

Hafta 6- Ruby

- Metotlar, Sınıflar, Hata Ayıklama

Metotlar

- def ile başlar
- küçük harf ile başlar

```
In [1]: def merhaba_soyle      # non parametric
        puts "Merhaba Pamukkale"
        end
```

```
Out[1]: :merhaba_soyle
```

```
In [2]: merhaba_soyle
```

Merhaba Pamukkale

```
In [7]: def selamla(isim="isimsiz kisi")
        puts "Merhaba #{isim}!"
        end
```

```
Out[7]: :selamla
```

```
In [8]: selamla()
```

Merhaba isimsiz kisi!

```
In [10]: def test(d1="Perl", d2="Ruby")  # varsayılan parametre değerleri belirleme
        puts "Öğrenmediğim Diller: #{d1} ve #{d2}"
        end
```

```
Out[10]: :test
```

```
In [11]: test
```

Öğrenmediğim Diller: Perl ve Ruby

```
In [12]: test "C", "C++"
```

Öğrenmediğim Diller: C ve C++

```
In [14]: test("C")      # metot çağırırken parantez kullanılabilir
```

Öğrenmediğim Diller: C ve Ruby

```
In [15]: def topla(a,b)
        a+b #son ifade geri dönüş değeridir
        end
```

```
Out[15]: :topla
```

```
In [16]: topla 3,5
```

```
Out[16]: 8
```

```
In [17]: def toplam_carpim_bolum(a,b)
          return a+b, a*b, a/b
          end
```

```
Out[17]: :toplam_carpim_bolum
```

```
In [18]: sayi1 = 12
          sayi2 = 3
          puts "Toplam: #{topla sayi1, sayi2}"
```

```
Toplam: 15
```

```
In [19]: t, c, d = toplam_carpim_bolum(sayi1, sayi2)
          puts "Toplam: #{t}, Çarpım: #{c}, bölüm: #{d}"
```

```
Toplam: 15, Çarpım: 36, bölüm: 4
```

```
In [20]: tc = toplam_carpim_bolum sayi1, sayi2
          puts tc # tc değişkeni Array tipindedir. 2 değeri tek değişkene atayamayacağı için-Dikkat
```

```
[15, 36, 4]
```

```
In [ ]:
```

```
In [21]: def yazdir(*args)
          puts(args)
          end
```

```
Out[21]: :yazdir
```

```
In [22]: yazdir 1,2,"merhaba", 45, 60
```

```
[1, 2, "merhaba", 45, 60]
```

```
In [24]: # each_with_index anlama
          a=[11,22,31,224,44]
          a.each_with_index { |value,index| puts "çarpım: #{index*value} -> value: #{value}"}
```

```
çarpım: 0 -> value: 11
çarpım: 22 -> value: 22
çarpım: 62 -> value: 31
çarpım: 672 -> value: 224
çarpım: 176 -> value: 44
```

```
Out[24]: [11, 22, 31, 224, 44]
```

```
In [25]: def multi_yazdir(*args)
          args.each_with_index {|v, i| puts "#{i}. parametre degeri: #{v}"}
          end
```

```
Out[25]: :multi_yazdir
```

```
In [26]: multi_yazdir 1,2,"merhaba"
```

```
0. parametre degeri: 1
1. parametre degeri: 2
2. parametre degeri: merhaba
```

```
Out[26]: [1, 2, "merhaba"]
```

```
In [27]: multi_yazdir 3.14, "dünya!", "", "!", 5
```

```
0. parametre degeri: 3.14
1. parametre degeri: dünya!
2. parametre degeri:
3. parametre degeri: !
4. parametre degeri: 5
```

```
Out[27]: [3.14, "d\u00FCnya!", "", "!", 5]
```

Sınıflar

- class ile başlar end ile biter,
- Sınıf isimler büyük harfle başlar
- initialize metodu yapıcı(constructor) görevindedir.
- her sınıf Class sınıfının birer örneği(instance) olan nesnelerdir.
- Nesneye ait örnek değişkenleri(instance variable) oluşturmak için değişkenin önüne @ sembolü getirilir.
- Nesnelerin örnek değişkenlerine dışarıdan erişebilmek için sınıf tanımlamasında üç farklı özellik kullanılabilir.
- **attr_accessor** okuma ve yazma,
- **attr_reader** sadece okuma
- **attr_writer** sadece yazma için kullanılan özelliklerdir.
- Bu özellikler kullanılmadan örnek değişkenlerine isimleriyle erişmek mümkün değildir

```
In [28]: class Product
end
```

```
In [29]: # Now using above class to create objects
object=Product.new
```

```
Out[29]: #<Product:0x007f888ba97930>
```

```
In [30]: ## Nesnelerin örnek değişkenlerine dışarıdan erişebilmek için sınıf tanımlamasında üç farklı
##- ** attr_accessor ** okuma ve yazma,
##- ** attr_reader ** sadece okuma
##- ** attr_writer ** sadece yazma için kullanılan özelliklerdir.

class Product
  attr_accessor :name, :type, :producer
end
```

```
In [31]: #nesne oluşturalım ve değişkenler değer atayalım x=Product.new x.name=""  
tell=Product.new
```

```
Out[31]: #<Product:0x007f888d16aec8>
```

```
In [32]: tell.name="Iphone"  
tell.type="Mobile"  
tell.producer="Apple"
```

```
Out[32]: "Apple"
```

```
In [33]: # objenin değişken değerine bakalım  
puts tell.name  
puts tell.type  
puts tell.producer
```

```
Iphone  
Mobile  
Apple
```

```
In [34]: puts tell
```

```
#<Product:0x007f888d16aec8>
```

```
In [35]: class Product  
          attr_accessor :name, :type, :producer  
          def to_s  
            puts "Telefon İsmi: #{@name}, Tipi: #{@type}, üretici: #{@producer}"  
          end  
        end
```

```
Out[35]: :to_s
```

```
In [36]: puts tell
```

```
Telefon İsmi: Iphone, Tipi: Mobile, üretici: Apple  
#<Product:0x007f888d16aec8>
```

```
In [37]: tel2=Product.new  
tel2.name="Redmi"  
tel2.type="Mobile"  
tel2.producer="Xiommi"
```

```
Out[37]: "Xiommi"
```

```
In [38]: puts tel2
```

```
Telefon İsmi: Redmi, Tipi: Mobile, üretici: Xiommi  
#<Product:0x007f888c110b40>
```

```
In [39]: tel3=Product.new("Redmi Note 8","S10","Xiommi")  
puts tel3
```

```
ArgumentError: wrong number of arguments (3 for 0)  
<main>:1:in `initialize'
```

```

<main>:1:in `new'
<main>:1:in `<main>'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/backen
d.rb:44:in `eval'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/backen
d.rb:44:in `eval'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/backen
d.rb:12:in `eval'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/kernel.
rb:87:in `execute_request'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/kernel.
rb:47:in `dispatch'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/kernel.
rb:37:in `run'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/comman
d.rb:70:in `run_kernel'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/comman
d.rb:34:in `run'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/bin/iruby:5:in `<
top (required)>'
/Users/ibrahim/anaconda/envs/untitled/bin/iruby:23:in `load'
/Users/ibrahim/anaconda/envs/untitled/bin/iruby:23:in `<main>'

```

In [41]:

```

class Product
  attr_accessor :name, :type, :producer
  def initialize
    puts "yapıcı fonksiyon çalışıyor"
  end
  def to_s
    puts "Telefon İsmi: #{@name}, Tipi: #{@type}, üretici: #{@producer}"
  end
end

```

Out[41]: :to_s

In [43]:

```
tel5=Product.new()
```

yapıcı fonksiyon çalışıyor

Out[43]: #<Product:0x007f888c8c7780>

In [44]:

```
puts tel5
```

Telefon İsmi: , Tipi: , üretici:
#<Product:0x007f888c8c7780>

In []:

In [46]:

```

#
class Product
  attr_accessor :name, :type, :producer
  def initialize (name, type, producer) # yapıcı method
    @name = name
    @type = type
    @producer = producer
  end
  def to_s
    puts "Telefon İsmi: #{@name}, Tipi: #{@type}, üretici: #{@producer}"
  end
end

```

```
end
```

Out[46]: :to_s

```
In [47]: tel6=Product.new("Redmi Note 8","S10","Xiami")
puts tel6
```

Telefon İsmi: Redmi Note 8, Tipi: S10, üretici: Xiami
#<Product:0x007f888c8a4578>

```
In [54]: class Araba
def initialize(marka, model, model_yili)
  @marka = marka
  @model = model
  @model_yili = model_yili
  #@marka, @model, @model_yili = marka, model, model_yili
end

def yazdir
  puts "Araba -> marka: #{@marka}, model: #{@model}, model_yılı: #{@model_yili}"
end

attr_accessor :marka, :model #yazma ve okuma
#attr_reader :model # sadece okuma
attr_writer :model_yili # sadece yazma
end
```

```
In [50]: araba = Araba.new("Volkswagen", "Passat", 2015)
araba.yazdir
```

Araba -> marka: Volkswagen, model: Passat, model_yılı: 2015

```
In [51]: #attr_accessor :marka
araba.marka = "Mercedes"
puts araba.marka
```

Mercedes

```
In [52]: #attr_reader :model # sadece okuma
puts araba.model
```

Passat

```
In [55]: araba.model = "C180"
```

Out[55]: "C180"

```
In [56]: puts araba.marka
puts araba.model
```

Mercedes
C180

```
In [57]: araba.model_yili = 2000
```

Out[57]: 2000

```
In [58]: puts araba.model_yili      ## sadece yazma özelliği var
```

```
NoMethodError: undefined method `model_yili' for #<Araba:0x007f888c82ac78>
<main>:in `<main>'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/backen
d.rb:44:in `eval'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/backen
d.rb:44:in `eval'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/backen
d.rb:12:in `eval'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/kernel.
rb:87:in `execute_request'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/kernel.
rb:47:in `dispatch'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/kernel.
rb:37:in `run'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/comman
d.rb:70:in `run_kernel'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/comman
d.rb:34:in `run'
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/bin/iruby:5:in `<
top (required)>'
/Users/ibrahim/anaconda/envs/untitled/bin/iruby:23:in `load'
/Users/ibrahim/anaconda/envs/untitled/bin/iruby:23:in `<main>'
```

```
In [ ]: #puts araba.a
```

Sınıf değişkenleri ve metotları örneklere değil de sınıfa aittir. Başka programlama dillerindeki static öğeler gibi davranırlar. Sınıf değişkeni oluşturmak için @@ sembolü değişkenin önüne eklenir. Sınıf metodu oluşturmak için ise tanımlama self.metot_adi şeklinde yapılır. to_s metodu nesnenin metin temsilidir.

```
In [59]: class Araba
  @@adet = 0 # sınıf değişkeni

  def initialize(marka, model, model_yili)
    @marka, @model, @model_yili = marka, model, model_yili
    @@adet += 1
  end

  def self.adet_yaz() # sınıf metodu
    puts "Araba sayısı: #{@@adet}"
  end

  def to_s
    "Araba => marka: #{@marka}, model: #{@model}, model yılı: #{@model_yili}"
  end
end
```

Out[59]: :to_s

```
In [60]: a1 = Araba.new "Ferrari", "458", 2010
```

Out[60]: #<Araba:0x007f888c827078 @marka="Ferrari", @model="458", @model_yili=2010>

```
In [61]: a2 = Araba.new "Lamborghini", "Gallardo", 2014
```

```
Out[61]: #<Araba:0x007f888b9faa18 @marka="Lamborghini", @model="Gallardo", @model_yili=2014>
```

```
In [62]: a3 = Araba.new "tesla", "Model S", 2020
```

```
Out[62]: #<Araba:0x007f888b9f32b8 @marka="tesla", @model="Model S", @model_yili=2020>
```

```
In [63]: puts a1,a2,a3
```

```
Araba => marka: Ferrari, model: 458, model yılı: 2010
Araba => marka: Lamborghini, model: Gallardo, model yılı: 2014
Araba => marka: tesla, model: Model S, model yılı: 2020
```

```
In [64]: Araba.adet_yaz
```

```
Araba sayısı: 3
```

```
In [65]: # global değişken $ sembolu ile oluşturulur

$global_variable = 10
class Class1
  def print_localglobal
    #@_lv=5
    puts "Global variable in Class1 is #{$global_variable}"
  end
end

class Class2
  def print_global
    puts "Global variable in Class2 is #{$global_variable}"
  end
end
```

```
Out[65]: :print_global
```

```
In [66]: class1obj = Class1.new
class1obj.print_localglobal
class2obj = Class2.new
class2obj.print_global
```

```
Global variable in Class1 is 10
Global variable in Class2 is 10
```

```
In [ ]: puts $global_variable
```

Kalıtım

- Kalıtım olgusu, bir sınıfta (class) tanımlanmış değişkenlerin ve/veya metotların (fonksiyon, procedure) yeniden tanımlanmasına gerek olmaksızın yeni bir sınıfa taşınabilmesidir.

Ruby programlama dilinde kalıtım < sembolü ile gerçekleştirilir.

- Aşağıda Araba sınıfından türetilen ElektrikliAraba sınıfı vardır. ElektrikliAraba sınıfında to_s metodu geçersiz kılınmıştır.

- #super kelimesi belirttiğimiz parametrelerin ana sınıfın yapıcı metotuna parametre göndermesini sağlayan metottur.
- super() metodunu kullanarak, üst sınıfının bir nesnesini yaratabilir ve onun değişkenlerine değer atayabilir.

In [67]:

```
class Araba
  @@adet = 0 # sınıf değişkeni

  def initialize(marka, model, model_yili)
    @marka, @model, @model_yili = marka, model, model_yili
    @@adet += 1
  end

  def self.adet_yaz() # sınıf metodu
    puts "Araba sayısı: #{@adet}"
  end

  def to_s
    "Araba => marka: #{@marka}, model: #{@model}, model yılı: #{@model_yili}"
  end
end
```

Out[67]: :to_s

In [68]:

```
class ElektrikliAraba < Araba
  attr_accessor :batarya_kapasitesi
  def initialize(marka, model, model_yili, batarya_kapasitesi)
    super marka, model, model_yili
    @batarya_kapasitesi = batarya_kapasitesi
  end

  def to_s
    "#{super.to_s}, batarya kapasitesi: #{@batarya_kapasitesi}"
  end
end
```

Out[68]: :to_s

In [69]:

```
tesla = ElektrikliAraba.new 'Tesla', "Model S", 2018, "60 kWh"
puts tesla
```

Araba => marka: Tesla, model: Model S, model yılı: 2018, batarya kapasitesi: 60 kWh

In [70]:

```
teslaX = ElektrikliAraba.new 'TeslaX', "Model SX", 2018, "70 kWh"
puts teslaX
```

Araba => marka: TeslaX, model: Model SX, model yılı: 2018, batarya kapasitesi: 70 kWh

In [71]:

```
puts ElektrikliAraba.adet_yaz
```

Araba sayısı: 2

HATA YAKALAMA

In [72]:

```
sayi = 1 / 0
```

```
ZeroDivisionError: divided by 0
```

```
<main>:in `/'  
<main>:in `<main>'  
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/backen  
d.rb:44:in `eval'  
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/backen  
d.rb:44:in `eval'  
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/backen  
d.rb:12:in `eval'  
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/kernel.  
rb:87:in `execute_request'  
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/kernel.  
rb:47:in `dispatch'  
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/kernel.  
rb:37:in `run'  
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/comman  
d.rb:70:in `run_kernel'  
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/lib/iruby/comman  
d.rb:34:in `run'  
/Users/ibrahim/anaconda/envs/untitled/lib/ruby/gems/2.2.0/gems/iruby-0.3/bin/iruby:5:in `<  
top (required)>'  
/Users/ibrahim/anaconda/envs/untitled/bin/iruby:23:in `load'  
/Users/ibrahim/anaconda/envs/untitled/bin/iruby:23:in `<main>'
```

In [73]:

```
begin  
  sayi = 1 / 0  
rescue ZeroDivisionError => e  
  puts e.message  
rescue StandardError  
  puts "hata"  
else  
  puts "Hata yok"  
ensure  
  puts "Hep çalışır"  
end
```

```
divided by 0  
Hep çalışır
```

In []: