



Обектно-ориентирано програмиране

Домашна работа №3

Пояснение:

Реализирайте задачите спазвайки добрите ООП принципи (абстракция, капсулация, наследяване, полиморфизъм)

Решение, в които не са спазени ООП принципите ще бъдат оценени с 0 точки.

Предадените от вас решения трябва да могат да се компилират успешно на Visual C++ или GCC.

Не е разрешено да ползвате библиотеки от STL и STL функции.

Изисквания за предаване:

Всички задачи ще бъдат проверени автоматично за преписване. Файловете с голямо съвпадение ще бъдат проверени ръчно и при установено плагиатство ще бъдат **анулирани**.

Предаване на домашното в указания срок от всеки студент като .zip архив със следното име:

(номер_на_домашно)_SI_(курс)_(група)_(факултетен_номер)

- (номер_на_домашно) е цяло число, отговарящо на номера на домашното за което се отнася решението (например 1);
- (курс) е цяло число, отговарящо на курс (например 1);
- (група) е цяло число, отговарящо на групата Ви (например 1);
- (факултетен_номер) е цяло число, отговарящо на факултетния Ви номер (например 12345);

Пример за .zip архив за домашно: 3_SI_1_1_12345.zip

Архивът да съдържа само изходен код (.cpp и .h/.hpp файлове) с решение отговарящо на условията на задачите, като файловете изходен код за всяка задача трябва да са разположени в папка с име (номер_на_задача).

Качване на архива на посоченото място в Moodle;





Обектно-ориентирано програмиране

Домашна работа №3

Задача: SVG файлове

В рамките на това домашно трябва да се разработи приложение, което работи със файлове в Scalable Vector Graphics (SVG) формат.

Приложението трябва да може да зарежда фигури от файла, да извършва дадени операции, върху тях, след което да може да записва промените обратно на диска.

За улеснение, в рамките на домашното 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>.

За улеснение ще считаме, че координатната система, в която работим, е тази по подразбиране:

- положителната полуос X сочи надясно.
- положителната полуос Y сочи нагоре.

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





Обектно-ориентирано програмиране

Домашна работа №3

Операции

| | |
|-----------|--|
| Print | Извежда на екрана всички фигури. |
| Create | Създава нова фигура. |
| Erase | Изтрива фигура |
| Translate | Транслира една или всички фигури. Ако потребителят не посочи конкретна фигура, тогава се транслират всички фигури; ако се посочи конкретна – променя се само тя. |
| Within | Извежда на екрана всички фигури, които изцяло се съдържат в даден регион. Потребителят може да укаже какъв да бъде регионът – кръг или правоъгълник |
| PointIn | Извежда на екрана всички фигури, които съдържат подадена точка. |
| Areas | Извежда лицата на всички фигури |
| Pers | Извежда обиколката на всички фигури. |





Обектно-ориентирано програмиране

Домашна работа №3

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">

<svg>

  <rect x="5" y="5" width="10" height="10" fill="green" />

  <circle cx="5" cy="5" r="10" fill="blue" />

  <rect x="100" y="60" width="10" height="10" fill="red" />

</svg>
```

Пример за работа на програмата

```
> open figures.svg
Successfully opened figures.svg

> print
1. rectangle are 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)
```





Обектно-ориентирано програмиране

Домашна работа №3

> *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*





Обектно-ориентирано програмиране

Домашна работа №3

> 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

