

Язык программирования Ruby, занятие 3

CS 212

13 ноября 2018 г.

План

- 1 Дата и время
- 2 Обработка исключений
- 3 Модули
- 4 Классы
- 5 Gems

Дата и время

```
1 require 'date'  
2  
3 d = DateTime.new(2001,2,3)  
4 d.year  
5 d.month  
6 d.day  
7 d.cweek  
8 d.cwday
```

Листинг 1: Дата и время

Дата и время

```

1 require 'date'
2 Date.parse('2001-02-03')      #=> #<Date:
   2001-02-03 ...>
3 Date.parse('20010203')      #=> #<Date:
   2001-02-03 ...>
4 Date.parse('3rd Feb 2001')   #=> #<Date:
   2001-02-03 ...>
5 Date.strptime('2001-02-03', '%Y-%m-%d') #=>
   #<Date: 2001-02-03 ...>
6 Date.strptime('2001-W05-6', '%G-W%V-%u') #=>
   #<Date: 2001-02-03 ...>
7 Date.strptime('2001 04 6', '%Y %U %w')  #=> #<Date:
   2001-02-03 ...>
8 Date.strptime('sat3feb01', '%a%d%b%y')  #=> #<Date:
   2001-02-03 ...>

```

Обработка исключений

Exception - класс-предок всех исключений

StandardError - дефолтная ошибка для rescue

RuntimeError - дефолтная ошибка для raise

ScriptError, SignalException - не отлавливаются по умолчанию

IOError, RangeError, TypeError - отлавливаются

<https://ruby-doc.org/core-2.4.0/Exception.html>

Обработка исключений

```
1
2 def raise_and_rescue
3   begin
4     puts 'I am before the raise.'
5     raise 'An error has occurred.'
6     puts 'I am after the raise.'
7   rescue
8     puts 'I am rescued.'
9   end
10  puts 'I am after the begin block.'
11 end
```

Листинг 3: Обработка исключений

Никогда не ловите Exception!

```
1 loop do
2   begin
3     sleep 1
4     eval "djsakru3924r9eiuorwju3498 +=
          5u84fior8u8t4ruyf8ihiure"
5   rescue Exception
6     puts "I refuse to fail or be stopped!"
7   end
8 end
```

Листинг 4: Обработка исключений 2

<https://stackoverflow.com/questions/10048173>

Обработка исключений

```
1 begin
2   # ...
3 rescue => e
4   warn e.message
5   raise # re-raise the current exception
6 end
```

Листинг 5: Обработка исключений 3

\$! последнее исключение, глобальная переменная

\$@ backtrace последнего исключения, глобальная переменная

Обработка исключений

```
1 begin
2   # ...
3 rescue ArgumentError => argument_error
4   # handle ArgumentError
5 rescue NameError
6   # handle NameError, TypeError
7 rescue =>
8   # handle any StandardError
9 end
```

Листинг 6: Разные классы исключений

Обработка исключений

```
1 begin
2   # ...
3 rescue
4   # ...
5 else
6   # this runs only when no exception was raised
7 ensure
8   # ...
9 end
```

Листинг 7: Полная форма с else и ensure

Определение модуля

```
1 module Mod
2   include Math
3   CONST = 1
4   def meth
5     # ...
6   end
7 end
8
9 Mod.class           #=> Module
10 Mod.constants      #=> [:CONST, :PI, :E]
11 Mod.instance_methods #=> [:meth]
```

Листинг 8: Модули

Вложенные модули

```
1 module M1
2   module M2
3     $a = Module.nesting
4   end
5 end
6 $a          #=> [M1::M2, M1]
7 $a[0].name  #=> "M1::M2"
```

Листинг 9: Вложенные модули

Предки модуля

```
1 module Mod
2   include Math
3   include Comparable
4   prepend Enumerable
5 end
6
7 Mod.ancestors      #=> [Enumerable, Mod,
8   Comparable, Math]
9 Math.ancestors     #=> [Math]
10 Enumerable.ancestors #=> [Enumerable]
```

Листинг 10: Ancestors

Include vs Prepend vs Extend

```
1 module Logging
2   def log(message)
3     # log in a file
4   end
5 end
6 module Debug
7   def log(message)
8     # debug output
9   end
10 end
11 class Service
12   include Logging
13   include Debug
14 end
15 p Service.ancestors # [Service, Debug, Logging, ...]
```

Include vs Prepend vs Extend

Extend не встраивает модуль в иерархию в классов. Методы подключенного таким образом модуля вызываются как методы класса, в который он был встроен.

Callback для Include

```
1 module A
2   def A.included(mod)
3     puts "#{self} included in #{mod}"
4   end
5 end
6 module Enumerable
7   include A
8 end
```

Листинг 12: Included

Prepend

```
1 module ServiceDebugger
2   def run(args)
3     puts "Service run start: #{args.inspect}"
4     result = super
5     puts "Service run finished: #{result}"
6   end
7 end
```

Листинг 13: Prepend 1

Prepend

```
1 class Service
2   prepend ServiceDebugger
3
4   def run(args)
5     args.each do |arg|
6       sleep 1
7     end
8     {result: "ok"}
9   end
10 end
11
12 > Service.ancestors
13 => [ServiceDebugger, Service, Object, ...]
```

Листинг 14: Prepend 2

Создание классов

```
1
2 def Worker
3   def do_job
4   end
5
6 private
7   def do_secret_job
8   end
9 end
```

Листинг 15: Создание классов

Наследование

```
1
2 def ConstructionWorker < Worker
3   def do_job
4     super
5     do_secret_job #error
6   end
7 end
```

Листинг 16: Наследование

Инициализация

```
1
2 def Worker
3   def initialize
4   end
5 end
6
7 worker = Worker.new
8 hollow_worker = Worker.allocate
9 worker.initialized? #true
10 hollow_worker.initialized #false
```

Листинг 17: Инициализация

Контроль наследования

```
1
2 class Foo
3   def self.inherited(subclass)
4     puts "New subclass: #{subclass}"
5   end
6 end
7
8 class Bar < Foo
9   end
10
11 class Baz < Bar
12   end
```

Листинг 18: Инициализация

Доступ к полям классов

```
1 class Hello
2   attr_reader :hello
3   #attr_writer :hello
4   #attr_accessor :hello
5   def initialize
6     @hello = "hello!"
7   end
8 end
9
10 h = Hello.new
11 h.hello
```

Листинг 19: Доступ к полям

Добавление метода к классу

```
1
2 class DateTime
3   def getlocal
4     self.to_time.getlocal.to_datetime
5   end
6 end
```

Листинг 20: Переобъявление класса

Добавление метода к классу

```
1 define_method(:my_method) do |foo, bar|
2   # do something
3 end
4 define_method(:my_method) do |*args|
5   # do something
6 end
```

Листинг 21: define method

Структура гема

```
gem_name/  
├── bin/  
│   └── gem_name  
├── lib/  
│   └── gem_name.rb  
├── test/  
│   └── test_gem_name.rb  
├── README  
├── Rakefile  
└── gem_name.gemspec
```

Список команд gema

```
1 gem install mygem
2 gem uninstall mygem
3 gem list
4 gem rdoc --all
5 gem fetch mygem
6 gem search STRING --remote
7 gem build mygem.gemspec
```

Листинг 22: Gems

Gemfile и Bundler

```
1 source 'https://rubygems.org'  
2  
3 gem 'rails', '~> 5.1.6'  
4 gem 'mysql2', '>= 0.3.18', '< 0.6.0'  
5 gem 'sass-rails', '~> 5.0'
```

Листинг 23: Gemfile

Gemfile и Bundler

```
1 bundle install
2 bundle update
3 bundle exec rake ... #perform in bundlers context
```

Листинг 24: Bundler