

Работа със SVG файлове

В рамките на този проект трябва да се разработи приложение, което работи със файлове във [Scalable Vector Graphics \(SVG\) формат](#). Приложението трябва да може да зарежда фигури от файла, да извършва върху тях дадени операции, след което да може да записва промените обратно на диска.

За улеснение, в рамките на проекта ще работим само с основните фигури (basic shapes) в SVG. Приложението ви трябва да поддържа поне три от тях. Например можете да изберете да се поддържат линия, кръг и правоъгълник. За повече информация за това кои са базовите фигури, вижте <https://www.w3.org/TR/SVG/shapes.html>.

Също така, за улеснение считаме, че координатната система, в която работим е тази по подразбиране: положителната полуос X сочи надясно, а положителната полуос Y сочи надолу.

Дизайнът на приложението трябва да е такъв, че да позволява при нужда лесно да можете да добавите поддръжка на нови фигури.

Когато зареждате съдържанието на един SVG файл, трябва да прочетете само фигурите, които приложението ви поддържа и можете да игнорирате всички останали SVG елементи.

След като заредите фигурите, потребителят трябва да може да изпълнява дадените в следващия раздел команди, които добавят, изтриват или променят фигурите.

Когато записвате фигурите във файл, трябва да генерирате валиден SVG файл

Операции

След като приложението отвори даден файл, то трябва да може да извършва посочените по-долу операции, в допълнение на общите операции (open, close, save, save as, help и exit):

print	Извежда на екрана всички фигури.
create	Създава нова фигура.
erase <n>	Изтрива фигура с пореден номер <n>.
translate [<n>]	Транслира фигурата с пореден номер <n> или всички фигури, ако <n> не е указано.
within <option> ...	Извежда на екрана всички фигури, които изцяло се съдържат в даден регион. Потребителят може да укаже чрез <option> какъв да бъде регионът – кръг (circle) или правоъгълник (rectangle)

Общи операции

Open

Зарежда съдържанието на даден файл. Ако такъв не съществува се създава нов с празно съдържание.

Всички останали команди могат да се изпълняват само ако има успешно зареден файл.

След като файлът бъде отворен и се прочете, той се затваря и приложението ви вече не трябва да работи с него, освен ако потребителят не поиска да запише обратно направените промени (вижте командата save по-долу), в който случай файлът трябва да се отвори наново. За целта трябва да изберете подходящо представяне на информацията от файла.

Ако при зареждането на данните, приложението ви открие грешка, то трябва да изведе подходящо съобщение за грешка и да прекрати своето изпълнение.

```
> open C:\Temp\file.xml
```

```
Successfully opened file.xml
```

Close

Затваря текущо отворения документ. Затварянето изчиства текущо заредената информация и след това програмата не може да изпълнява други команди, освен отваряне на файл (Open).

```
> close
```

```
Successfully closed file.xml
```

Save

Записва направените промени обратно в същия файл, от който са били прочетени данните.

```
> save
```

```
Successfully saved file.xml
```

Save As

Записва направените промени във файл, като позволява на потребителя да укаже неговия път.

```
> saveas "C:\Temp\another file.xml"
```

```
Successfully saved another file.xml
```

Help

Извежда кратка информация за поддържаните от програмата команди.

```
> help
```

The following commands are supported:

open <file>	opens <file>
-------------	--------------

close	closes currently opened file
-------	------------------------------

save	saves the currently open file
------	-------------------------------

saveas <file>	saves the currently open file in <file>
---------------	---

help	prints this information
exit	exists the program

Exit

Излиза от програмата

```
> exit
```

Exiting the program...

Примерен SVG файл figures.svg

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
  "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd
```

Пример за работа на програмата

```
> open figures.svg
```

Successfully opened figures.svg

```
> print
```

1. rectangle 5 5 10 10 green
2. circle 5 5 10 blue
3. rectangle 100 60 10 10 red

```
> create rectangle -1000 -1000 10 20 yellow
```

Successfully created rectangle (4)

```
> print
```

1. rectangle 1 1 10 20 green
2. circle 5 5 10 blue

3. rectangle 100 60 10 10 red
4. rectangle 1000 1000 10 20 yellow

> within rectangle 0 0 30 30

1. rectangle 5 5 10 10 green
2. circle 5 5 10 blue

> within circle 0 0 5

No figures are located within circle 0 0 5

> erase 2

Erased a circle (2)

> erase 100

There is no figure number 100!

> print

1. rectangle 5 5 10 10 green
2. rectangle 100 60 10 10 red
3. rectangle 1000 1000 10 20 yellow

> translate vertical=10 horizontal=100

Translated all figures

> print

1. rectangle 105 15 10 10 green
2. rectangle 200 70 10 10 red
3. rectangle 1100 1010 10 20 yellow

> save

Successfully saved the changes to figures.svg

> exit

Exit