**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Кафедра «информационные технологии и компьютерные системы»

**ОТЧЕТ**

о выполнении расчетно-графической работы №1

по дисциплине: «Технологии разработки программного обеспечения»

Выполнил:

ст.гр.ИВТб-21-2-о\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Родионов Алексей Андреевич

Проверил:

Бакликов Дмитрий Олегович

Севастополь, 2023 г.

**ТЕМА РАБОТЫ:**

ИЗУЧЕНИЕ КЛАССОВ И МОДУЛЕЙ В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ RUBY

**ЦЕЛЬ РАБОТЫ:**

Получить практические навыки по работе с классами и модулями в языке

программирования Ruby.

**ПОСТАНОВКА ЗАДАЧИ:**

Для выполнения работы необходимо:

1. Переписать банкомат из лабораторной работы №4 на работу с классами. Класс должен называться CashMachine. Программа должна запускаться с помощью метода класса init, создавать экземпляр класса и взаимодействовать с пользователем согласно условиям задачи.

2. Переписать банкомат из ЛЗ #4 (класс CashMachine) на работу с веб сервером.

Необходимо заменить ввод и вывод данных (по условию изначальной задачи) на запросы, вместо работы с консолью.

GET /deposit?value=

GET /withdraw?value=

GET /balance

**ХОД РАБОТЫ**

Код программы:

CashMachine.rb:

class CashMachine

def initialize (balance)

@balance = balance

end

def deposit (amount)

raise ArgumentError.new("Amount musn't be negative") if (amount.to\_i < 0)

@balance += amount.to\_i

end

def withdraw (amount)

raise ArgumentError.new("Amount must be lower or equal to current balance") if (amount.to\_i > @balance)

raise ArgumentError.new("Amount musn't be negative") if (amount.to\_i < 0)

@balance -= amount.to\_i

end

def balance ()

@balance

end

end

CashMachineServlet.rb:

require "webrick"

require\_relative "../Entities/CashMachine"

class CashMachineServlet < WEBrick::HTTPServlet::AbstractServlet

def initialize (server, cmObj)

@cmObj = cmObj

super(server)

end

def do\_GET (request, response)

result = ""

if request.query["value"]

amount = request.query["value"]

response.status = 200

response.content\_type = "text/plain"

case request.path

when "/deposit"

result = @cmObj.deposit(amount)

when "/withdraw"

result = @cmObj.withdraw(amount)

end

else

case request.path

when "/balance"

result = @cmObj.balance()

end

end

response.body = result.to\_s + "\n"

end

end

main.rb:

require 'webrick'

require\_relative "Infrastructure/CashMachineServlet"

require\_relative "Entities/CashMachine"

server = WEBrick::HTTPServer.new(:Port => 8000)

atmObj = CashMachine.new(500)

server.mount "/", CashMachineServlet, atmObj

trap('INT') { server.shutdown }

server.start

Структура программы:

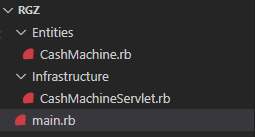


Рисунок 1 – Структура программы РГЗ

Результаты выполнения программы (Рис. 2-6):

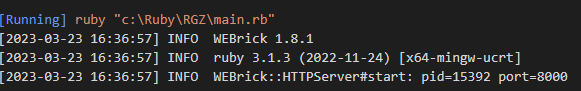


Рисунок 2 – Выполнение программы в терминале VS Code

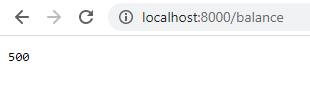


Рисунок 3 – Выполнение запроса balance (GET /balance)

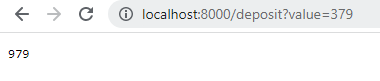


Рисунок 4 – Выполнение запроса deposit (GET /deposit?value=)

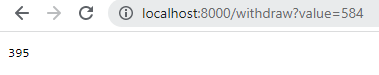


Рисунок 5 – Выполнение запроса withdraw (GET /withdraw?value=)



Рисунок 6 – Поведения программы при возникновении ошибки

Репозиторий: https://github.com/fralorange/cdg-practice-and-lab-works

**ВЫВОД**

В ходе выполнения данной расчетно-графической работы были получены знания и практические навыки по работе с классами и простейшими локальными HTTP веб-серверами.