

Лабораторная работа № 3 «Обобщённые классы в Scala»

18 марта 2024 г.

Сергей Виленский, ИУ9-62Б

Цель работы

Целью данной работы является приобретение навыков разработки обобщённых классов на языке Scala с использованием неявных преобразований типов.

Индивидуальный вариант

Класс `Range[T : Ordering]`, представляющий интервал на множестве значений типа `T` с операциями определения вхождения одного интервала в другой, пересечения двух интервалов и определения принадлежности значения интервалу. В случае, если тип `T` — числовой, дополнительно должна быть реализована операция вычисления длины интервала.

Реализация

```
class Range[T]( val start: T, val stop: T )( implicit ord: Ordering[T] ) {
  require(ord.lteq(start, stop),
    "Range start must be after stop.")

  override def toString: String = s"Range(${start}, ${stop})"

  def hasIntersect( other: Range[T] ) =
    ord.lteq(other.start, this.stop) &&
    ord.gteq(other.stop, this.start)

  def getIntersect( other: Range[T] ): Option[Range[T]] = {
    if (!hasIntersect(other)) {
      None
    } else {
      Some(new Range(

```

```

        ord.min(other.start, this.start),
        ord.max(other.stop, this.stop)
    ))
    }
}

def isIncludes( point: T ) =
    ord.lteq(start, point) &&
    ord.lteq(point, stop)

def getLength( )( implicit ops: RangeOps[T] ) = ops.diff(start, stop)
}

trait RangeOps[T] {
    def diff( a: T, b: T ): T
}

object RangeOps {
    implicit def num_ops[T]( implicit num: Numeric[T] ): RangeOps[T] =
        new RangeOps[T] {
            def diff( a: T, b: T ): T = num.minus(b, a)
        }
}

object Main extends App {
    val r1 = new Range(1, 2)
    val r2 = new Range(5, 6)
    val r3 = new Range(2, 5)
    val r4 = new Range("h", "z")
    val r5 = new Range("a", "i")

    val p1 = 2
    val p2 = 4
    val p3 = "c"
    val p4 = "j"

    println(r1, r2, r1.hasIntersect(r2), r1.getIntersect(r2))
    println(r1, r3, r1.hasIntersect(r3), r1.getIntersect(r3))
    println(r2, r3, r2.hasIntersect(r3), r2.getIntersect(r3))
    println(r4, r5, r4.hasIntersect(r5), r4.getIntersect(r5))

    println(r1, p1, r1.isIncludes(p1))
    println(r2, p1, r2.isIncludes(p1))
    println(r1, p2, r1.isIncludes(p2))
    println(r2, p2, r2.isIncludes(p2))
    println(r4, p3, r4.isIncludes(p3))
}

```

```

println(r5, p3, r5.isIncludes(p3))
println(r4, p4, r4.isIncludes(p4))
println(r5, p4, r5.isIncludes(p4))

println(r1, r1.getLength())
println(r2, r2.getLength())
println(r3, r3.getLength())
// не скомпилируется
// println(r4, r4.getLength())
}

```

Тестирование

Результат запуска программы:

```

(Range(1, 2), Range(5, 6), false, null)
(Range(1, 2), Range(2, 5), true, Range(1, 5))
(Range(5, 6), Range(2, 5), true, Range(2, 6))
(Range(h, z), Range(a, i), true, Range(a, z))
(Range(1, 2), 2, true)
(Range(1, 2), 4, false)
(Range(5, 6), 4, false)
(Range(h, z), c, false)
(Range(a, i), c, true)
(Range(h, z), j, true)
(Range(a, i), j, false)
(Range(1, 2), 1)
(Range(5, 6), 1)
(Range(2, 5), 3)

```

Вывод

В результате выполнения данной работы были приобретены навыки разработки обобщённых классов на языке Scala с использованием неявных преобразований типов.