

VU Software Engineering 2

DEAD

Personal data:

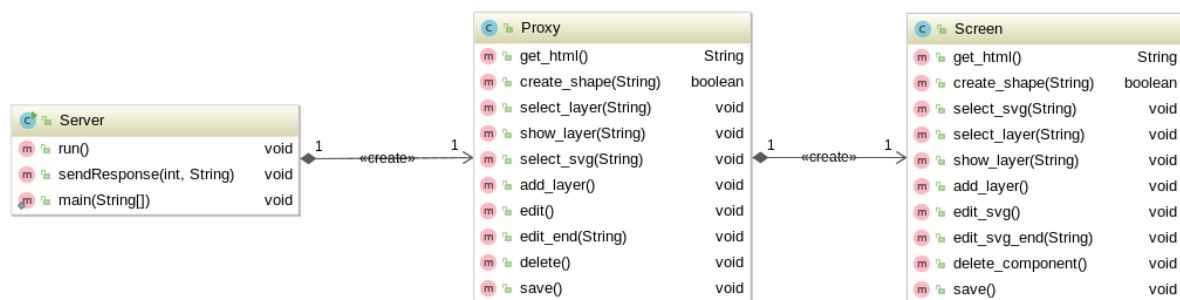
First name, Surname:	Klaus Bareis 01501513 Fabian Schmon 01568351 Margaryta Simkina 01446530
Date:	Januar 2019

Design patterns

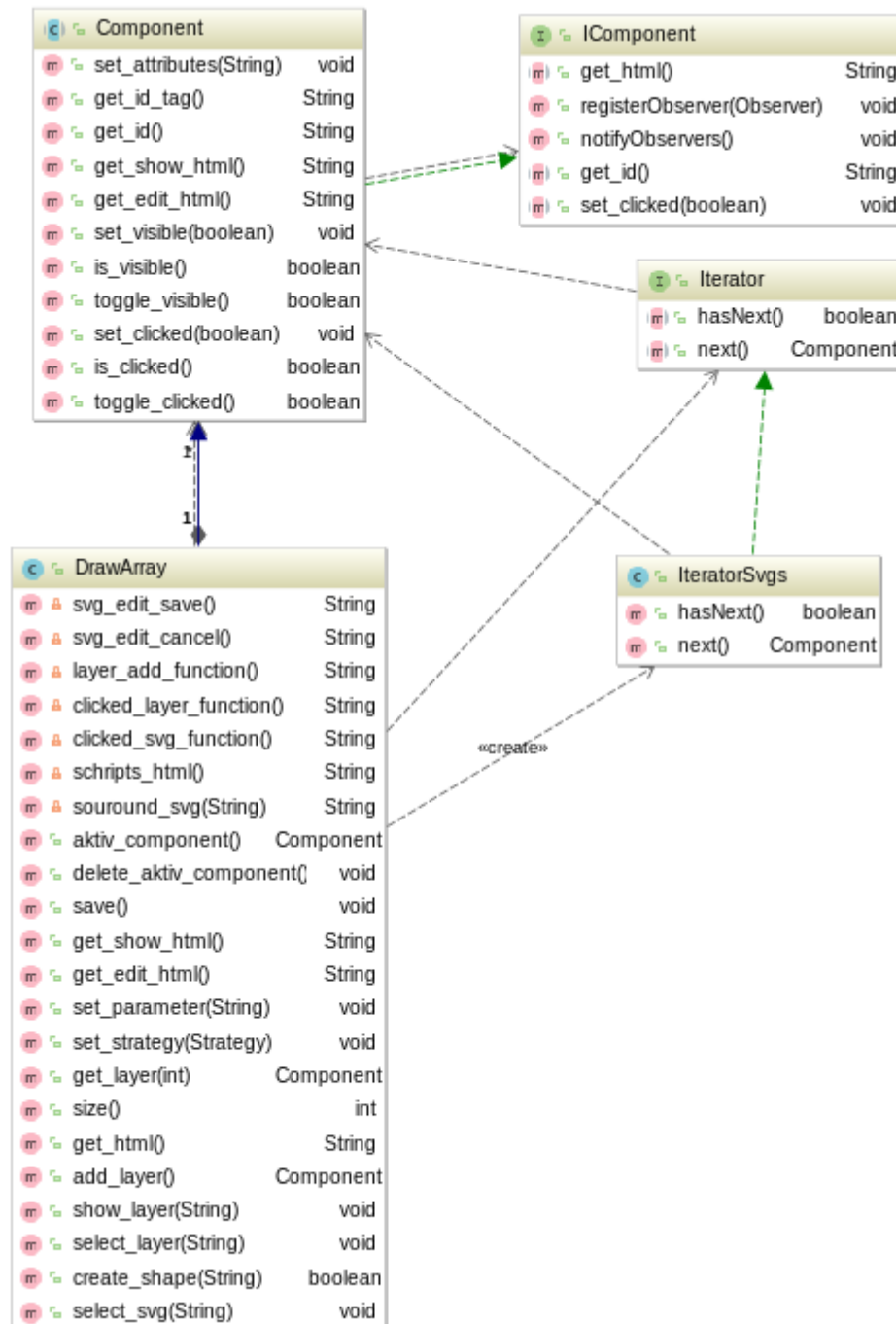
discussing where the required design patterns occur in your solution. Discuss at least one occurrence for each required design pattern in the code in detail. Support the decision to use a specific pattern with compelling arguments. Create UML diagrams to illustrate how the patterns have been applied.

• 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.

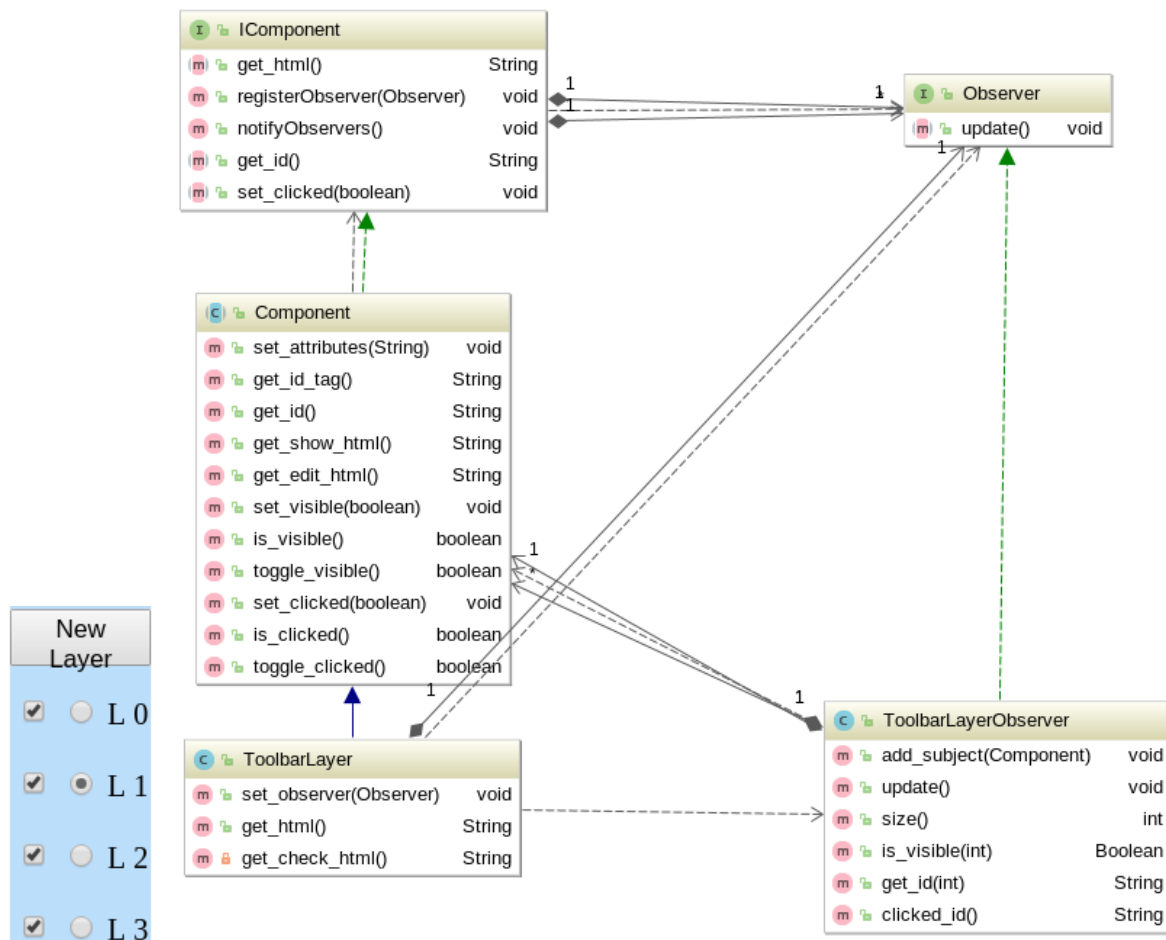


• Iterator Pattern



• 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.

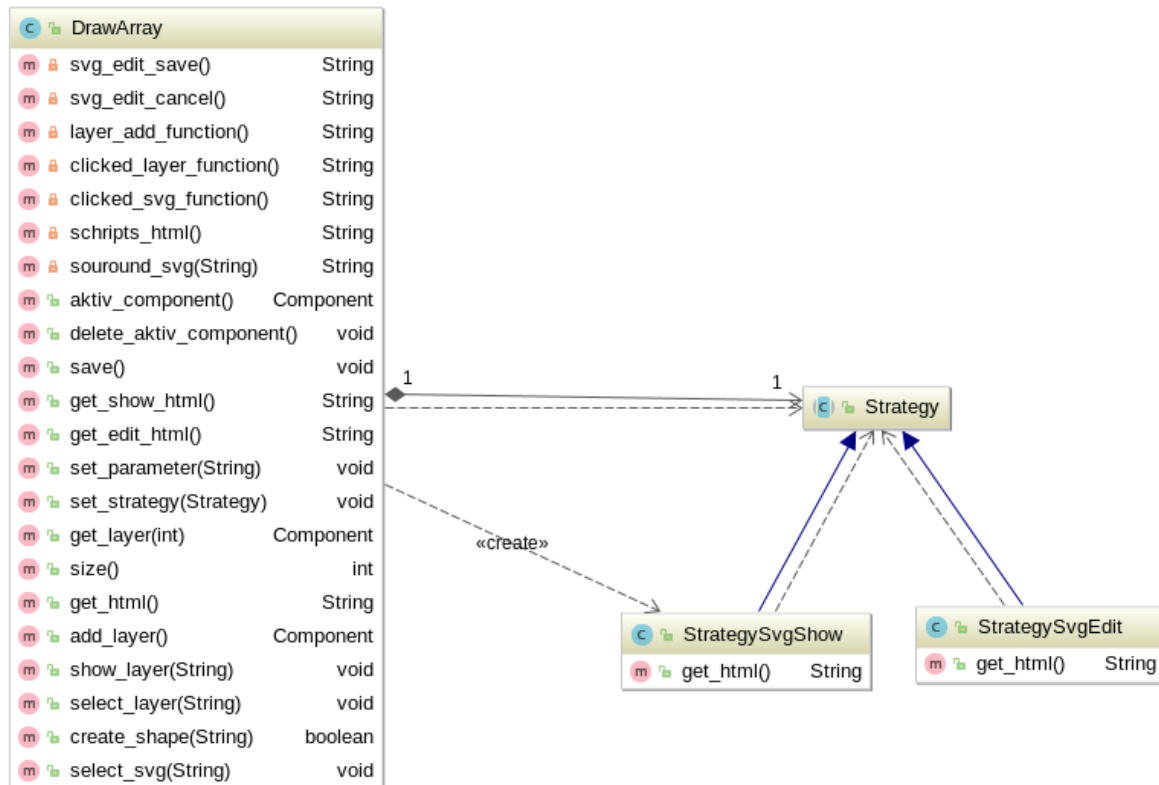


```

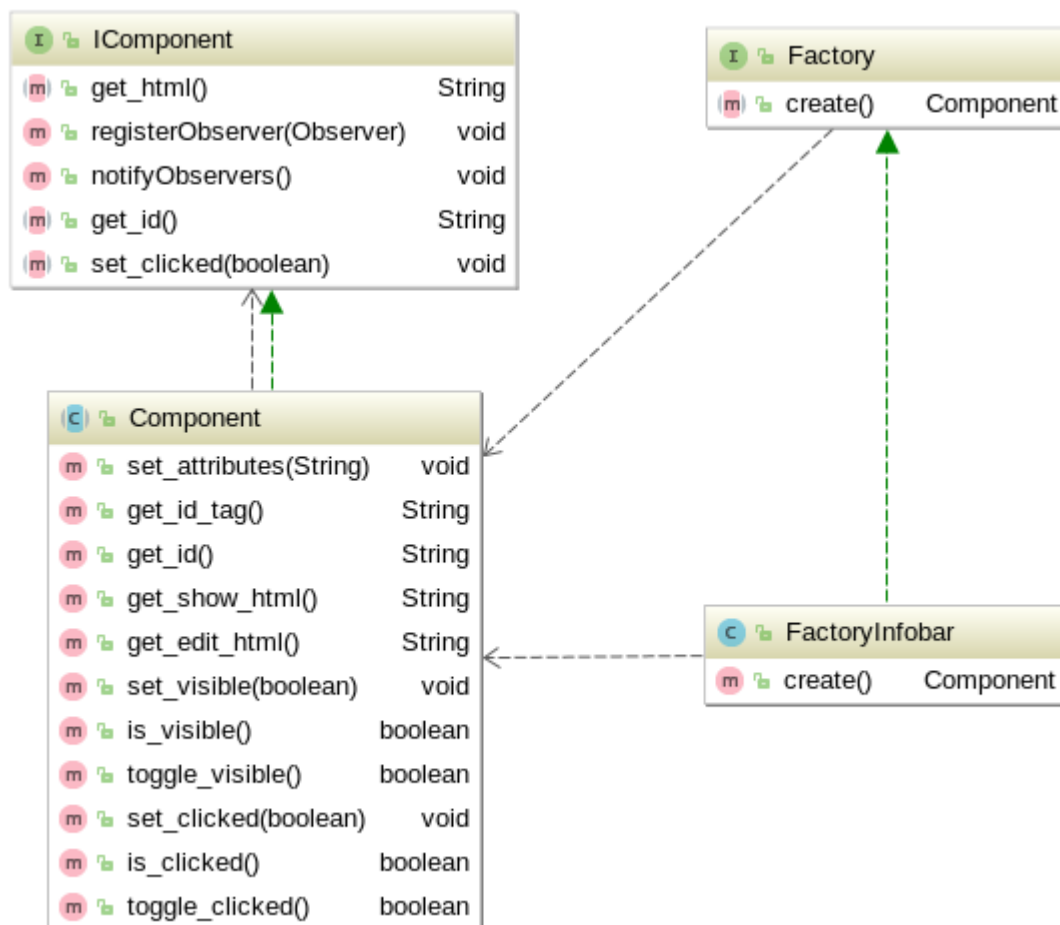
signal: Observer.ToolbarLayerObserver::update()
signal: Observer.ToolbarLayerObserver::update()
signal: Observer.ToolbarLayerObserver::update()

```

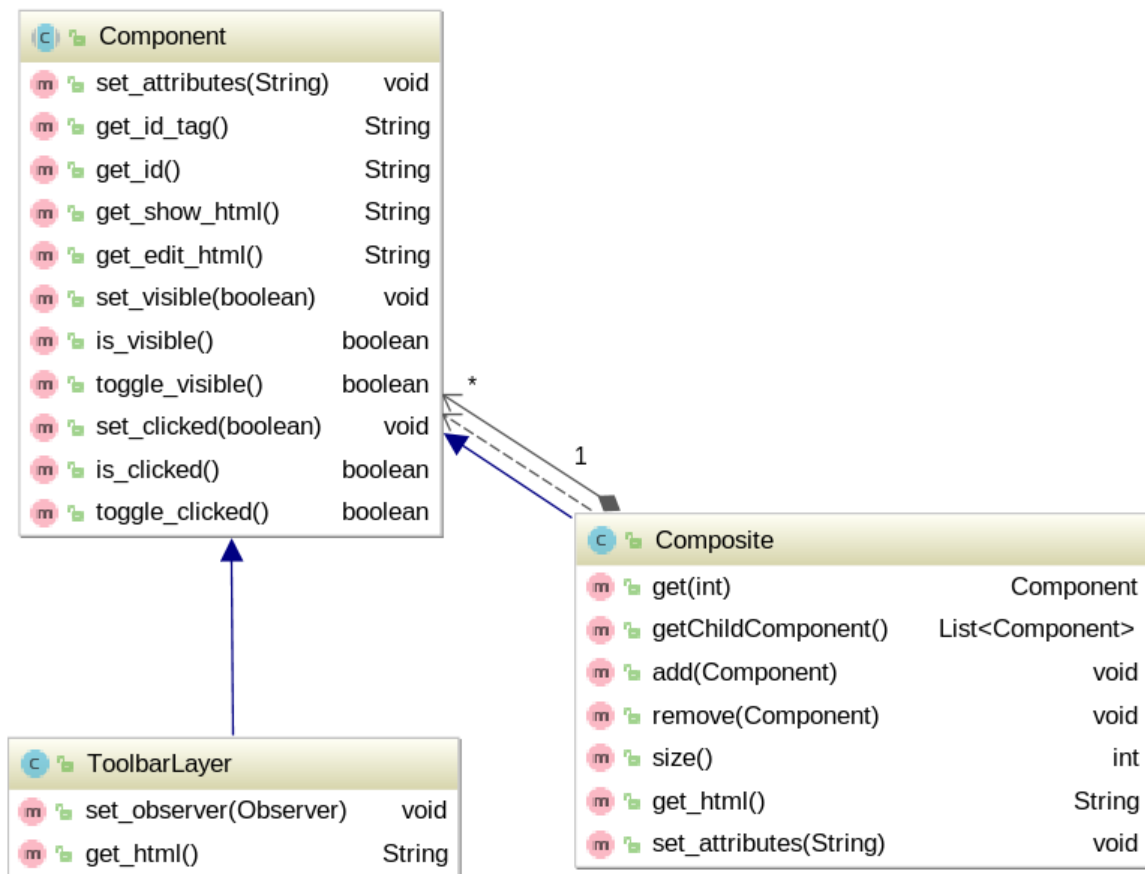
• Strategy Pattern



• Factory Method Pattern



- Composite Pattern



Coding practices

~~describing how and to what extent you have considered coding practices. Discuss and show examples from your code.~~

Defensive programming

~~about how and to what extent you have considered defensive programming. Discuss and show examples from your code.~~

Code metrics

~~of your final implementation, covering the same code metric requirements as in SUPD. Include also a discussion on code bugs found, and their resolution.~~

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
- deletion Schmon
- editing Bareis
- movement (Bareis over editing)

FR3

Bareis

FR4

Schmon

Quality Requirements (QRs)

QR1

QR2

QR3

QR4

QR5

QR6

QR7:

- | | | |
|---------------------|--------------------------|-------------------------|
| • Basic files | Server | Bareis, Schmon, Simkina |
| | Toolbar | Bareis, Schmon, Simkina |
| | ToolbarLayer | Bareis, Schmon, Simkina |
| | ToolbarOperation | Bareis, Schmon, Simkina |
| | Menubar | Bareis, Schmon |
| | Draws | Bareis, Schmon |
| • Observer Pattern | ToolbarLayerObserver | Bareis |
| • Strategy Pattern | StrategySvgEdit | Bareis |
| | StrategySvgShow | Bareis |
| • Iterator Pattern | IteratorLayer (not used) | Bareis |
| | IteratorSvgs | Bareis |
| • Composite Pattern | for Screen | Bareis |
| • Proxy Pattern | Proxy for HTML | Bareis, Schmon |

• Abstract Factory Pattern	ColorAbstractFactory	Simkina
	ColorDraw	Simkina
	Color	Simkina
	DrawAbstractFactory	Simkina
	DefaultAbstractFactory	Simkina
• Factory Method Pattern	FactoryDrawArray	Bareis
	FactoryInfobar	Bareis
	FactoryLayer	Bareis
	FactoryMenubar	Bareis
	FactoryToolbar	Bareis
	FactoryToolbarLayer	Bareis
	FactoryToolbarOperation	Bareis
• Decorator Pattern	Format	Schmon
	DecoratorFileSVG	Schmon
	PlainFormat	Schmon
	LightDesign	Schmon
	DarkDesign	Schmon

QR8

QR9

HowTo

~~documenting how the application is to be launched, initialized, and tested.~~