

Министерство науки и высшего образования РФ  
ФГАУВО «Севастопольский государственный университет»  
Кафедра информационных технологий и компьютерных систем

## ОТЧЕТ

Лабораторная работа №4

по дисциплине

«Вычислительные системы»

Выполнил: ст. гр. ИВТ/б-17-1-о

Кожевников И.В.

Севастополь  
2020

**Постановка задачи:** изучить базовые основы языка программирования Ruby.

### Задание:

- 1) Переписать банкомат из ЛЗ #3 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) Пользователь запускает программу и отвечает на вопрос, с каким ресурсом он хочет взаимодействовать
  - После чего он может передать тип запроса GET/POST/PUT/DELETE
  - GET index - должен возвращать все посты из памяти и их индекс в массиве (прим. 0. Hello World \n 1. Hello (again))
  - GET show - должен запрашивать идентификатор поста и показывать пост по переданному идентификатору (как в index только 1 пост)
  - POST create - должен запрашивать текст поста, добавлять его в массив постов и возвращать в ответ идентификатор поста и сам пост
  - PUT update - должен запрашивать идентификатор поста, потом новый текст поста и заменить его. В результате выводит пост (как в функции index)
  - DELETE destroy - должен запрашивать идентификатор поста, затем удалять его из массива постов

Нужно реализовать логику для PostsController

- Добавить CommentsController самостоятельно

В ответе по заданию расписать, понимание работы класса Router, при этом, по которому используется `extend` для модуля `Resource` В случае неправильного ввода (команды), ваша программа должна выдать соответствующее сообщение об ошибке, которое говорит клиенту, как ее исправить. Нельзя просто выводить "Error!" - это не поможет

### Текст программы:

```
1)
require 'io/console'

DefBalance = 100.0

class CashMachine

  def self.init
    CashMachine.new.choose_option
  end

  def readBalance
    if File.exist?('balance.txt')
      balance = File.read('balance.txt').chomp
      puts "Your current balance: #{balance}"
    end
  end
end
```

```

    return balance
  else
    balance = DefBalance
    return balance
  end
end

def balance
  if File.exist?('balance.txt')
    balance = File.read('balance.txt').chomp
    puts "Your current balance: #{balance}"
    choose_option
  else
    balance = DefBalance
    puts "Your current balance: #{balance}"
    choose_option
  end
end

def deposit
  balance = readBalance.to_f
  puts 'Enter amount to be deposited...'
  input = gets.chomp
  case input
  when /\d*\.\?\d/
    balance += input.to_f
    File.write('balance.txt', balance.to_s)
  else
    puts 'Input must be a positive number'
    deposit
  end
  choose_option
end

def withdraw
  balance = readBalance.to_f
  puts 'Enter amount to be withdrawn...'
  input = gets.chomp
  case input
  when /\d*\.\?\d/
    balance -= input.to_f
    File.write('balance.txt', balance.to_s)
  else
    puts 'Input must be a positive number'
    withdraw
  end
  choose_option
end

def choose_option
  puts 'D(deposit), W(withdraw), B(balance), Q (quit)'
  key = STDIN.getch.upcase!.chomp.to_s
  puts key
  case key
  when 'D'
    deposit
  when 'W'
    withdraw
  when 'B'
    balance
  when 'Q'
    puts 'Exiting...'
    exit
  else
    choose_option
  end
end

```

```

    end
  end
end
CashMachine.init

```

2)

```

class Post
  attr_accessor :content, :comments
  def initialize(content)
    @content = content
    @comments = []
  end
end

module Resource
  def connection(routes)
    if routes.nil?
      puts "No route matches for #{self}"
      return
    end

    loop do
      print 'Choose verb to interact with resources (GET/POST/PUT/DELETE) / q to exit: '
      verb = gets.chomp.downcase
      puts verb.inspect
      break if verb == 'q'

      if verb != "get" and verb != "post" and verb != "put" and verb != "delete" and verb !=
"q" then
        puts "Exception: wrong argument"
        next
      end

      action = nil

      if verb == 'get'
        print 'Choose action (index/show) / q to exit: '
        action = gets.chomp.downcase
        break if action == 'q'
      end

      if action.nil?
        routes[verb].call
      else
        routes[verb][action].call
      end
    end
  end
end

class PostsController
  extend Resource

  def initialize(posts)
    @posts = posts
  end

  def index
    (0..(@posts.size - 1)).each { |i|
      puts i.to_s + " " + @posts[i]
    }
  end
end

```

```

def show
  print 'Choose index of post: '
  index = gets.chomp
  puts @posts[index.to_i]
end

def create
  puts "Enter your post's content:"
  content = gets.chomp
  @posts.push(Post.new(content))
end

def update
  print 'Choose index of post: '
  index = gets.chomp
  puts "Rewrite your post's content:"
  content = gets.chomp
  @posts[index.to_i] = content
end

def destroy
  print 'Choose index of post: '
  index = gets.chomp
  @posts.delete_at(index.to_i)
end

end

#Controller for comments
class CommentsController
  extend Resource

  def initialize(comments)
    @comments = comments
  end

  def index
    (0..(@posts.size - 1)).each { |i|
      puts "post index: #{i}"
      puts "post content:\n" + @posts[i].content
      puts "comments: "
      (0..(@posts[i].comments.size - 1)).each { |j|
        puts j.to_s + " " + @posts[i].comments[j]
      }
    }
  end

  def show

```

## Результат работы программы:

1)

```
=> :choose_option
lr4_pt1.rb(main):089:0> CashMachine.initD(deposit), W(withdraw), B(balance), Q (quit)
B
Your current balance: 279.0
D(deposit), W(withdraw), B(balance), Q (quit)
D
Your current balance: 279.0
Enter amount to be deposited...
100
D(deposit), W(withdraw), B(balance), Q (quit)
B
Your current balance: 379.0
D(deposit), W(withdraw), B(balance), Q (quit)
W
Your current balance: 379.0
Enter amount to be withdrawn...
120
D(deposit), W(withdraw), B(balance), Q (quit)
B
Your current balance: 259.0
D(deposit), W(withdraw), B(balance), Q (quit)
Q
Exiting...
```

2)

```
C:\Prog1\Ruby27-x64\bin\ruby.exe C:/Users/PussyDestroyer/Documents/GitHub/glowing-octo-fiesta/LR4/lr4_pt2.rb
Choose resource you want to interact (1 - Posts, 2 - Comments, q - Exit):1
Choose verb to interact with resources (GET/POST/PUT/DELETE) / q to exit:put
"put"
Choose index of post:1
Rewrite your post's content:
qweqwe
Choose verb to interact with resources (GET/POST/PUT/DELETE) / q to exit:get
"get"
Choose action (index/show) / q to exit:show
Choose index of post:1
qweqwe
Choose verb to interact with resources (GET/POST/PUT/DELETE) / q to exit:post
"post"
Enter your post's content:
asdadsa
Choose verb to interact with resources (GET/POST/PUT/DELETE) / q to exit:delete
"delete"
Choose index of post:2
Choose verb to interact with resources (GET/POST/PUT/DELETE) / q to exit:get
"get"
```

## Вывод:

В ходе выполнения работы были изучены базовые основы языка программирования Ruby.