и информация за това къде се намира този регион.

При отворен файл могат да се изпълняват и следните команди за работа с файл:

- **save**

Запамятава направените промени в текущия файл. Ако този файл е нов, се извежда съобщение "**The current file is new. Enter a directory to save it in and the file name:**" и потребителят трябва да укаже пътя на новия файл.

- **saveas**

Запамятава направените промени в нов файл, като потребителят трябва да укаже пътя на новия файл.

- **close**

Затваря текущо отворения документ, независимо дали промените в него са запазени. Затварянето изчиства всички промени. След това програмата може да изпълнява само командите **help**, **open** и **exit**.

Ако информацията при въвеждането на командите е непълна, се извежда подходящо съобщение за грешка.

2. Основни класове

В програмният код са декларирани, както следва:

- клас **Point**
- базов клас **Figure**
- наследяващи го класове **Circle**, **Rectangle** и **Line**
- клас **Group**
- клас **XMLFileParser**
- клас **Commands**

В класа **Point** са декларирани:

- две целочислени променливи *x* и *y* за координатите, на които се намира точката
- конструктор по подразбиране и конструктор с параметри
- селектори и мутатори за *x* и *y*
- функция, която изчислява разстоянието между две точки
- приятелски функции за извеждане на данните

В базовият клас **Figure** са декларирани:

- enum *shape*, който съхранява типа на фигурата
- член-данните на класа са всички компоненти общи за класовете, които го наследяват – точка *point*, *fill* (цвета на вътрешността на фигурата) и *type* (типа на фигурата)
- конструктор с подразбиращи се параметри, конструктор с параметри и конструктор за копиране
- оператор `=`
- селектори и мутатори за *point*, *fill* и *type*
- виртуална функция за транслиране на фигурата
- виртуални функции, които проверяват дали фигурата се намира в правоъгълник или в кръг
- виртуална функция за задаване на стойностите на декларираните в класа променливи

Извън класа е предефиниран оператор `<<` за *shape* и *Figure*

В класът **Circle**, който е производен на класа *Figure*, са декларирани:

- целочислена променливи *radius* за радиуса на кръга
- конструктор по подразбиране и конструктор с параметри

- селектори и мутатори за radius
- функции, които проверяват дали фигурата се намира в правоъгълник или в кръг
- функция за задаване на стойностите на декларираните в класа променливи

В класа **Rectangle**, който е производен на Figure, са декларирани:

- целочислени променливи width и height за дължината и широчината на правоъгълника
- конструктор по подразбиране и конструктор с параметри
- селектори и мутатори за width и height
- функции, които проверяват дали фигурата се намира в правоъгълник или в кръг
- функция за задаване на стойностите на декларираните в класа променливи

В класа **Line**, който е производен на Figure, са декларирани:

- точка end за края на отсечката
- конструктор по подразбиране и конструктор с параметри
- селектори и мутатори за end
- функция за транслиране на фигурата
- функции, които проверяват дали фигурата се намира в правоъгълник или в кръг
- функция за задаване на стойностите на декларираните в класа променливи

В класа **Group** са декларирани:

- вектор от указатели към фигури shapes
- голяма четворка
- функция, която връща броя на фигурите в групата
- функция, която добавя фигура в края на shapes
- функция, която премахва фигура от shapes
- функция, която премахва фигура от края на shapes
- оператор []
- функция, която проверява кои фигури от shapes са в правоъгълник или в кръг
- функция, която изчиства динамично заделената памет

- приятелска функция, предефинираща оператор << за класа Класът е реализиран като singleton.

В класа **XMLFileParser** са декларирани следните статични функции:

- функция, която по въведен път на файла връща името му
- функция, която отваря и прочита файл, като запазва информацията в него във вектор от низове, като всеки низ отговаря на ред във файла
- функция, която прочита текста на файла и добавя към групата от фигури тези, за които има информация във файла
- функция, която прочита ред от файла и добавя към групата от фигури тези, за които има информация във файла
- функция, която връща число, намиращо се в низ
- функция, която прочита информацията за кръг във файла и го добавя към групата от фигури
- функция, която прочита информацията за правоъгълник във файла и го добавя към групата от фигури
- функция, която прочита информацията за отсечка във файла и я добавя към групата от фигури
- функция, която връща текст на файл, съдържащ всички фигури в групата от фигури
- функция, която отваря и пренаписва файл

В класа **Commands** са декларирани следните статични функции:

- функция за въвеждане на команда, която запазва въведената информация във вектор от низове, които са думи от командата
- функции за всяка една от командите help, create, erase, translate, within, save, saveas, open и close
- функция меню, в която се въвеждат и изпълняват командите

3. Използван чужд код

При изготвяне на проекта не е използван чужд код. Използвани са библиотеките string, vector, iostream и fstream.