Ruby

A Programmer's Best Friend

Arbeiten mit Ruby - Klassen

Andreas Neumann M.A.

21.05.12

Klassen

- Klassen werden mit dem keyword **class** eingeleitet
- Methoden werden mit **def** **methodenname** innerhalb der Klasse definiert
- Methoden innerhalb der Klasse sind erst nach dem erzeugen der Klasse vorhanden
- Klassen werden durch den Aufruf der Methode **new** erzeugt

```ruby
#!/usr/bin/ruby

class Example

  def hello
    puts "Hello"
  end


end

x = Example.new

p x
x.hello
```

Eine Klasse instanziieren

- Klassen werden mit **new** ins leben gerufen

- **new** entspricht der Methode **initialize**

- Nimmt **initialize** Argumente entgegen, gilt dies auch für **new**

- Klassenvariablen werden mit @ markiert

```ruby
#!/Usr/bin/ruby

class Greeter
  def initialize(name)
    @name=name
  end

  def hello
    puts "Hello: #{@name}"
  end
end

x = Greeter.new("Andi")
puts x.inspect
x.hello
```

```
#<Greeter:0x10c7939e0 @name="Andi">
Hello: Andi
```

Getter, Setter, Accessors

- Um Instanzvariablen von außen zugreifbar oder sogar veränderbar zu machen stehen diese drei Methoden zu Verfügung
  - **attr_reader** :NameDesAttributs
  - **attr_writer** :NameDesAttributs
  - **attr_accessor** :NameDesAttributs

```ruby
#!/Usr/bin/ruby
class Greeter
  attr_accessor :name

  #.. Wie vorher
end

x = Greeter.new("Andi")
puts x.inspect
x.hello
x.name = "Blasius"
puts x.inspect
x.hello
```

```
#<Greeter:0x10785c520 @name="Andi">
Hello: Andi
#<Greeter:0x10785c520 @name="Blasius">
Hello: Blasius
```

inspect

- Alle Klassen implementieren die Methode **inspect**

- Inspect gibt alle Variablen, den Namen und die Referenz auf die aktuelle Klasseninstanz aus

- **p <Klasseninstanz>** entspricht **puts <KLasseninstanz>.inspect**

```
x = Greeter.new("Andi")

puts x.inspect

x.name = "Blasius"

p x
```

```
#<Greeter:0x10785c520 @name="Andi">
Hello: Andi
#<Greeter:0x10785c520 @name="Blasius">
Hello: Blasius
```

Vererbung

- < lässt die linke Klasse von der rechten Klasse erben

- innerhalb eines Methodenaufrufs ruft **super** die Methode der Elternklasse auf

```ruby
class GloriousGreeter < Greeter
  def hello
    super
    puts ",Sir!"
  end

  def hail
    puts "Hail #{name} !"
  end

end

greeter = GloriousGreeter.new("Andi")
greeter.hail
greeter.hello

p greeter
```

```
Hail Andi !
Hello: Andi
,Sir!
#<GloriousGreeter:0x10c7e8da0 @name="Andi">
```

Klasenmethoden / statische Methoden

- normalerweise sind Methoden nur verfügbar, wenn eine Klasse instanziiert wurde

- durch Definition der Methode mit vorangestellten Klasennamen ist die Methode auch verfügbar, wenn die Klasse nicht instaziiert wurde

```ruby
#!/Usr/bin/ruby

class Greeter
  def hello
    puts "Hello: #{@name}"
  end

  def Greeter.bow
    puts "Greeter bows"
  end
end

Greeter.bow
Greeter.hello
```

```
Greeter bows

NoMethodError: undefined method 'hello' for Greeter:Class

at top level  in example_class.rb at line 14
```