



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

**О Т Ч Е Т**

по лабораторной работе № 6

Название: Итерационные циклы

Дисциплина: Языки интернет-программирования

Студент

ИУ6-31Б

(Группа)

\_\_\_\_\_  
(Подпись, дата)

Лазарев Е.С.

(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

Малахов Д.В.

(И.О. Фамилия)

Москва, 2022

Цель работы: изучить итерационные цикла языка Ruby

## Часть 1

Решить задачу, организовав итерационный цикл с точностью  $\xi = 10^{-4}, 10^{-5}$ .

Вычислить сумму ряда:  $S = \sum_{k=1}^{\infty} \frac{1}{k(k+1)}$ , точное значение равно 1. Определить, как изменяется число итераций при изменении точности.

Основная программа:

```
def Laba61(eps)
  sum = 0
  k = 1.0
  prev = 1.0
  loop do
    prev = sum
    sum += 1.0 / (k*(k+1))
    k += 1
    break if (prev - sum).abs < eps
  end
  if eps == 0.0001 || eps == 0.00001
    return sum
  else
    return 'Неверно введённое значение точности'
  end
end
```

Программа для взаимодействия с пользователем через консоль:

```
require_relative 'main61.rb'
puts "Введите значение точности: "
eps = gets.to_f
puts(Laba61(eps))
```

Программа для автоматического тестирования на основе MiniTest::Unit:

```
require_relative 'main61.rb'
RSpec.describe "SuperMain" do
  it 'should do ok if eps is correct' do
    expect(Laba61(0.0001)).to eq 0.9900990099009898
    expect(Laba61(0.00001)).to eq 0.9968454258675077
  end
  it 'should not do ok if eps is incorrect' do
    pr = Random.new
    expect(Laba61(pr.rand(100))).to eq 'Неверно введённое значение точности'
  end
end

end
```

```
● yulitylight@yulitylight-HLYL-WXX9:~/Документы/Лабы по ЯИП/Лаба 6$ rspec test61.rb
..

Finished in 0.00275 seconds (files took 0.06987 seconds to load)
2 examples, 0 failures
```

## Часть 2

Решить предыдущее задание с помощью Enumerable или Enumerator.

Основная программа:

```
def Laba62(eps)
  list = Enumerator.new do |yielder|
    sum, prev, counter = 0.0, 1.0, 1.0
    loop do
      yielder.yield sum, prev
      prev = sum
      sum += 1.0 / (counter*(counter+1))
      counter += 1
    end
  end
  if eps == 0.0001 || eps == 0.00001
    a = []
    a.push(list.take_while { |sum, prev| (prev - sum).abs > eps })
    return a[-1][-1][-1]
  else
    return 'Неверно введённое значение точности'
  end
end
```

Программа для взаимодействия с пользователем через консоль:

```
require_relative 'main62.rb'
puts "Введите значение точности: "
eps = gets.to_f
puts(Laba62(eps))
```

Программа для автоматического тестирования на основе MiniTest::Unit:

```
require_relative 'main62.rb'
RSpec.describe "SuperMain" do
  it 'should do ok if eps is correct' do
    expect(Laba62(0.0001)).to eq 0.9898989898989896
    expect(Laba62(0.00001)).to eq 0.9968253968253966
  end
  it 'should not do ok if eps is incorrect' do
    pr = Random.new
    expect(Laba62(pr.rand(100))).to eq 'Неверно введённое значение точности'
  end
end
```

```
● yulitylight@yulitylight-HLYL-WXX9:~/Документы/Лабы по ЯИП/Лаба 6$ rspec test62.rb
..

Finished in 0.00202 seconds (files took 0.05755 seconds to load)
2 examples, 0 failures
```

## Часть 3

Составить метод `intprg` вычисления определенного интеграла по формуле прямоугольников:  $S = \frac{b-a}{n} \sum_{i=1}^n f(x_i)$ , где  $n$  – количество отрезков разбиения. В основной программе использовать метод `intprg` для вычисления интегралов:  $\int_0^1 \frac{e^x}{x+1} dx$  и  $\int_0^2 x(x-1) dx$ .

Реализовать вызов метода двумя способами: в виде передаваемого `lambda`-выражения и в виде блока.

Основная программа:

```
def intprg(a1, b1, a2, b2, xd2, n)
  h1 = (b1 - a1).fdiv(n)
  h2 = (b2 - a2).fdiv(n)
  x1 = a1.to_f
  x2 = a2.to_f
  sum1, sum2 = 0, 0
  (1..n).each do |i|
    xds1 = yield x1
    xds2 = xd2.call(x2)
    sum1 += xds1
    sum2 += xds2
    x1 += h1
    x2 += h2
  end
  if n <= 0
    return 'Было введено некорректное значение'
  else
    return 'Первый интеграл равен: ', sum1 * h1, 'Второй интеграл равен: ', sum2 * h2
  end
end
```

Программа для взаимодействия с пользователем через консоль:

```
require_relative 'main63.rb'
xd1 = -> (x1) { Math.exp(x1) / (x1 + 1) }
a1, b1 = 0, 1
xd2 = lambda { |x2| x2*(x2-1) }
a2, b2 = 0, 2
puts 'Введите количество отрезков, на которые следует разбить наши прямоугольники: '
n = gets.to_i
puts(intprg(a1, b1, a2, b2, xd2, n, &xd1))
```

Программа для автоматического тестирования на основе RSpec:

```
require_relative 'main63.rb'
RSpec.describe "SuperMain" do
  it 'should do ok if n is postive' do
    xd1 = -> (x1) { Math.exp(x1) / (x1 + 1) }
    xd2 = lambda { |x2| x2*(x2-1) }
    result = intprg(0, 1, 0, 2, xd2, 99, &xd1)
    expect(result[1]).to eq 1.1235780181190325
    expect(result[3]).to eq 0.6466006870047234
  end
end
```

```
• yultylight@yultylight-HLYL-WXX9:~/Документы/Лабы по ЯИП/Лаба 6$ rspec test63.rb
.  
  
Finished in 0.00171 seconds (files took 0.056 seconds to load)  
1 example, 0 failures
```

Вывод: я изучил итерационные цикла языка Ruby