Ruby Gems
Лекция 3

План

- Gems
- Bundler
- Создание и публикация пакетов
- TDD
- Test::Unit
- Minitest

rails 6.0.0

Ruby on Rails is a full-stack web framework optimized for programmer happiness and sustainable productivity. It encourages beautiful code by favoring convention over configuration.

VERSIONS:

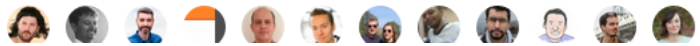
6.0.0 - August 16, 2019 (6.5 KB)
6.0.0.rc2 - July 22, 2019 (6.5 KB)
6.0.0.rc1 - April 24, 2019 (6.5 KB)
6.0.0.beta3 - March 13, 2019 (6.5 KB)
6.0.0.beta2 - February 25, 2019 (6.5 KB)

Show all versions (358 total) →

RUNTIME DEPENDENCIES (14):

actioncable = 6.0.0
actionmailbox = 6.0.0
actionmailer = 6.0.0
actionpack = 6.0.0
actiontext = 6.0.0
actionview = 6.0.0
activejob = 6.0.0
activemodel = 6.0.0
activerecord = 6.0.0
activestorage = 6.0.0
activesupport = 6.0.0
bundler >= 1.3.0
railties = 6.0.0
sprockets-rails >= 2.0.0

OWNERS:



Star 44,076

TOTAL DOWNLOADS
189,550,182

FOR THIS VERSION
225,248

GEMFILE:

```
gem 'rails', '~> 6.
```

INSTALL:

```
gem install rails
```

LICENSE:

MIT

REQUIRED RUBY VERSION:

>= 2.5.0

REQUIRED RUBYGEMS VERSION:

>= 1.8.11

LINKS:

Homepage

Source Code

Команды

- `gem search rails`
- `gem search ^rails`
- `gem search ^rails$ -d`

Команды

```
$ gem search ^rails$ -d

*** REMOTE GEMS ***

rails (6.0.0)
    Author: David Heinemeier Hansson
    Homepage: https://rubyonrails.org

    Full-stack web application framework.
```

Команды

- `gem install rails`
- `gem install rails –no-doc`

Что с моим ruby?

```
irb -rpp #require + pretty_print
>> pp $LOAD_PATH
```

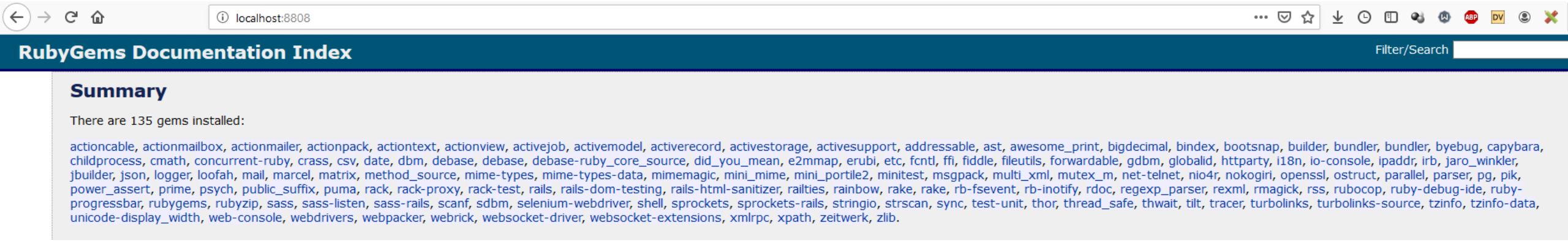
```
irb(main):002:0> pp $LOAD_PATH
["C:/Ruby26-x64/lib/ruby/gems/2.6.0/gems/did_you_mean-1.3.0/lib",
 "C:/Ruby26-x64/lib/ruby/site_ruby/2.6.0",
 "C:/Ruby26-x64/lib/ruby/site_ruby/2.6.0/x64-msvcrt",
 "C:/Ruby26-x64/lib/ruby/site_ruby",
 "C:/Ruby26-x64/lib/ruby/vendor_ruby/2.6.0",
 "C:/Ruby26-x64/lib/ruby/vendor_ruby/2.6.0/x64-msvcrt",
 "C:/Ruby26-x64/lib/ruby/vendor_ruby",
 "C:/Ruby26-x64/lib/ruby/2.6.0",
 "C:/Ruby26-x64/lib/ruby/2.6.0/x64-mingw32"]
```

Что с моим ruby?

```
irb -rpp #require + pretty_print
>> require 'ap'
=> true
>> pp $LOAD_PATH
```

Команды

- `gem uninstall rails`
- `gem list`
- `ri puts #имя метода`
- `gem server`



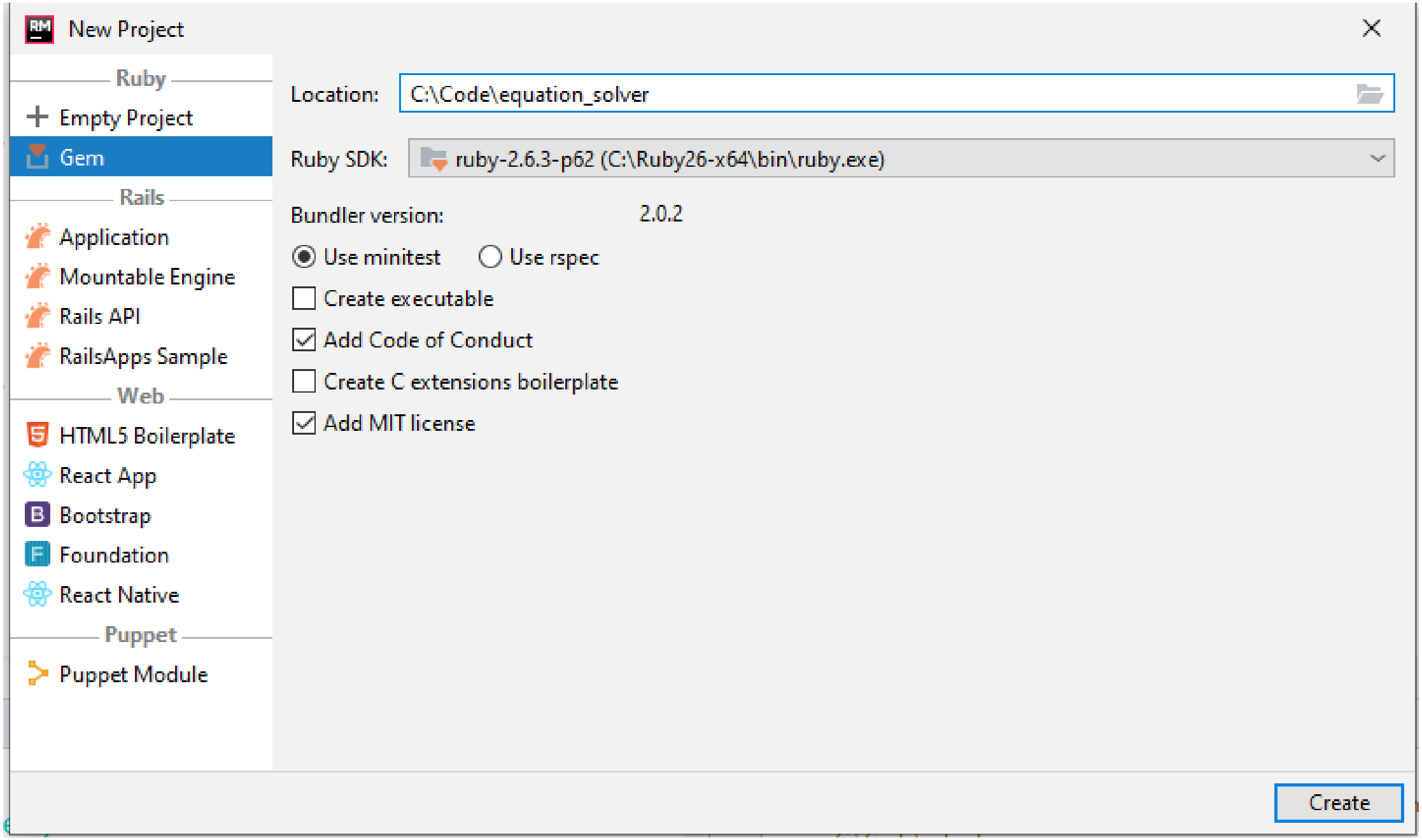
Что внутри у чужих гемов?

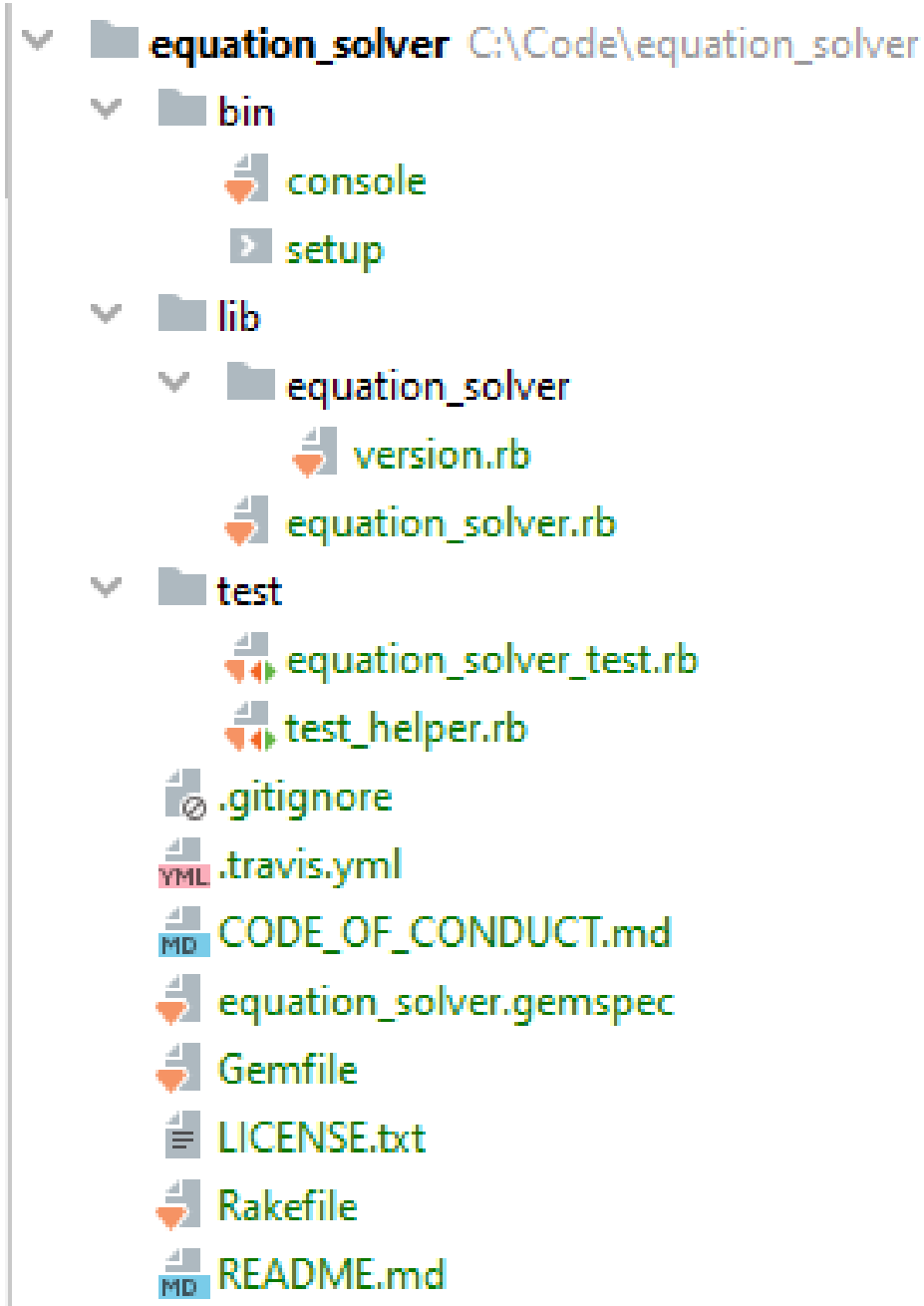
- gem fetch rails
- gem unpack rails

C Extensions

Что внутри у моих гемов?

- Код + тесты
- Документация
- Gemspec


New Project
Ruby
Empty Project
Gem
Rails
Application
Mountable Engine
Rails API
RailsApps Sample
Web
HTML5 Boilerplate
React App
Bootstrap
Foundation
React Native
Puppet
Puppet Module
Location:
C:\Code\equation_solver
Ruby SDK:
ruby-2.6.3-p62 (C:\Ruby26-x64\bin\ruby.exe)
Bundler version:
2.0.2
Use minitest
Use rspec
Create executable
Add Code of Conduct
Create C extensions boilerplate
Add MIT license
Create


equation_solver C:\Code\equation_solver
bin
console
setup
lib
equation_solver
version.rb
equation_solver.rb
test
equation_solver_test.rb
test_helper.rb
.gitignore
.travis.yml
CODE_OF_CONDUCT.md
equation_solver.gemspec
Gemfile
LICENSE.txt
Rakefile
README.md

RDoc

- Видит комментарии перед объявлением классов
- Видит комментарии перед функциями
- Выписывает публичные методы
- Диаграммы, разметка,

Bundler

- `bundle install`
- `bundle update`
- `bundle exec rake`

Gemfile

```
source 'https://rubygems.org'
git_source(:github) { |repo| "https://github.com/#{repo}.git" }

ruby '2.4.4'

# Bundle edge Rails instead: gem 'rails', github: 'rails/rails'
gem 'rails', '~> 5.2.3'
# Use mysql as the database for Active Record
gem 'mysql2', '>= 0.4.4', '< 0.6.0'
# Use Puma as the app server
gem 'puma', '~> 3.11'
# Use SCSS for stylesheets
gem 'sass-rails', '~> 5.0'
```

Gemfile.lock

```
GEM
  remote: https://rubygems.org/
  specs:
    actioncable (5.2.3)
      actionpack (= 5.2.3)
      nio4r (~> 2.0)
      websocket-driver (>= 0.6.1)
    actionmailer (5.2.3)
      actionpack (= 5.2.3)
      actionview (= 5.2.3)
```

GEMSPEC

```ruby
Gem::Specification.new do |s|
  s.name        = 'freewill'
  s.version     = '1.0.0'
  s.summary     = "Freewill!"
  s.description = "I will choose Freewill!"
  s.authors     = ["Nick Quaranto"]
  s.email       = 'nick@quaran.to'
  s.homepage    = 'http://example.com/freewill'
  s.files       = ["lib/freewill.rb", ...]
end
```

Начинаем писать код

```ruby
require "equation_solver/version"

module EquationSolver
  class Error < StandardError; end
end
```

Начинаем писать код

```ruby
require "equation_solver/version"

module EquationSolver
  class Error < StandardError;
  end

  def self.solve(a, b, c)

  end

end
```

Начинаем писать код

```ruby
require "equation_solver/version"

# Solves quadratic equations
module EquationSolver
  class Error < StandardError;
  end

  # equation a*x^2+b*x+c=0
  # output: [x1, x2], nil if equation has now solutions
  def self.solve(a, b, c)

  end

end
```

Как решать квадратные уравнения?

$$ax^2 + bx + c = 0$$

$$x_{1,2} = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Готово?

```ruby
def self.solve(a, b, c)
  d = Math.sqrt(b**2 - 4 * a * c)
  [(-b - d) / (2 * a), (-b + d) / (2 * a)]
end
```

```
$ gem build equation_solver.gemspec
  Successfully built RubyGem
  Name: equation_solver
  Version: 0.1.0
  File: equation_solver-0.1.0.gem
```

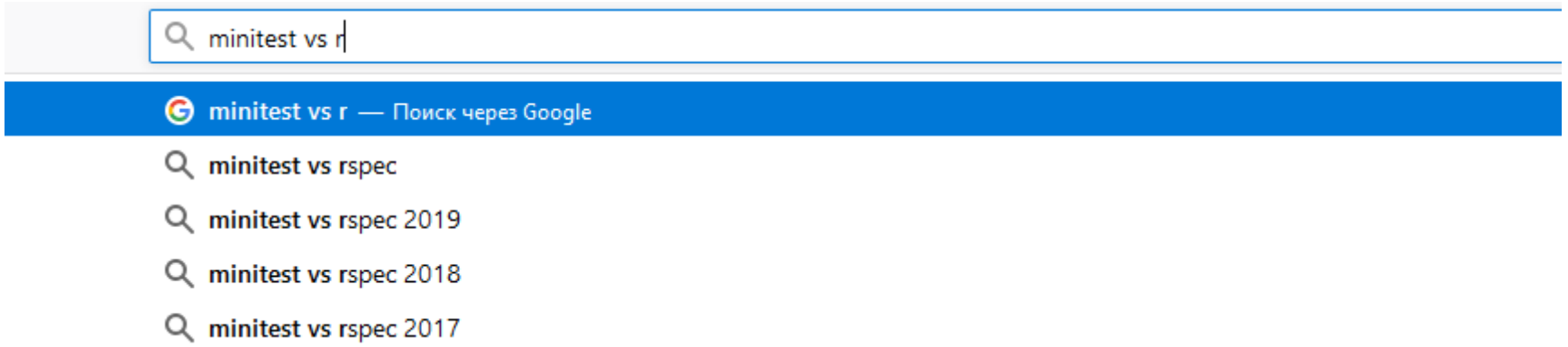
Проверяем

```
$ irb
irb(main):001:0> require 'equation_solver'
=> true
irb(main):002:0> EquationSolver.solve(1,4,0)
=> [-4.0, 0.0]
irb(main):003:0> EquationSolver.solve(0,4,0)
=> [-Infinity, NaN]
irb(main):004:0>
```

Публикация

- `gem push equation_solver-0.1.0.gem`
- `gem yank equation_solver -v 0.1.0`

Minitest vs RSpec vs Test::Unit




```
rspec is a testing DSL. minitest is ruby.

-- Adam Hawkins, "Bow Before MiniTest"
```

TDD

- Написать тесты
- Выполнить тесты и убедиться, что они не проходят
- Написать минимальный код, который пройдет тесты
- Выполнить тесты и убедиться, что они проходят
- Выполнить рефакторинг

Шаги выполнения теста

- Подготовка UUT (unit under test)
- Выполнение кода
- Валидация результатов
- Очистка

Хорошие практики TDD

- Короткие итерации
- Маленькие модули
- Самодокументируемые тесты
- Отделяйте общую настройку окружения от частной для теста

Хорошие практики TDD

- Отдельные наборы тестов должны тестировать отдельные модули
- Учитывайте ошибочные негативные срабатывания при тестировании долгих операций
- Проводите code review
- Пишите красивый и качественный тестовый код

Плохие практики

- Тесты, зависящие от состояния, созданного другими тестами
- Зависимости между тестами
- Точное тестирование производительности или времени выполнения
- Проектирование всезнающих "оракулов"
- Тестирование реализации
- Медленные тесты

Test::Unit

```
require "test/unit"

class TestSequence < Test::Unit::TestCase

  def test_normal
    assert_equal(1, sequence(0, 1, 3))
  end

  def test_small_value
    assert_equal(1, sequence(1, 2, 1))
  end
end
```

Test::Unit

```
assert(false, "This was expected to be true")
assert_equal(expected, actual, text)

assert_block do
  [1, 2, 3].any? { |num| num < 1 }
end

assert_nothing_raised RuntimeError do
  raise Exception #Assertion passes, Exception is  not a RuntimeError
end

assert_nothing_raised do
  raise Exception #Assertion fails
end
```

Test::Unit

```
assert_respond_to("hello", :reverse) #Succeeds
assert_respond_to("hello", :does_not_exist) #Fails


assert_send(["Hello world", :include?, "Hello"]) #-> pass
assert_send(["Hello world", :include?, "Goodbye"]) # -> fail
```

TDD vs BDD

- Модульные тесты VS Specs
- Описываем
- TDD для тестирования модулей
- BDD для тестирования сценариев => интеграционные тесты
- ATDD = BDD + TDD

Терминология BDD

Story/ Feature	История
As a	Как (в роли)
In order to	Чтобы достичь
I want to	Я хочу, чтобы

Scenario	Сценарий
Given	Дано/Пусть
When	Когда
Then	Тогда
And	И
But	Но

Пример истории

Feature: User Authentication Background:
Given the user is already registered to the website
Scenario: Given the user is on the login page
When the user inputs the correct email address
And the user inputs the correct password
And the user clicks the Login button
Then the user should be authenticated
And the user should be redirected to their dashboard
And the user should be presented with a success message

Пример истории

Feature: User Authentication Background:
Given the user is already registered to the website
Scenario: Given the user is on the login page
When the user inputs the ‘test@example.org’ in ‘email’
And the user inputs the ‘strong_pass’ in ‘password’
And the user clicks the ‘login’ button
Then the user should be authenticated
And the user should be redirected to their dashboard
And the user should be presented with a success message

Терминология Test Double

- **Dummy** – объекты, которые используются в качестве формальных параметров для передачи в функции.
- **Fake** – объекты, у которых есть реализация каких-то методов, но она упрощена по сравнению с той реализацией, которая будет использоваться в ПО.
- **Stubs** – сущности, которые предоставляют результаты для определённых обращений (вызовов методов, запросов).
- **Spies** – stubs + статистика обращений.
- **Mocks** – stubs + верификация полученных вызовов: исключение, если вызов не ожидался, подсчет полученных вызовов

Пишем тесты

```ruby
#test_helper.rb
$LOAD_PATH.unshift File.expand_path("../lib", __dir__)
require "equation_solver"

require "minitest/autorun"
```

Пишем тесты

```
test/equation_solver_test.rb
require "test_helper"

class EquationSolverTest < Minitest::Test
  def test_that_it_has_a_version_number
    refute_nil ::EquationSolver::VERSION
  end

  def test_it_does_something_useful
    assert false
  end
end
```

Выполняем тесты

```
$ ruby -Ilib:test test/equation_solver_test.rb
Run options: --seed 32402

# Running:

.F

Failure:
EquationSolverTest#test_it_does_something_useful [test/equation_solver_test.rb:9]:
Expected false to be truthy.


rails test test/equation_solver_test.rb:8
```

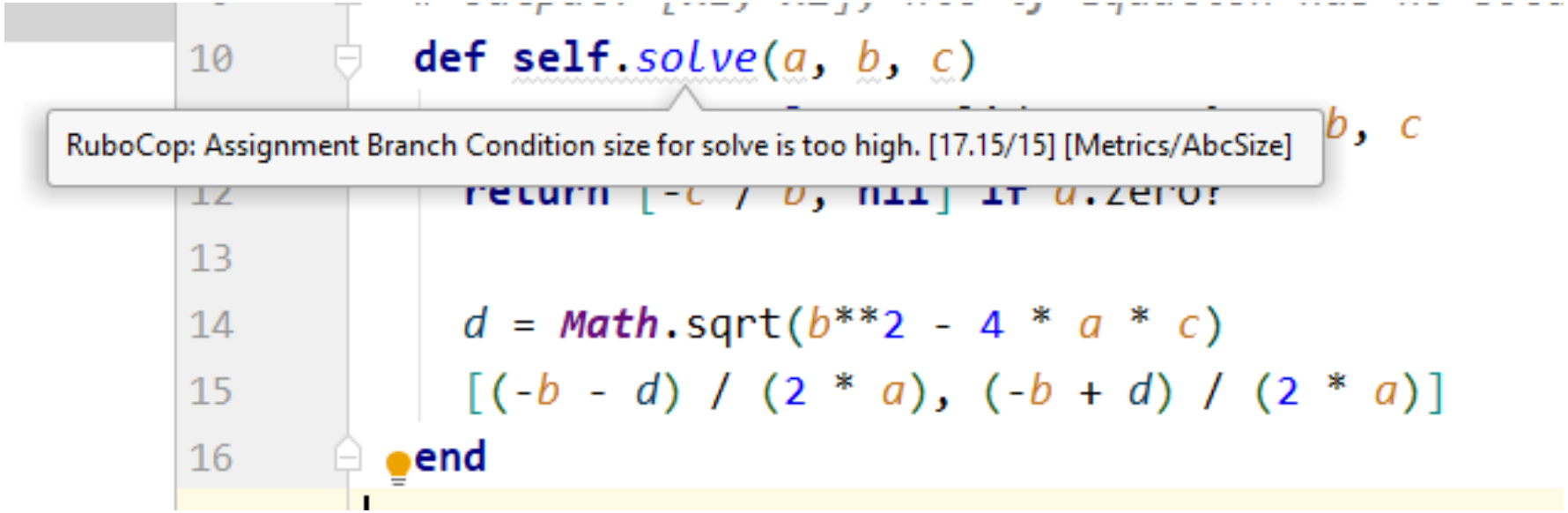
Сначала пишем тесты

```
def test_that_it_handles_wrong_input
  assert_nil ::EquationSolver.solve(0, 1, 'test')
end

def test_that_it_handles_linear_equation
  assert_equal [1, nil], ::EquationSolver.solve(0, 1, -1)
end
```

Потом код

```
def self.solve(a, b, c)
  return nil unless valid_params? a, b, c
  return [-c / b, nil] if a.zero?
  return [- Math.sqrt(a), Math.sqrt(a) ]
                          if b.zero? && c.zero?

  d = Math.sqrt(b**2 - 4 * a * c)
  [(-b - d) / (2 * a), (-b + d) / (2 * a)]
end
```

Тесты проходят, но код не очень



Рефакторинг

```ruby
# equation a*x^2+b*x+c=0
# @return [Numeric, Numeric] or nil if equation has no solutions
def self.solve(a, b, c)
  return nil unless valid_params? a, b, c
  return [solve_linear(b, c), nil] if a.zero?
  return pair(Math.sqrt(a)) if b.zero? && c.zero?

  pair(discriminant(a, b, c)).map{ |x| (x - b) / (2 * a) }.sort
end

# equation k*x+b=0
# @return Numeric, nil if equation has no solutions
def self.solve_linear(k, b)
  return nil if k.zero?

  -b / k
end
```

Рефакторинг

```ruby
def self.discriminant(a, b, c)
 Math.sqrt(b**2 - 4 * a * c)
end

def self.valid_params?(a, b, c)
  [a, b, c].all? { |x| x.is_a? Numeric }
end

def self.pair(x)
  [-x, x]
end

private_class_method :valid_params?
private_class_method :solve_linear
private_class_method :discriminant
private_class_method :pair
```

Ссылки

- rubygems.org
- http://rdoc.sourceforge.net/doc/
- https://www.martinfowler.com/bliki/TestDouble.html
- https://github.com/rubocop-hq/ruby-style-guide