

# Документация на курсов проект по дисциплината Обектно-ориентирано програмиране

## Тема:

### “Приложение за работа с векторни изображения”

#### Съдържание:

1. Увод
  - 1.1. Цел на поставената задача
  - 1.2. Описание и идея на проекта
2. Използвани концепции
3. Архитектура на проекта
  - 3.1. Структура на класовете и предназначението им
  - 3.2. Структура на проекта
4. Тестване на приложението
5. Потенциални разширения на проекта

#### 1. Увод:

Задачата, която ни се поставя в рамките на този проект е реализиране на конзолно приложение, чрез което да се взаимодейства с фигури от векторни изображения, като най-важно е изискването за спазване на добрите ООП принципи и практики по време на неговата реализация.

Приложението трябва да реализира четене на фигури от подаден файл по абсолютен или релативен път. Поддържаните фигури по изискване са четири основни типа: отсечки, дъги от окръжности, многоъгълници и правоъгълници, всяка от която се характеризира със своите параметри. На потребителя трябва да му се предостави възможността да извършва операции над тези фигури като скалиране, транслиране, групиране, извеждане на фигурите на конзолния екран и т.н. Приложението позволява още и създаване на нови фигури от поддържаните типове. По време и след приключване на работа с приложението се предоставя възможност за запис на извършените промени по данните съответно в първоначалния файл или в нов такъв, посочен от потребителя. Освен файлове от SVG формат, приложението предоставя и опция за четене и запис в текстови файлове. От приложението се поддържа и още един

тип базова фигура „група“, като потребител може да групира елементи в зададен регион, както и да добавя и премахва обекти от дадена група.

## 2. Използвани концепции:

За реализацията на задачата са използвани основни ООП концепции и практики, като пример за някои от тях са наследяване, полиморфизъм, динамично свързване, абстракция, енкапсулация. В реализацията на някои класове са включени и шаблони за дизайн с цел удобство в разработката на проекта. Пример за използвани шаблони са „Абстрактна фабрика“ и „Сингълтън“.

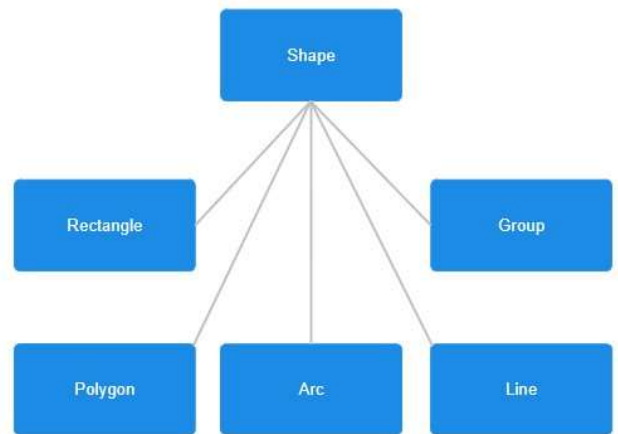
## 3. Архитектура на проекта:

### 3.1. Структура на класовете и предназначението им

Един от основните класове в проекта е класът System. Той служи за мост между потребителя и приложението. Основната цел на този клас е да осигури интерфейса на конзолната апликация и да обработва потребителските заявки. Негова грижа е правилното разбиване на дадена заявка съответно на команда и нейните параметри и подготвянето им за употреба в следващия етап от работа на приложението. Друга отговорност на класа е да обработва грешни заявки, като в отговор на потребителя се извежда подходящо съобщение, ако бъдат въведени такава.

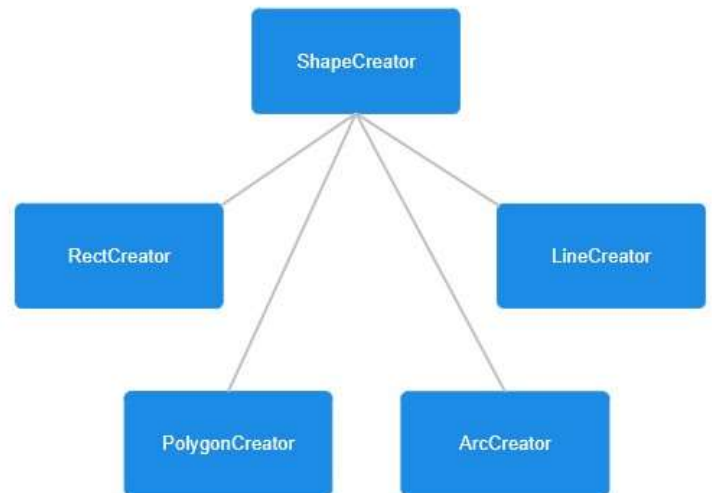
Друг основен клас е ObjectManager. Той е отговорен за съхраняването на обектите, заредени от файла, който е посочил потребителя. В него е реализирана функционалността на проекта, като се ползват подготвените данни от класа System, за да се изпълнят коректно потребителските заявки върху фигурите.

Фигурите представляват йерархия от класове, чийто базов клас е класът Shape. Той задава интерфейс, който наследниците Line, Arc, Rectangle, Polygon и Group реализират, за да се осъществи функционалността им по време на работа с приложението. Във фигура 1 нагледно се вижда структурата на йерархията Shapes.



Фигура 1

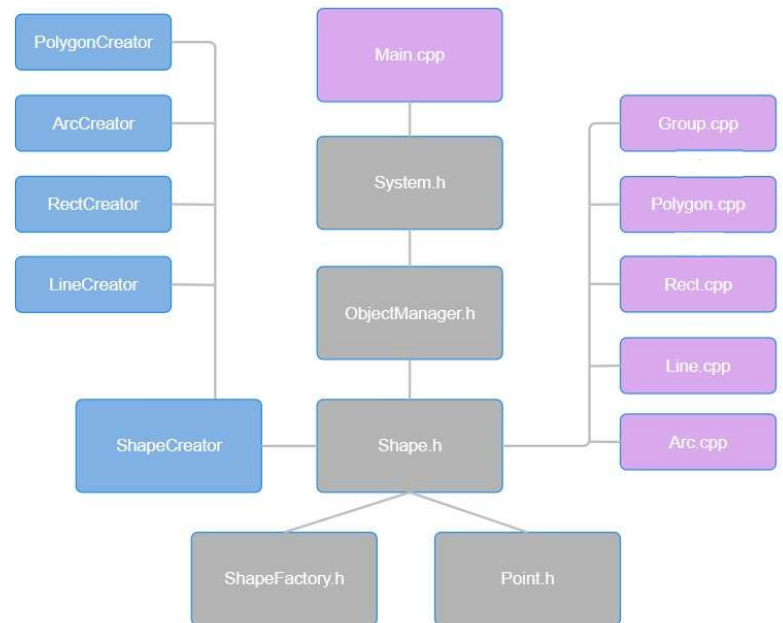
За създаването на фигури в приложението се грижи реализираният шаблон за проектиране “Абстрактна фабрика”. По този начин отговорността за създаване на валидни обекти е поставена върху йерархията фабрики. Това улеснява поддръжката на кода, бивайки отделен от йерархията Shapes и предоставя удобство в добавянето на нова логика за бъдещо разширение на апликацията. На фигура 2 е представена йерархията на класовете фабрики.



Фигура 2

### 3.2. Структура на проекта:

На фигура 3 е представена диаграма на основните части от структурата на проекта. В лилав цвят са изобразени .cpp файловете, в сив цвят съответно .h файловете, а със синьо е показана връзката на абстрактната фабрика в апликацията. Помощни библиотеки за реализацията на проекта са `<iostream>`, `<fstream>`, `<string>`, `<stdexcept>`, `<vector>`, `<cstring>`. В основата на структурата е разделянето на кода на смислови части, като всеки клас отговаря за изпълнението на конкретна задача и по този начин се получава по-лесен за поддръжка и по-качествен краен продукт.



Фигура 3

### 4. Тестване на приложението

В следните два примера е показана примерна работа с приложението

1:

```
open fig1.svg
```

```
print
```

```
translate vertical=100 horizontal=200
```

```
group 0 4000 4000 0
```

```
print
```

```
scale 5 2 2
```

```
extract 13 5
```

```
insert 13 1
```

```
print
```

```
ungroup 13
```

```
save
```

```
exit
```

```
2:
```

```
help
```

```
open <path>\fig2.txt
```

```
create line 0 0 200 200 red
```

```
hide 8
```

```
print
```

```
scale 2 2
```

```
show 8
```

```
print
```

```
save
```

```
exit
```

## 5. Потенциални разширения на проекта:

- Разпознаване на нови фигури
- Промяна на цвят на фигура – контур и/или запълване
- Ротация на елементи