# VU Software Engineering 2 DEAD

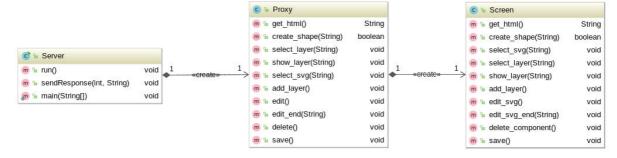
## Personal data:

First name, Surname:	Klaus Bareis 01501513 Fabian Schmon 01568351 Margaryta Simkina 01446530
Date:	Januar 2019

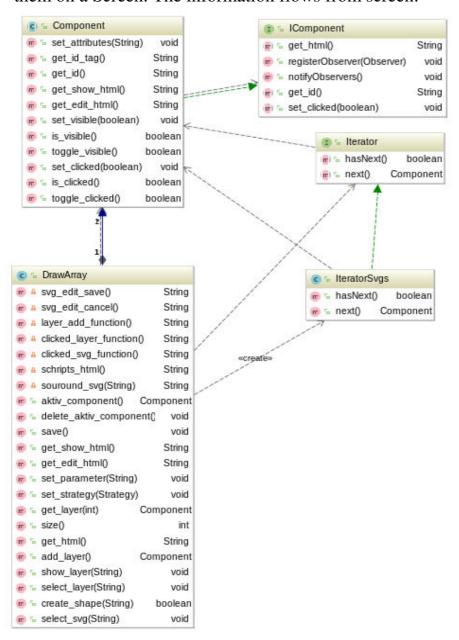
# **Design patterns**

## Proxy Pattern

The proxy pattern was meant to control the client's access to the screen. In normal operation, the contents should be transferred as html, when saving as svg. Saving was moved to the server. For this reason, the pattern is only partially implemented.

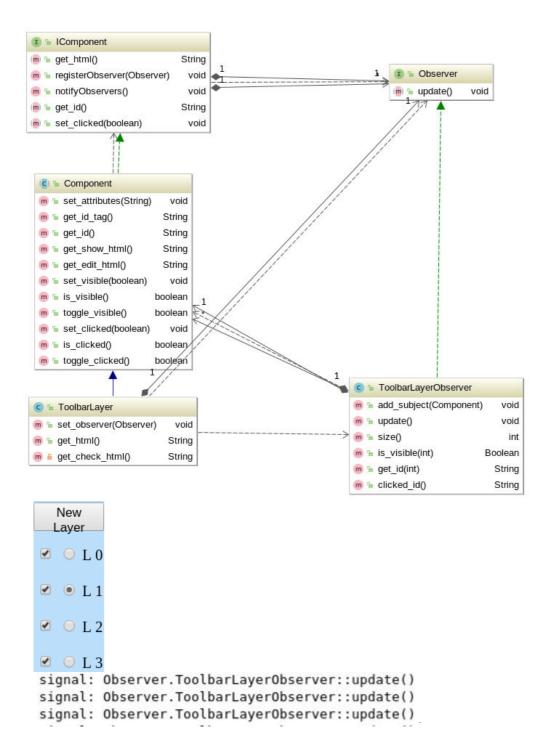


• **Iterator Pattern** We use Iterator to access objects, and navigate through them on a Screen. The information flows from screen.



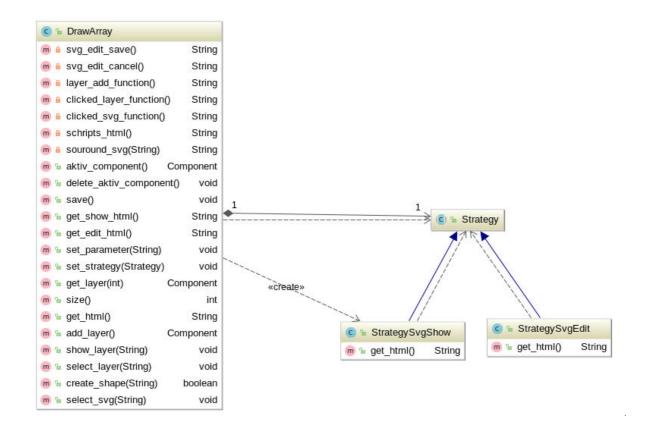
#### Observer Pattern

One use of the observer in our application is that of ToolbarLayerObserver. In the toolbar there is a button to create new layer in/on the screen. A click on this button informs the screen that he should create a new layer. Also, enable and disable, and switching between layers via this signal/slot concept is monitored.



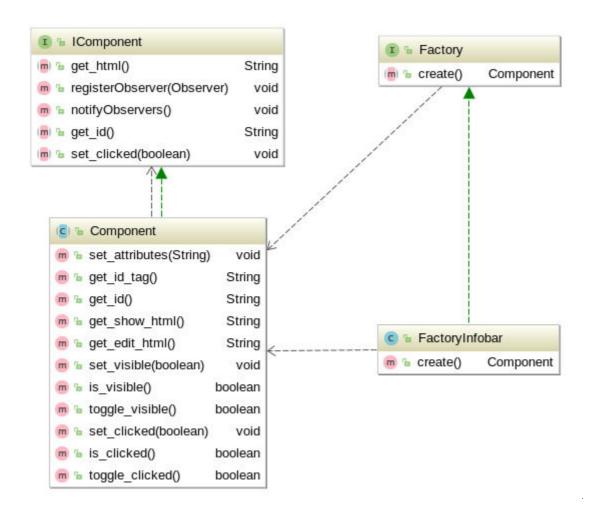
## · Strategy Pattern

This sequence diagram illustrates the implementation of the strategy pattern. The "Context" consists of an "Component" and an "Operation". Operation consists of "OperationEdit" and "OperationMove". Depending on the chosen strategy set by the "Observer", the incoming data will be handled in different ways.



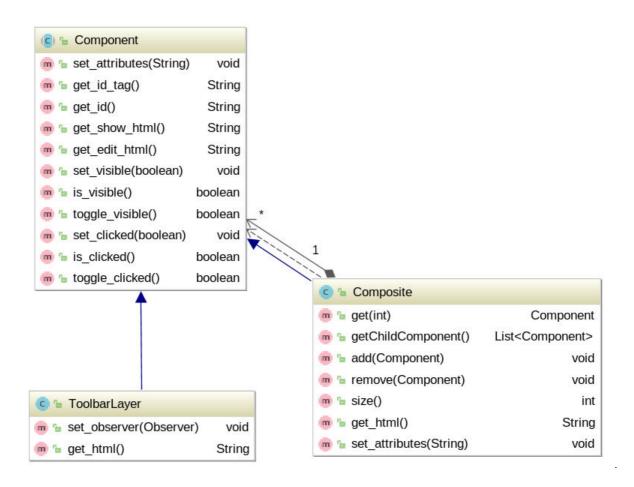
## Factory Method Pattern

This diagram shows the information flow when different UI elements are created. The "FactoryUI" makes use of the abstract factory pattern, the use of the illustrated interfaces enables easy handling of all UI parts.

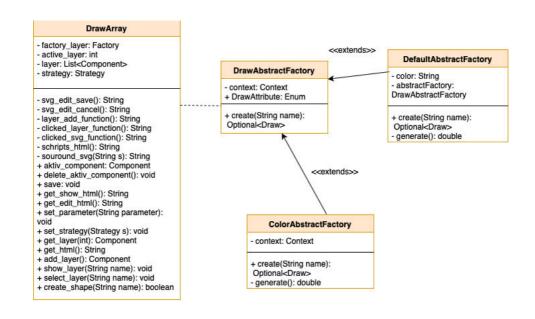


## • Composite Pattern

The structure of the "Component" class reflects the composite pattern. One "Component" consists of multiple screen elements which can consist of multiple components as well. This provides the ability to easily perform an action on multiple composites of screen elements.



## • Abstract Factory Pattern



Our AbstractoryFactory is initially responsible for creating multiple shapes with same attribute – in our case we chose color as an attribute

which is to be given in a menu, and after that color is basically switched, we are creating shapes with the same color until we turn on the next color basically. In that perspective, we use *ColorAbstractFactory(params)* for specific "family of shapes" based on one factor together. Otherwise we use *DefaultAbstractFactory()*. In both ways, further on, we will create other factories inside of those factories, and Color one will still call ColorDraw with necessary parameters through Optional.

## Dependency inversion principle

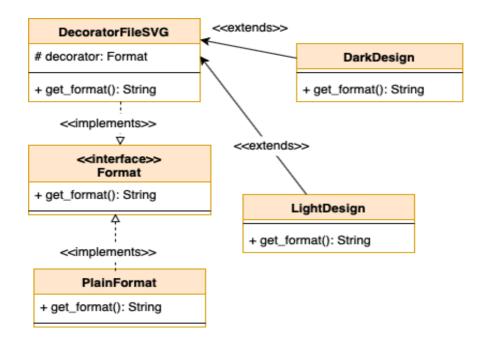


We decided as well to add dependency inversion Principle as a very good practice among current approaches: this enables way of decoupling software modules, getting rid of strong dependencies and being able to "reach out" from different level to another one. In our case we used BeanContainer which stores Context object inside. In that way, we are able to use our context object (which, for instance, we used a lot for creating/editing shapes), we would be able to use Context Object from any part of a project basically.

#### Decorator Patter

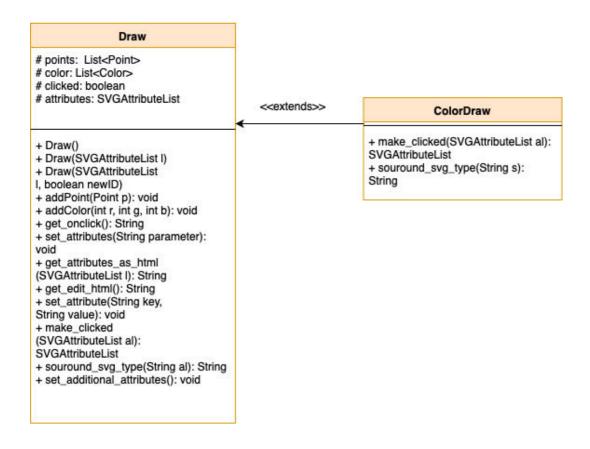
In our project, we decided to implement two different Decorator Patterns because it is considered to be extremely good practice if reading some news in this area, it is on a way to replace inheritance.

**First decorator Pattern** is a change of color of our scheme: on a Toolbar left, one can change it from light to dark and back without adding additional dependencies and implementing unnecessary methods.



**Second decorator Pattern** is position handler: we got a default one by simply clicking on a button, but also a Decorator one, when one gives parameters to create it in a. certain place. This wasn't done with extension, but with external additional functionality in design pattern.

Further, when function will be called, it will still go through a default constructor, but code itself won't be copied.



# **Coding practices**

#### 1. Readability & Clearness.

We always tried write code that simple to read and which will be understandable for developers, but still a bit of optimization was necessary at the end.

#### 2. Architecture first.

We approached first the architecture and a Diagram, and then made an attempt to build the whole project on our patterns and answering the planned structure, only a little bit of changes were then done.

### 3. Simple.

We tried to keep it simple and self-explanatory, using names, that are speaking for theirs functionality.

#### 4. Comment.

In general, we avoided unnecessary comments and also tried to make it already clear for others. But still, for some functions, that were not that easy, we added necessary comments.

#### 5. Reviews.

What I found important, we were giving each other reviews and if necessary, corrected mistakes, bugs of each other, which helped a lot.

#### 6. No deep nesting.

We tried to use as little coupling as possible, but as much, as necessary.

#### 7. Structured and short.

We used limited line and class length, separated classes in packages and followed the structure.

# **Defensive programming**

In a project there are some Exceptions or Try Catch blocks, but rather we could have extend it to a bigger coverage to avoid crushs. We tried to do basic checks for input variables though.

## **Code metrics**

## 1. Code Metrics General Analysis

Analysis of SE2\_Shapes
General Information

Total lines of code: 1317

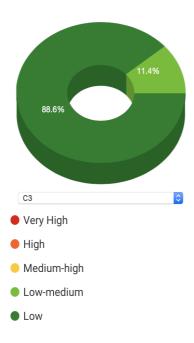
Number of classes: 67

Number of packages: 15

Number of external packages: 1

Number of problematic classes: 0

Number of highly problematic classes: 0

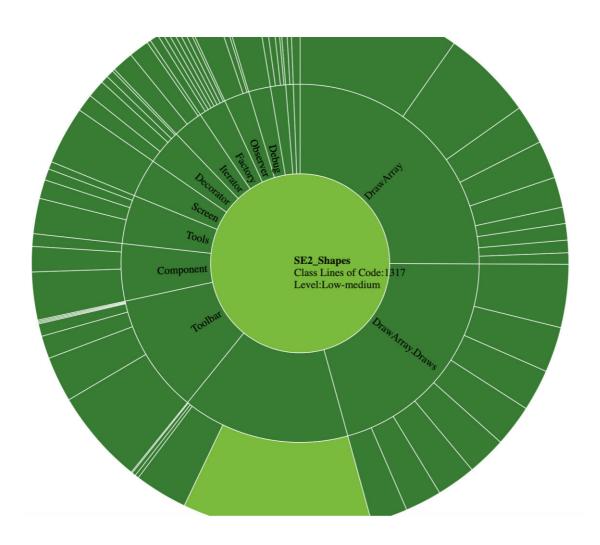


## 2. Distribution of Quality Attributes

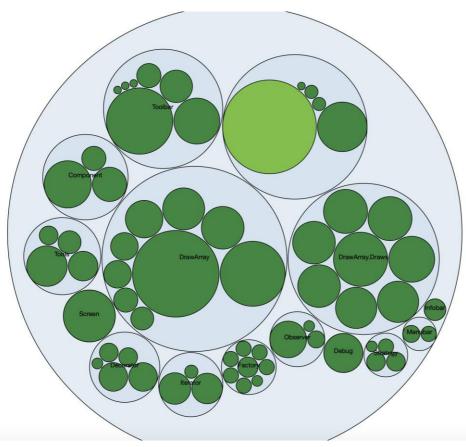
Complexity, Coupling, Cohesion, and Size



## 3. Metric Values in Sunburst Chart



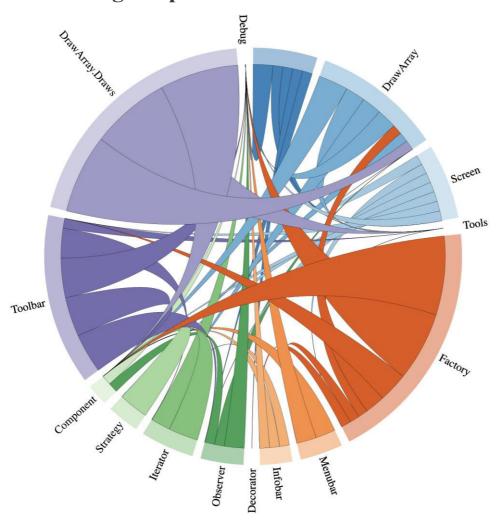
# 4. Metric Values by Packages



# 5. Metric Values in Treemap Chart



# 6. Package Dependencies



# **Team contribution**

Design: Bareis

Basic Implementation: Bareis

**Functional Requirements (FRs)** 

FR1

lines Bareis
circles Bareis
ellipses Bareis
triangles Bareis
quadrangles Bareis
n-gons Bareis
stars Bareis
text Bareis

#### FR2

• addition Bareis. Simkina

deletion Schmonediting Bareis

• movement (Bareis over editing)

#### FR<sub>3</sub>

**Bareis** 

#### FR4

Schmon

#### **Quality Requirements (QRs)**

**QR1-QR2:** Each team member was responsible for creating readable and self explainatory code, otherwise it should have been properly commented

Q3: We tried to avoid Code Grouping, limit line and class length/size, huge hierarchy and consistent indentation. Still, structure needed a *refactoring* which we implemented by coupling, creating new classes and dividing functionality into smaller pieces, but not dependent on each as in *inheritance*. We reduced switches size by some returns. Which did not work 100%, we still have big dependencies but that way we avoided too long and repetitive code. What we tried to organize well, was folder structure and proper classes and variables names.

**Q4:** We used defensive programming such as checking *input function parameters*, having a *proxy* which took care of further UI mistakes, as well, we used *optional* in order to avoid null returns.

**Q5:** We set up all necessary application type, concerns, technologies and so in a very beginning. Only things were added later by each team member were quality attributes. Our crosscutting concerns were Instrumentation and Logging (through a debug, we could also watch out threading).

**QR6:** We implemented few tests for logic (Composite, Component, Factory), as well as a basic test creation.

#### **QR7:**

Barcis, Schmon, Simkina  Barcis, Schmon, Simkina  Barcis, Schmon, Simkina  Barcis, Schmon, Simkina  Barcis, Schmon  Barcis, Schmon  Barcis, Schmon  Draws  Barcis, Schmon  Observer Pattern  Strategy SvgEdit  Strategy SvgEdit  Barcis  Strategy SvgEdit  Barcis  Iterator Pattern  Proxy for HTML  Abstract Factory Pattern  FactoryInfobar  FactoryInfobar  FactoryLayer  FactoryMethod Pattern  FactoryLayer  FactoryLayer  FactoryLayer  FactoryLayer  FactoryLayer  FactoryLayer  FactoryLayer  FactoryDrawArray  FactoryInolbarLayer  FactoryToolbarLayer  FactoryToolbarLayer  FactoryToolbarLayer  FactoryToolbarLayer  FactoryToolbarLayer  FactoryToolbarLayer  FactoryToolbarLayer  FactoryToolbarLayer  FactoryToolbarLayer  FactoryToolbarDepration  Decorator Pattern  Decorator Pattern  OlorAbstractFactory  Decorator Pattern  Decorator Pattern  Decorator Pattern  Decorator Pattern  OlorAbstractFactory  DefaultAbstract  Factory Simkina  DefaultAbstractFactory  Simkina  Bareis  FactoryLayer  Bareis  FactoryToolbarLayer  Bareis  FactoryToolbarLayer  Bareis  FactoryToolbarLayer  Bareis  FactoryToolbarDepration  Bareis  FactoryToolbarDepration  Bareis  FactoryToolbarOperation  Bareis  FactoryToolbarDepration  Bareis  FactoryToo	• Basic files	Server		Bareis,	
Bareis, Schmon, Simkina  Bareis, Schmon, Simkina  ColbarDperation  Bareis, Schmon  Bareis, Schmon  Bareis, Schmon  Bareis, Schmon  Observer Pattern  StrategySvgEdit  Iterator Pattern  Proxy Pattern  Proxy Pattern  Proxy Pattern  Proxy Factory Pattern  FactoryInfobar  Bareis  FactoryInfobar  B	Schmon, Simkina	т	'aalhar		
Bareis, Schmon, Simkina  Schmon, Simkina  Bareis, Schmon  Bareis, Schmon  Bareis, Schmon  Observer Pattern  StrategySvgEdit  StrategySvgShow  Bareis  Iterator Pattern  Iterator Pattern  Proxy  Proxy  Pattern  Proxy  Pattern  Proxy  Pattern  PactoryToolbar Pattern  FactoryToolbar Pattern  Pactory Poolbar Pattern  Pactory Poolbar Pareis  Pactory Pattern  Pactory Pattern  Pormat  LightDesign  Schmon  LightDesign  Schmon  Pattery  Porway  Simkina  Pactory Pattern  Porway  Simkina  Pactory Pattern  Porway  Simkina  Paroxy  Proxy  Proxy  Proxy  Simkina  Simkina  Simkina  Simkina	Bareis Schmon Simkina	1	oordar		
Bareis, Schmon, Simkina  Schmon, Simkina  Bareis, Schmon  Draws  Bareis, Schmon  Observer Pattern Strategy Pattern Strategy SygEdit Iterator Pattern Iterator Layer (not used) Iterator Sygs Bareis  Composite Pattern Proxy Pattern Proxy Pattern OlosabstractFactory ObefaultAbstractFactory Factory Method Pattern Factory Method Pattern Factory Toolbar Factory Decorator Pattern Powart Factory Decorator Pattern Powart Factory Decorator Pattern Powart Factory Decorator Pattern Powart Factory DefaultAbstractFactory Simkina Pareis Factory Method Pattern Fact	Barcis, Schillon, Shirkina	Toolbarl aver			
Schmon, Simkina  Schmon, Simkina  Bareis, Schmon  Bareis, Schmon  Observer Pattern  Strategy Pattern  Strategy SvgEdit  Iterator Pattern  Iterator Pattern  Ocomposite Pattern  Proxy Pattern  Oclor Proxy Pattern  Oclor Proxy Pattern  Color Draw  Color  DrawAbstractFactory  DefaultAbstractFactory  Factory Method Pattern  Factory Joobbar  Factory Toolbar  Proxy Pattern  Factory Degration  Factory Degration  Factory Simkina  Draw Abstract Factory  Simkina  DefaultAbstractFactory  Simkina  DefaultAbstractFactory  Simkina  Factory Method Pattern  Factory Joobar  Factory Joobar  Factory Joobar  Factory Toolbar  Factory Toolbar  Factory Toolbar  Factory Schmon  Decorator Pattern  Format  Decorator Pattern  Decorator Pattern  Ocolor Draw  Factory Schmon  Dark Design  Dark Design  Schmon  Darw Array  Simkina  Abstract Factory Pattern  Color Draw  Simkina  Schmon  Schmon  Decorator Pattern  Ocolor Draw  Simkina  Schmon  Schmon  Decorator Pattern  Format  Schmon  Darw Array  Simkina  Abstract Factory Pattern  Color Draw  Simkina  Factory Poxy  Simkina  Schmon  Darw Array  Simkina  Schmon  Darw Array  Simkina  Daw Array  Simkina  Paway Simkina  Daw Abstract Factory  Simkina  Schmon  Decorator Pottern  Ocolor Abstract Factory  Simkina  Schmon  Depaw Abstract Factory  Simkina  Schmon  Depaw Abstract Factory  Simkina  Schmon  Darw Abstract Factory  Simkina  Schmon  Schm	Bareis, Schmon, Simkina	1	OOTOUT LU	y C1	
Schmon, Simkina  Bareis, Schmon  Observer Pattern Strategy Pattern Strategy Pattern Iterator Pattern Iterator Pattern Proxy Pattern Ookorous Pattern Iterator Sygs Iterator Sygs Composite Pattern Proxy Pattern Proxy Pattern Ookor Pattern Proxy Pattern Ookor Pattern Proxy Pattern Ookor Pattern Proxy Pattern Proxy Pattern Ookor Proxy Pattern Ookor Proxy Pattern Proxy Pattern Ookor Proxy Pattern Ookor Proxy Pattern Proxy Pattern Ookor Proxy Pattern Ookor Proxy Pattern Ookor Proxy Pattern Ookor Proxy Pattern Proxy Pattern Ookor Proxy Pattern Ookor Proxy Pattern Ookor Proxy Pattern Ookor Proxy Proxy Pattern Ookor Proxy Proxy Proxy Proxy Ookor Proxy Proxy Ookor Proxy Proxy Ookor Proxy Ook	<b>2 2</b> 13, 2	ToolbarOperation		Bareis.	
Bareis, Schmon Observer Pattern Observer Pattern Strategy Pattern Strategy Pattern Iterator Pattern Ocomposite Pattern Proxy Pattern Officer Proxy Pattern Officer Proxy Pattern Officer Proxy Pattern Proxy Pattern Officer Proxy Pattern Proxy Pattern Ocolor Pattern Officer Pattern Ocolor Simkina Ocolor Oraw AbstractFactory Osimkina Ocolor OrawAbstractFactory Osimkina Ocolor OrawAbstractFactory Osimkina Ocolor OrawAbstractFactory Ocolor O	Schmon, Simkina		<b>.</b>		,
Bareis, Schmon  Observer Pattern StrategySvgEdit StrategySvgShow Bareis  Iterator Pattern IteratorLayer (not used) IteratorSvgs Bareis  Composite Pattern Proxy Pattern Proxy Pattern Proxy Pattern Proxy Pattern OlorabstractFactory Pattern Proxy Pattern OlorabstractFactory Pattern PractoryDrawArray FactoryDrawArray FactoryLayer FactoryDear FactoryToolbar FactoryToolbar PactoryToolbarOperation Pacroator Pattern Powarray Pocorator Pattern Pocorator Pattern Pocorator Pattern Pocorator Pattern Pormat Pocorator Pattern Pormat DecoratorFileSVG PlainFormat DecoratorFileSVG PlainFormat LightDesign ColorDraw Simkina Description Bareis FactoryJoolbar Bareis FactoryToolbar Bareis FactoryToolbar Bareis FactoryToolbarOperation ParioryToolbarOperation Pocorator Pattern Pormat DecoratorFileSVG Schmon DarkDesign Schmon DarkDesign Schmon DarkDesign Schmon DarkDesign Schmon DarwArray Simkina DrawArray Simkina DrawArray Simkina DrawAbstractFactory Simkina Simki		Menubar			
Bareis, Schmon  Observer Pattern Strategy SygEdit Bareis Strategy SygShow Bareis  Iterator Pattern IteratorLayer (not used) IteratorSygs Bareis  Composite Pattern Proxy Fattern Proxy Pattern Proxy Port HTML Abstract Factory Pattern OclorDraw ColorDraw DrawAbstractFactory FactoryIndobar FactoryLayer FactoryLayer FactoryToolbar FactoryToolbar FactoryToolbarQperation Pocorator Pattern Decorator Pattern Powarray DefaultAbstractFactory Decorator Pattern Decorator Pattern Decorator Pattern Decorator Pattern Decorator Pattern Pool Decorator Pattern Abstract Factory Pattern DefaultAbstractFactory Simkina DrawAbstractFactory Bareis FactoryLayer Bareis FactoryLayer Bareis FactoryToolbar Bareis FactoryToolbarLayer Bareis FactoryToolbarLayer Bareis FactoryToolbarDeperation Bareis PactoryToolbarOperation Decorator Pattern Decorator Pattern Decorator Pattern Decorator Pattern DarkDesign Schmon DarkDesign Schmon DarkDesign Schmon DarwArray Simkina DrawArray Simkina DrawArray Simkina DrawAbstractFactory Simkina	Bareis, Schmon				
<ul> <li>Observer Pattern</li> <li>Strategy Pattern</li> <li>StrategySvgEdit StrategySvgShow</li> <li>Bareis</li> <li>Iterator Pattern</li> <li>IteratorLayer (not used) Bareis</li> <li>IteratorSvgs Bareis</li> <li>Composite Pattern</li> <li>Proxy Pattern</li> <li>Proxy for HTML Bareis, Schmon</li> <li>Abstract Factory Pattern</li> <li>Color AbstractFactory Simkina Color Draw AbstractFactory Simkina</li> <li>Color Simkina DefaultAbstractFactory Simkina DefaultAbstractFactory Simkina</li> <li>Factory Method Pattern</li> <li>FactoryDrawArray Bareis FactoryLayer Bareis FactoryHolbar Bareis FactoryHolbar Bareis FactoryToolbar Bareis FactoryToolbar Bareis</li> <li>FactoryToolbar Decorator Pattern</li> <li>Decorator Pattern</li> <li>Decorator FileSVG Schmon DarkDesign Schmon</li> <li>Decorator Pattern 2 ColorDraw Simkina DrawArray Simkina</li> <li>Abstract Factory Pattern</li> <li>ColorDraw Simkina Schmon</li> <li>Decorator Pattern Schmon Schmon</li> <li>Decorator Pattern Schmon Schmon</li> <li>DarkDesign Schmon</li> <li>Abstract Factory Pattern</li> <li>ColorDraw Simkina DrawArray Simkina</li> <li>Abstract Factory Pattern</li> <li>DefaultAbstractFactory Simkina DrawArray Simkina DrawArray Simkina</li> <li>PorawAstractFactory Simkina DrawAstractFactory Simkina Simkina Proxy</li> <li>Dependency inversion</li> <li>BeanContainer</li> <li>Simkina Simkina Simkina Simkina Proxy</li> </ul>		Γ	<b>D</b> raws		
<ul> <li>Strategy Pattern         StrategySvgEdit         StrategySvgShow         Bareis         Iterator Pattern         IteratorLayer (not used)         IteratorSvgs         Bareis         IteratorSvgs         Bareis</li></ul>					
• Iterator Pattern  IteratorLayer (not used) IteratorSvgs Bareis  • Composite Pattern  • Proxy Pattern  • Proxy Pattern  • Abstract Factory Pattern  • Factory Method Pattern  • Pactory Method Pattern  • Decorator Pattern  • Decorator Pattern  • Abstract Factory Pattern  • Dependency inversion  • IteratorLayer (not used) IteratorSvgs Bareis Bareis Bareis Bareis Schmon  Simkina  Color Simkina Color Simkina  Color Simkina  Color Simkina  ParawAbstractFactory Simkina  Bareis Factory Simkina  Bareis FactoryInfobar Bareis FactoryLayer Bareis FactoryLayer Bareis FactoryToolbar Bareis FactoryToolbarLayer FactoryToolbarLayer Bareis FactoryToolbarCoperation  Bareis  • Decorator Pattern  • Schmon  DarkDesign  • Schmon  Simkina  • Abstract Factory Pattern  ColorAbstractFactory Simkina  Proxy  • Dependency inversion  BeanContainer  Simkina		•			
<ul> <li>Iterator Pattern IteratorLayer (not used)         IteratorSvgs Bareis</li> <li>Composite Pattern for Screen Bareis</li> <li>Proxy Pattern Proxy for HTML Bareis, Schmon</li> <li>Abstract Factory Pattern ColorAbstractFactory Simkina         Color Draw Simkina         Color Simkina         DrawAbstractFactory Simkina         DefaultAbstractFactory Simkina         Pactory Method Pattern FactoryDrawArray Bareis         FactoryLayer Bareis         FactoryToolbar Bareis         FactoryToolbar Bareis         FactoryToolbar Bareis         FactoryToolbarLayer Bareis         FactoryToolbarDepration Bareis         FactoryToolbarOperation Bareis         PactoryToolbarOperation Schmon         Decorator Pattern Schmon         Decorator Pattern Schmon         DarkDesign Schmon         OrawArray Simkina         OrawArray Simkina         OrawArray Simkina         OrawArray Simkina         OrawAstractFactory Simkina         OrawAbstractFactory Si</li></ul>	<ul> <li>Strategy Pattern</li> </ul>				
• Composite Pattern • Proxy Pattern • Proxy Pattern • Abstract Factory Pattern • Abstract Factory Pattern • Factory Method Pattern • Pactory Method Pattern • Pactory Method Pattern • Decorator Pattern • Decorator Pattern • Decorator Pattern • Abstract Factory Pattern • Factory Method Pattern • Format • Decorator Pattern • De		e. e			
<ul> <li>Composite Pattern</li> <li>Proxy Pattern</li> <li>Proxy for HTML</li> <li>Abstract Factory Pattern</li> <li>Color AbstractFactory</li> <li>Color Simkina</li> <li>Color Simkina</li> <li>Color Simkina</li> <li>DefaultAbstractFactory</li> <li>Factory Method Pattern</li> <li>FactoryDrawArray</li> <li>FactoryLayer</li> <li>FactoryMenubar</li> <li>FactoryToolbar</li> <li>FactoryToolbarDeration</li> <li>Pareis</li> <li>FactoryToolbarOperation</li> <li>Decorator Pattern</li> <li>Decorator Pattern 2</li> <li>Decorator Pattern</li> <li>OleorDraw</li> <li>DarwArray</li> <li>Simkina</li> <li>Bareis</li> <li>Bareis</li> <li>FactoryLayer</li> <li>Bareis</li> <li>FactoryToolbar Bareis</li> <li>FactoryToolbarOperation</li> <li>Bareis</li> <li>Schmon</li> <li>Decorator Pattern</li> <li>DecoratorFileSVG</li> <li>PlainFormat</li> <li>Schmon</li> <li>DarwDesign</li> <li>Schmon</li> <li>DarwDesign</li> <li>Schmon</li> <li>OrlorDraw</li> <li>Dimkina</li> <li>DrawArray</li> <li>Simkina</li> <li>DefaultAbstractFactory</li> <li>Simkina</li> <li>DefaultAbstractFactory</li> <li>Simkina</li> <li>DefaultAbstractFactory</li> <li>Simkina</li> <li>DrawAbstractFactory</li> <li>Simkina</li> <li>DefaultAbstractFactory</li> <li>Simkina</li> <li>DefaultAbstractFactory</li> <li>Simkina</li> <li>DefaultAbstractFactory</li> <li>Simkina</li> <li>DefaultAbstractFactory</li> <li>Simkina</li> <li>DefaultAbstractFactory</li> <li>Simkina</li> <li>Simkina</li></ul>	• Iterator Pattern	•	sed)		
<ul> <li>Proxy Pattern</li> <li>Abstract Factory Pattern</li> <li>ColorAbstractFactory</li> <li>Simkina</li> <li>Color Simkina</li> <li>Color Simkina</li> <li>Color Draw AbstractFactory</li> <li>DrawAbstractFactory</li> <li>Simkina</li> <li>Factory Method Pattern</li> <li>FactoryDefaultAbstractFactory</li> <li>FactoryDefaultAbstractFactory</li> <li>Simkina</li> <li>Factory Method Pattern</li> <li>FactoryDeawArray</li> <li>FactoryInfobar</li> <li>FactoryLayer</li> <li>Bareis</li> <li>FactoryMenubar</li> <li>FactoryToolbar</li> <li>Bareis</li> <li>FactoryToolbarLayer</li> <li>Bareis</li> <li>FactoryToolbarOperation</li> <li>Bareis</li> <li>FactoryToolbarOperation</li> <li>Bareis</li> <li>Schmon</li> <li>Decorator Pattern</li> <li>Format</li> <li>DecoratorFileSVG</li> <li>Schmon</li> <li>LightDesign</li> <li>Schmon</li> <li>DarkDesign</li> <li>Schmon</li> <li>DarkDesign</li> <li>Schmon</li> <li>OrawArray</li> <li>Simkina</li> <li>DrawArray</li> <li>Simkina</li> <li>DefaultAbstractFactory</li> <li>Simkina</li> <li>DrawAbstractFactory</li> <li>Simkina</li> <li>DrawAbstractFactory</li> <li>Simkina</li> <li>DrawAbstractFactory</li> <li>Simkina</li> <li>DrawAbstractFactory</li> <li>Simkina</li> <li>DrawAbstractFactory</li> <li>Simkina</li> </ul>		<u>e</u>			
• Abstract Factory Pattern ColorAbstractFactory Simkina Color DrawAbstractFactory DefaultAbstractFactory Factory Method Pattern FactoryInfobar FactoryLayer FactoryToolbar FactoryToolbarLayer FactoryToolbarOperation Perorator Pattern Format Decorator Pattern Decorator Pattern  DefaultAbstractFactory Simkina DrawAbstractFactor Simkina					
ColorDraw Color Simkina Color Simkina DrawAbstractFactory DefaultAbstractFactory Simkina ObefaultAbstractFactory Simkina  Factory Method Pattern FactoryDrawArray FactoryInfobar FactoryLayer FactoryMenubar FactoryToolbar FactoryToolbarLayer FactoryToolbarLayer FactoryToolbarOperation FactoryToolbarOperation Decorator Pattern Format DecoratorFileSVG PlainFormat LightDesign DarkDesign Schmon DarkDesign Schmon DarkDesign Schmon DarwArray Simkina OrawArray Simkina OrawArray Simkina DrawArray Simkina DrawArray Simkina DrawAbstractFactory DefaultAbstractFactory Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina Schmon DrawAbstractFactory Simkina DrawAbstractFactory Simkina	5				
• Factory Method Pattern • Factory Menubar • Bareis • Factory Menubar • Bareis • Factory Toolbar Deration • Bareis • Factory Toolbar Operation • Bareis • Factory Toolbar Operation • Decorator Pattern • Format • Decorator Pattern • Format • Decorator File SVG • Schmon • Decorator Pattern • Decorator Pattern • Color Draw • Dark Design • Dark Design • Color Draw • Draw Array • Simkina • Abstract Factory Pattern • Color Abstract Factory • Default Abstract Factor • Draw Abstract Factor • Draw Abstract Factor • Draw Abstract Factor • Simkina • Proxy • Dependency inversion • Bean Container • Simkina	• Abstract Factory Pattern		ry		
• Factory Method Pattern • Factory Method Pattern • Factory Infobar • Factory Infobate • Factor					
• Factory Method Pattern FactoryDrawArray FactoryInfobar FactoryLayer FactoryMenubar FactoryToolbar FactoryToolbar FactoryToolbarDperation Poecorator Pattern Format DecoratorFileSVG PlainFormat LightDesign DarkDesign DarkDesign FactoryPattern PorawArray FactoryPattern ColorAbstractFactory Factory Simkina DefaultAbstractFactory Simkina DefaultAbstractFactory Simkina DrawAbstractFactory Simkina Proxy Simkina					
• Factory Method Pattern FactoryDrawArray FactoryInfobar FactoryLayer FactoryMenubar FactoryToolbar FactoryToolbarLayer FactoryToolbarOperation Bareis Schmon DecoratorFileSVG Schmon PlainFormat Schmon LightDesign Schmon DarkDesign Schmon DarkDesign Schmon OrawArray Simkina DrawArray Simkina DrawArray Simkina DefaultAbstractFactory DefaultAbstractFactory Simkina DrawAbstractFactory Simkina Proxy  • Dependency inversion BeanContainer Simkina		· · · · · · · · · · · · · · · · · · ·			
FactoryInfobar Bareis FactoryLayer Bareis FactoryMenubar Bareis FactoryToolbar Bareis FactoryToolbarLayer Bareis FactoryToolbarLayer Bareis FactoryToolbarOperation Bareis  • Decorator Pattern Format Schmon DecoratorFileSVG Schmon PlainFormat Schmon LightDesign Schmon DarkDesign Schmon  • Decorator Pattern 2 ColorDraw Simkina DrawArray Simkina • Abstract Factory Pattern DefaultAbstractFactory Simkina DrawAbstractFactory Simkina Proxy • Dependency inversion BeanContainer Simkina		•			
FactoryLayer Bareis FactoryMenubar Bareis FactoryToolbar Bareis FactoryToolbarLayer Bareis FactoryToolbarDeration Bareis  • Decorator Pattern Format Schmon DecoratorFileSVG Schmon PlainFormat Schmon LightDesign Schmon DarkDesign Schmon  • Decorator Pattern 2 ColorDraw Simkina DrawArray Simkina • Abstract Factory Pattern ColorAbstractFactory Simkina DrawAbstractFactory Simkina Proxy • Dependency inversion BeanContainer Simkina	Factory Method Pattern	· ·			
FactoryMenubar Bareis FactoryToolbar Bareis FactoryToolbarLayer Bareis FactoryToolbarOperation Bareis  • Decorator Pattern Format Schmon DecoratorFileSVG Schmon PlainFormat Schmon LightDesign Schmon DarkDesign Schmon  • Decorator Pattern 2 ColorDraw Simkina DrawArray Simkina  • Abstract Factory Pattern ColorAbstractFactory Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina Simkina Simkina Simkina					
FactoryToolbar Bareis FactoryToolbarLayer Bareis FactoryToolbarOperation Bareis  • Decorator Pattern Format Schmon DecoratorFileSVG Schmon PlainFormat Schmon LightDesign Schmon DarkDesign Schmon  • Decorator Pattern 2 ColorDraw Simkina DrawArray Simkina  • Abstract Factory Pattern ColorAbstractFactory Simkina DefaultAbstractFactory Simkina DrawAbstractFactory Simkina Simkina		<u> </u>			
FactoryToolbarLayer Bareis FactoryToolbarOperation Bareis  • Decorator Pattern Format Schmon DecoratorFileSVG Schmon PlainFormat Schmon LightDesign Schmon DarkDesign Schmon  • Decorator Pattern 2 ColorDraw Simkina DrawArray Simkina  • Abstract Factory Pattern DefaultAbstractFactory Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina Simkina Proxy  • Dependency inversion BeanContainer Simkina		FactoryToolbar			
• Decorator Pattern Format DecoratorFileSVG PlainFormat LightDesign DarkDesign OrawArray Simkina DrawArray Simkina DefaultAbstractFactory DefaultAbstractFactory Proxy • Dependency inversion  FactoryToolbarOperation Bareis Schmon Schmon Schmon Schmon Schmon Simkina					
<ul> <li>Decorator Pattern</li> <li>Format</li> <li>DecoratorFileSVG</li> <li>PlainFormat</li> <li>LightDesign</li> <li>DarkDesign</li> <li>DarkDesign</li> <li>Schmon</li> <li>DarkDesign</li> <li>Schmon</li> <li>Double Schmon</li> <li>ColorDraw</li> <li>DrawArray</li> <li>Simkina</li> <li>DrawArray</li> <li>Simkina</li> <li>DefaultAbstractFactory</li> <li>DefaultAbstractFactor</li> <li>DrawAbstractFactory</li> <li>Simkina</li> <li>DrawAbstractFactory</li> <li>Simkina</li> <li>DrawAbstractFactory</li> <li>Simkina</li> <li>DrawAbstractFactory</li> <li>Simkina</li> <li>Simkina</li> <li>Proxy</li> <li>Dependency inversion</li> <li>BeanContainer</li> <li>Simkina</li> </ul>		•			
DecoratorFileSVG Schmon PlainFormat Schmon LightDesign Schmon DarkDesign Schmon  • Decorator Pattern 2 ColorDraw Simkina DrawArray Simkina • Abstract Factory Pattern ColorAbstractFactory Simkina DefaultAbstractFactor Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina Simkina Proxy • Dependency inversion BeanContainer Simkina	Decorator Pattern	•			
PlainFormat Schmon LightDesign Schmon DarkDesign Schmon  • Decorator Pattern 2 ColorDraw Simkina DrawArray Simkina  • Abstract Factory Pattern ColorAbstractFactory Simkina DefaultAbstractFactory Simkina DefaultAbstractFactory Simkina Proxy  • Dependency inversion BeanContainer Simkina	Decorator rattern				
LightDesign Schmon DarkDesign Schmon  • Decorator Pattern 2 ColorDraw Simkina DrawArray Simkina  • Abstract Factory Pattern ColorAbstractFactory Simkina DefaultAbstractFactory Simkina DrawAbstractFactory Simkina DrawAbstractFactory Simkina Proxy  • Dependency inversion BeanContainer Simkina					
DarkDesign Schmon  Output  Decorator Pattern 2 ColorDraw Simkina  DrawArray Simkina  Output  ColorAbstractFactory Simkina  DefaultAbstractFactor Simkina  DrawAbstractFactory Simkina  DrawAbstractFactory Simkina  DrawAbstractFactory Simkina  DrawAbstractFactory Simkina  Proxy  Dependency inversion BeanContainer Simkina					
<ul> <li>Decorator Pattern 2 ColorDraw Simkina         <ul> <li>DrawArray Simkina</li> </ul> </li> <li>Abstract Factory Pattern ColorAbstractFactory Simkina             <ul> <li>DefaultAbstractFactor Simkina</li> <li>DrawAbstractFactory Simkina</li> <li>Proxy</li> </ul> </li> <li>Dependency inversion BeanContainer Simkina</li> </ul>					
• Abstract Factory Pattern  • Abstract Factory Pattern  • ColorAbstractFactory DefaultAbstractFactor DrawAbstractFactory Simkina DrawAbstractFactory Simkina Proxy  • Dependency inversion  • Dependency inversion  • Simkina	• Decorator Pattern 2	C			
<ul> <li>Abstract Factory Pattern ColorAbstractFactory DefaultAbstractFactor Simkina DrawAbstractFactory Simkina DrawAbstractFactory Proxy</li> <li>Dependency inversion BeanContainer Simkina</li> </ul>	_				
DefaultAbstractFactor Simkina DrawAbstractFactory Simkina Proxy  • Dependency inversion BeanContainer Simkina	<del></del>		ry		
Proxy • Dependency inversion BeanContainer Simkina	•		•	Simkina	
Proxy • Dependency inversion BeanContainer Simkina				Simkina	
• Dependency inversion BeanContainer Simkina			-		
container Context Simkina	<ul> <li>Dependency inversion</li> </ul>	BeanContainer		Simkina	
	container	Context		Simkina	

**QR8:** We have implemented the whole project using Maven. **QR9:** In a package Runnable we got our JAR file.

# HowTo

Application can be launched by running a JAR in a terminal which is  $\verb"java"$  -  $\verb"jar"$  Shapes.

Otherwise one can /implementation/src/main/Server.java "Run" -> "as Application" and then Application runs on port :8080. After that browser should be open on localhost.