Ministry of Education of Republic of Moldova

Technical University of Moldova

Faculty of Computers, Informatics and Microelectronics

Computer Science

Object-Oriented Programming

Laboratory work #4

SOLID principles

Authors:
Daniel Surdu

Supervisor:
Anastasia Șerșun

Chișinău - 2018

1 Work purpose:

The study and implementation of the last of the SOLID object-oriented design principles (Open/-closed principle).

2 Task implementation

2.1 Open/closed principle

Software entities (classes, modules, functions, etc.) should be open for extension, but closed for modification. The main idea behind this one is that in one part of your code, you have your abstractions such as classes, modules, higher order functions that do one thing and do it well. You want to keep them pretty, cohesive and well behaved.

```
require './square.rb'
require './rectangle.rb'
require './triangle.rb'

class FigureGenerator

def generate(fig = Square)
  @figure = fig.new
end

def show_perimeter
  puts "Perimeter: #{@figure.get_perimeter}"
end

def show_area
  puts "Area: #{@figure.get_area}"
end

def output_figure
  @figure.output_edges
end

end
```

This class has the method generate(fig = Square) that make this class Open for extension and close for modifications. Any time we can add another figure to this class and the program will work with no problems. In the LAB_3 I had only Square, Triangle and Rectangle that were working with

this class and method. Now I added one more class called Parallelogram and program continue to work as good as before.

```
require './square.rb'
require './rectangle.rb'
require './triangle.rb'
require './parallelogram'
require './figure_generator'

generated_figure = FigureGenerator.new

[Triangle, Rectangle, Square, Parallelogram].each do |figure|

  puts "\n#{figure}"

  generated_figure.generate(figure)

  generated_figure.show_perimeter
  generated_figure.show_area
  generated_figure.output_figure
end
```

and the results are:

```
Triangle
Perimeter: 21
Area: 0.0
a = 10
b = 9
c = 2

Rectangle
Perimeter: 38
Area: 34
a = 2
b = 17
c = 2
z = 17

Square
Perimeter: 52
```

```
Area: 169
a = 13
b = 13
c = 13
z = 13

Parallelogram
Perimeter: 34
Area: 15
a = 15
b = 2
c = 15
z = 2
h = 1
```

Conclusion

During this laboratory work, I studied the last out of five SOLID principles: Open/Closed Principle. This enabled me to get a better understanding of the principles and to apply this knowledge in making my code better.

References

1 SOLID Principles, `https://robots.thoughtbot.com/back-to-basics-solid`

2 Open-Closed Principles, `https://medium.com/@tedtoer/open-closed-principle-in-ruby-exa`