

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

ОТЧЕТ
Лабораторная работа №3
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)

У вас есть файл, со следующей структурой и содержащий не менее 10 строк: Имя, фамилия и возраст

Прим.

Иван Иванов 20

Петр Петров 21

Необходимо прочитать файл, затем запросив у пользователя ввод возраста, найти в результатах чтения файла, студента чей возраст равен введенному числу и записать этого студента(тов) в другой файл с названием results.txt. После этого перезапросить ввод.

Программа завершается выводом на экран содержимого файла results.txt построчно, если

все студенты из первого файла были записаны во второй или если пользователь ввел с клавиатуры -1

2) Напишите программу, которая начинается с чтения банковского баланса клиента из файла с именем balance.txt. Этот файл содержит одну строку со стартовым балансом клиента. Если файл не существует, используйте стартовый баланс 100.0, который должен быть константой. После этого программа повторно предложит клиенту внести деньги, вывести деньги, вывести деньги, проверить баланс или выйти, используя буквы D(deposit), W(withdraw), B(balance) и Q (quit), программа должна принимать на вход значения в верхнем или нижнем регистре. Для депозитов (D), программа подсказывает сумму. Сумма должна быть больше нуля. Если сумма действительна, программа добавляет сумму депозита к балансу клиенту и отображает новый баланс. При снятии средств программа выдаст запрос на сумму. Сумма должна быть больше нуля и меньше или равна текущему балансу. Если сумма корректна, то программа вычитает сумму вывода из баланса клиента и отображает новый баланс. При снятии средств (W) программа выдаст запрос на сумму. Сумма должна быть больше нуля и меньше, или равна текущему балансу. Если сумма корректна, то программа вычитает сумму вывода из баланса клиента и отображает новый баланс. Для проверки баланса (B) программа просто выводит текущий баланс. Когда клиент решит выйти из программы (Q), программа запишет текущий баланс обратно в файл balance.txt. В случае неправильного ввода (команда или сумма), ваша программа должна выдать соответствующее сообщение об ошибке, которое говорит клиенту, как её исправить. Нельзя выводить "Error!" - это не поможет.

Текст программы:

1)

```
require 'fileutils'
SRC = 'Temp.txt'
DEST = 'input.txt'

def openReadDelete(age)
  File.foreach('input.txt') { |line| checkForConformity(age, line) }
  FileUtils.cp(SRC, DEST)
  ageInput
end

def checkForConformity(age, line)
  if line.include?(age.to_s)
    File.write('results.txt', "#{line.chomp}\n", mode: 'a')
  else
    File.write('Temp.txt', "#{line.chomp}\n", mode: 'a')
  end
end
```

```

def ageInput
  puts 'Enter age or -1 to exit'
  age = gets.chomp.to_s
  if age.include?('-1') == true
    FileUtils.cp(SRC, DEST)
    if File.exist?('Temp.txt')
      File.delete('Temp.txt')
    end
    exit
  else
    if File.exist?('Temp.txt')
      File.delete('Temp.txt')
    end
    openReadDelete(age)
  end
end
ageInput

```

2)

```

require 'io/console'

DefBalance = 100.0

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

```

Результат работы программы:

1)

```

C:\Progi\Ruby27-x64\bin\ruby.exe C:/Users/PussyDestroyer/Documents/GitHub/glowing-octo-fiesta/LR3/lr3_pt1.rb
Enter age or -1 to exit
11
Enter age or -1 to exit
12
Enter age or -1 to exit
13
Enter age or -1 to exit
14
Enter age or -1 to exit
-1

```

Process finished with exit code 0

1	Isaiah Martinez 16	1	Ethan Jackson 11
2	Andre Gray 17	2	Jesse Dominic Griffin 11
3	Steven Parker 18	3	Connor Ramirez 12
4	Jeremiah Coleman 19	4	Christian Andre Coleman 12
5	Zachary Scott 20	5	Nathan Ross 13
6	Gavin Oen Gray 16	6	Nathan Mason Wright 13
7	Zachary Diego Patterson 17	7	Daniel Jones 14
8	Samuel Nicholas Poell 18	8	Sebastian Jake Young 14
9	Angel Christopher Butler 19		
10	Juan Carter Washington 20		
11	Jordan Garcia 15		
12	Aaron Landon Alexander 15		

2)

```
=> :choose_option
lr3_pt2.rb(main):100:0> choose_optionD(deposit), W(withdraw), B(balance), Q (quit)
B
Your current balance: 1222.3
D(deposit), W(withdraw), B(balance), Q (quit)
D
Your current balance: 1222.3
Enter amount to be deposited...
1000
D(deposit), W(withdraw), B(balance), Q (quit)
B
Your current balance: 2222.3
D(deposit), W(withdraw), B(balance), Q (quit)
W
Your current balance: 2222.3
Enter amount to be withdrawn...
100
D(deposit), W(withdraw), B(balance), Q (quit)
B
Your current balance: 2122.3
D(deposit), W(withdraw), B(balance), Q (quit)
Q
Exiting...
```

1 2122.3

Вывод:

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