Знакомство со Scala

Седьмое занятие

Неявные преобразования

- Назначение - расширение чужого кода, существующих библиотек
- Активно используются в стандартной библиотеке Scala для адаптации Java
- Контролируются областью видимости
- Применяется только одно
- Система типов Scala – полная по Тьюрингу
- Требуют основательного подхода

Неявные преобразования

```
import java.util._

implicit def func2comp[T](f: (T, T) => Int): Comparator[T] =
  new Comparator[T]{ def compare(x: T, y: T): Int = f(x, y) }

val a = Array("a", "b", "aa", "bb")

Arrays.sort(a, (x: String, y: String) => y.length - x.length)

// a: Array[java.lang.String] = Array(bb, aa, b, a)
```

Неявные преобразования

```
class TimeToMillis(i: Int) {
  def seconds: Long = i * 1000
  def minutes: Long = seconds * 60
  def hours: Long = minutes * 60
}
implicit def int2ttm(i: Int) = new TimeToMillis(i)
println(5.hours + 3.minutes + 2.seconds) // 18182000


import java.text.SimpleDateFormat
val format = new SimpleDateFormat(“dd/MM/yyyy”)
implicit def str2date(s: String) = format.parse(s)
println("11/11/2011".getTime) // 1320948000000
```

Неявные параметры

```
def parseDate(s: String)(implicit f: DateFormat): Date =
  f.parse(s)
implicit val format = new SimpleDateFormat(“dd/MM/yyyy”)
println(parseDate("11/11/2011"))

def findMax[T](l: List[T])(implicit order: T => Ordered[T]):
T = if (l.size == 1) l(0) else {
  val maxInTail = findMax(l.tail)(order)
  if (order(l.head) > maxInTail) l.head
  else maxInTail
}
implicit val myStrOrd = (s: String) => new Ordered[String] {
 def compare(that: String): Int = that.length – s.length
}
println(findMax(List(“aaa”, “aa”, “a”)) // “a”
```

Задание 7-1

```
// java.lang.String
// public static String format(String format, Object... args)
// реализовать:
“Hello, %s!”.fmt(“World”)

// ---------------------------

// java.io.File
// реализовать def foreach(f: String ⇒ Unit): Unit для:
for (line <- new File(“text.txt”)) println(line)
```

Импорт и пакеты

```
package com.mypackage // #1

package com { // #2
package mypackage {
}
}

import java.io.File // #3
import java.util._ // #4
import Arrays._ // #5
import java.util.{Collection, List => JList} // #6
import java.util.{Map => _, _} // #7
```

Пакетные объекты

```
object `package` { // файл com/package.scala
  // ...
}

package object mypackage { // файл com/package.scala
  // ...
}

package com.mypackage {
package util {
  // здесь видны объявления из пакетного объекта
}
}
```

Модификаторы доступа

- Не специфицировано – **public**
- **protected** – доступно в самом классе, inner классах и наследниках
- **private** – доступно в самом классе и inner классах
- **private[X], protected[X]** – дополнительно доступно в X, где X имя пакета, класса или **this**

Модификаторы доступа

```
class Outer {
  class Inner {
    private def f() { println("f") }
    class InnerMost {
      f() // OK
    }
  }
  (new Inner).f() //error
}
```