OOP in R

Alexey Osipov

11.02.2020

Systems of classes in R.

- S3-classes,
- S4-classes,
- RC- or R5-classes,
- R6-classes.

Types of OOP.

- Message-passing OOP (Java, C++, C#).
  A message is sent to an object, an object chooses a method.

```
canvas.drawRect( blue )
```

- Generic-function OOP.
  A generic function decides which method to call.

S3 classes

- Class is determined by class attribute.
- Generic function:

  ```
  UseMethod( mean , x)
  ```

  We decide which mean we call basing on class:

  ```
  mean.numeric, mean.data.frame,
   mean.matrix, mean.default.
  ```

- Inheritance:

  ```
  NextMethod.
  ```

S4 classes

- Class has: name, representation (slots), contains (character vector of classes that it inherits from).
- Create class with setClass, create instance with new, set method with setMethod.
- In S4 we check the types of the slots.
- We access a slot via @ or slot or [[.
- S4 is stricter than S3, but still generic function OOP.
- `setGeneric , setMethod .`

R5 or RC classes

- Message-passing OOP, mutability aka pass by reference.
  An implementation is environments + S4-classes.
- Class is: contains, fields, methods.
- It is possible to add methods after creation.
- `setRefClass , $new() , obj$method()`

  We modify fields with

  `<<-`

Drawbacks of R5 classes.

1. They are slow.
2. They are not portable.
3. No private methods or fields.
4. We detect fields via

   <<-

```
bar <<- 1           #bar is a field

bar <- 1             #bar is a variable
```

R6 classes

- Message-passing OOP, mutability aka pass by reference.
- public, private (aka protected)
- active binding (non-trivial getter)
- `class$new(), obj$method(), obj$field.`
- we detect field via:

  `self$, private$`
- Inheritance.

R6 classes, continued

- Special treatment of fields passed by reference.
- Portable vs non-portable.
- Possibility of class modification after creation.
- Faster and simpler than R5-classes.
- Debug is tricky:

  ```
  class$debug('method')
  ```

  enables debug, then

  ```
  debug(object$method).
  ```