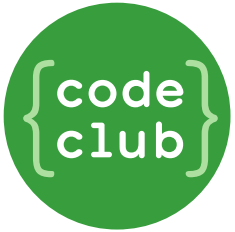
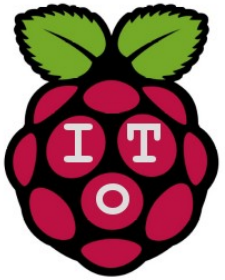


Internet of Things



Подпрограммы в **Ruby**

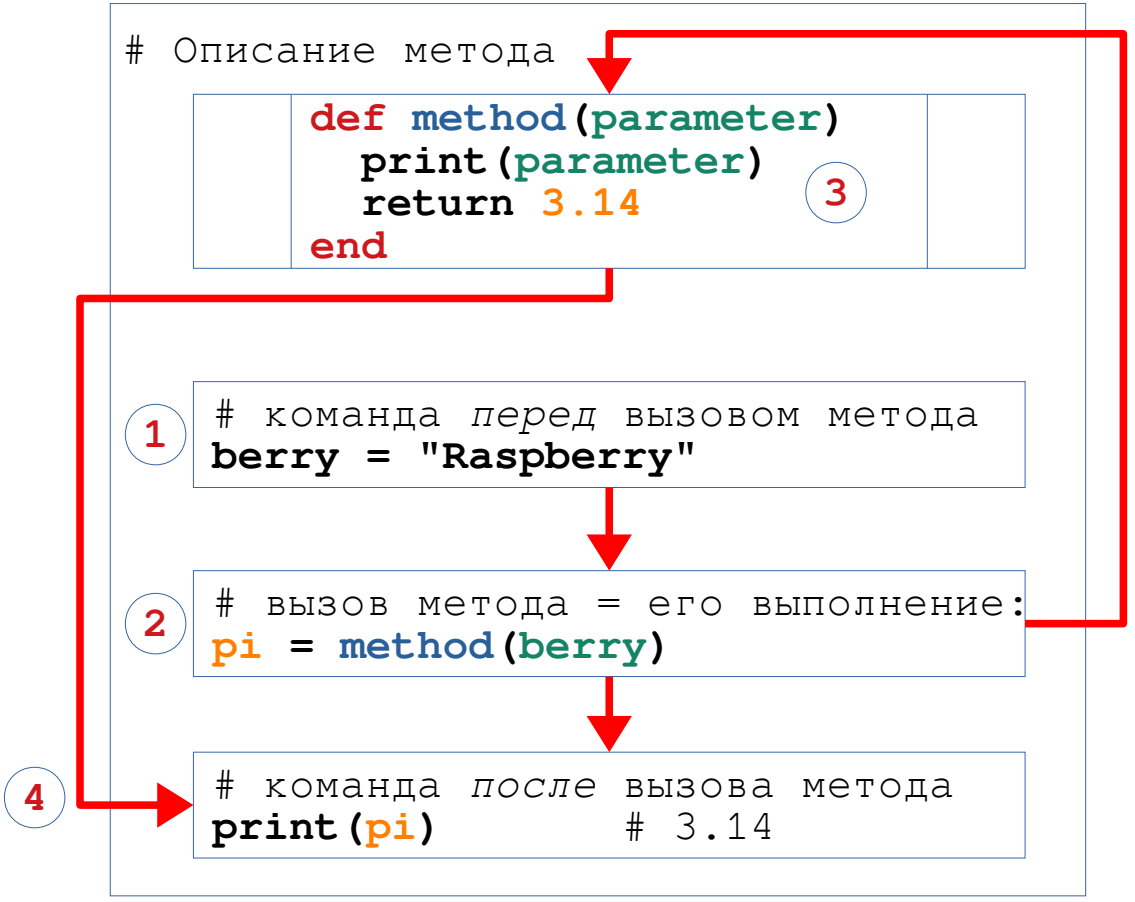
Шадринск
2018-2019

М. В. Шохирев

Методы (подпрограммы)

```
def method(p1, p2)        # описание метода с двумя параметрами
  result = p1 + p2        # команды в теле метода
  return result           # вернуть результат действий
end                       # конец описания метода

method(40, 2)             # вызвать метод (возвращаемое значение отбросить)
r = method(40, 2)         # вызвать метод (возвращаемое значение поместить в r)

class LED
  def dot                 # описание метода без параметров для класса LED
    on_for(0.25)          # включить светодиод на 1/4 секунды
    off_for(0.25)         # выключить светодиод на 1/4 секунды
  end                     # конец описания метода
end

led = LED.new(18)         # создать объект led класса LED с параметром 18
3.times { led.dot }       # 3 раза вызвать для объекта led метод dot()
```

Выполнение метода

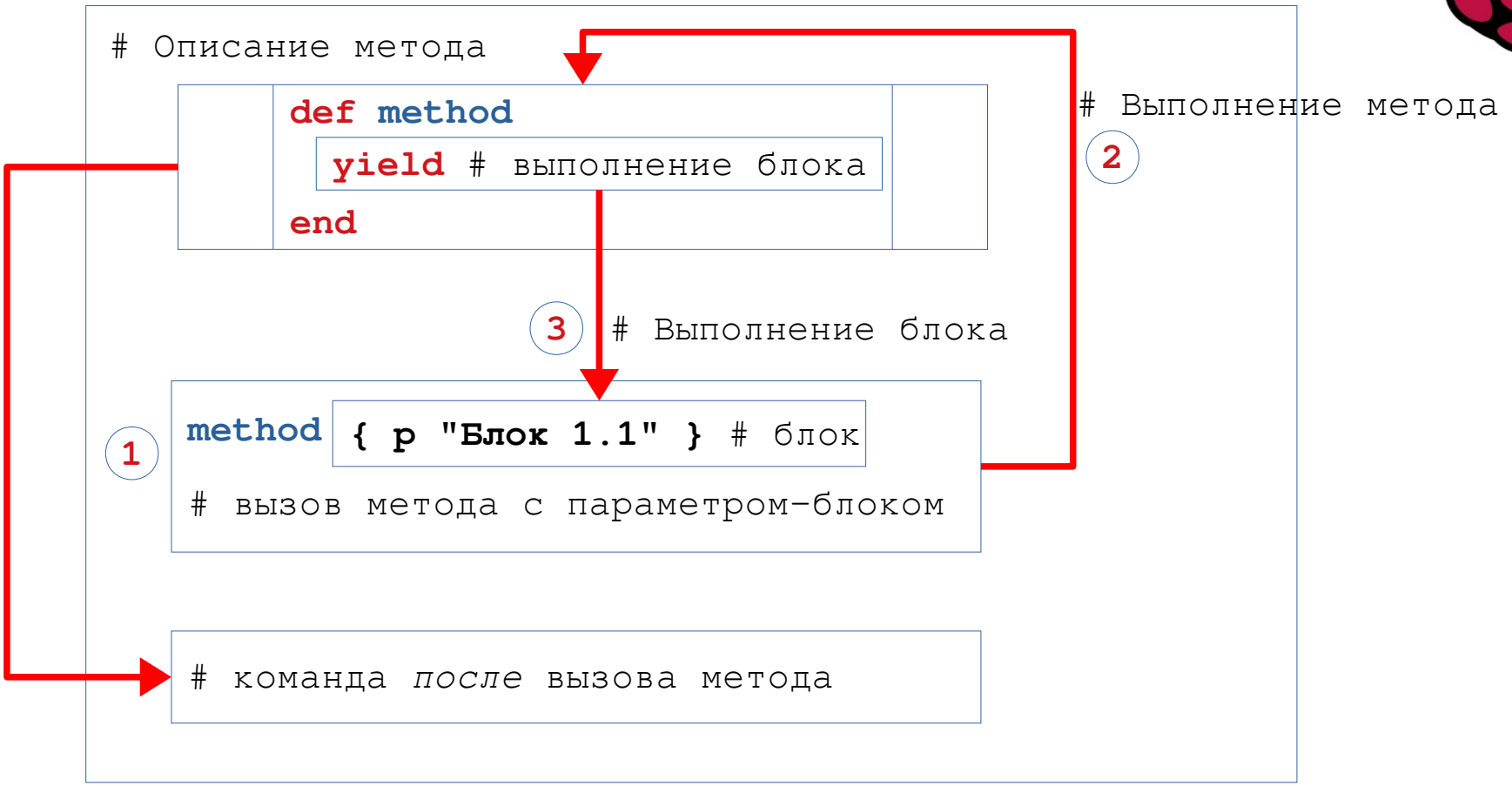


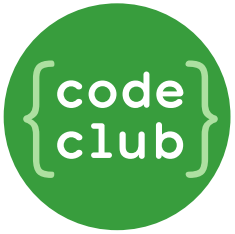
Блоки и методы

Блок — это программный код, который можно передать любому методу последним аргументом внутри фигурных скобок **{ }** или ключевых слов **do** и **end**.

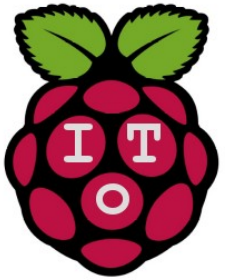
```
# метод, которому можно передавать блок как параметр
def method_1                                  # начало описания метода
  print "> Начало работы метода 1\n"          #   команды в теле метода
  yield if block_given?                       #   блок выполняется только по yield
  print "< Конец работы метода 1\n"           #   команды в теле метода
end                                           # конец описания метода

method_1                                      # вызов метода без блока

method_1 { p "Выполняется блок 1.1" }         # вызов метода с блоком между { и }

# вызов метода с блоком между do и end
method_1 do                                   # начало блока
  print "~ Выполняется блок 1.21\n"           #   команды в теле блока
end                                           # конец блока
```

Выполнение блока





Блоки с параметрами



При вызове метода блоку можно передавать значения параметров (аргументы) так:

```
method { |parameter| do_something_with(parameter) }
```

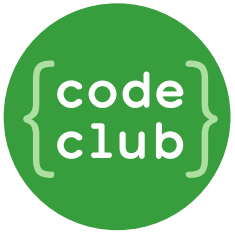
или так:

```
do |parameter|
  do_something_with(parameter)
end
```

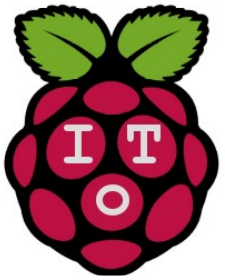
В методе параметр передаётся блоку так:

```
yield(parameter)
```

```
# метод с параметром
def method_2(parameter)                    # начало описания метода
  print "> Начало работы метода 1\n"      #   команды в теле метода
  yield(parameter)                         #   выполнить блок с параметром
  print "< Конец работы метода 1\n"       #   команды в теле метода
end                                        # конец описания метода

method_2("Блок 2.1") { |p1| print "---"+p1+"---" }

method_2("Блок 2.2") { |p2| print "==="+p2+"===" }
```



Справочник и учебник по Ruby

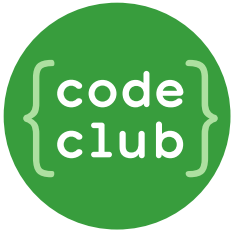


Краткий справочник по синтаксису языка — в файле:

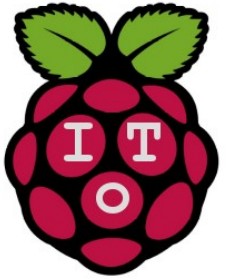
`IoT-Ruby_syntax.pdf`

Учебник для начинающих по языку — в файле:

`~/Documents/books/Learn_To_Program-Ch.Pine-ru.pdf`



Что непонятно?



Какие конструкции языка
для вас сложны?

Что нужно объяснить дополнительно?