

Лабораторная работа № 2 «Введение в объектно-ориентированное программирование

на языке Scala» % 1 марта 2023 г. % Яровикова Анастасия, ИУ9-61Б

Цель работы

Целью данной работы является изучение базовых объектно-ориентированных возможностей языка Scala.

Индивидуальный вариант

Множество, конструируемое как множество строк, содержащих некоторую строку *s*. Операции: объединение («+»), пересечение («*»), проверка принадлежности строки множеству («in»).

Реализация

```
def removeDups(s: List[String]): List[String] = {  
  var st = s.toSet.toList  
  st  
}  
  
class mySet(sstr: String, st: List[String]) {  
  val str = sstr  
  val set = st  
  
  def this(sstr: String) = this(sstr, List())  
  
  def in(s: String): Boolean = {  
    if (s.indexOf(str) != -1)  
      true  
    else  
      false  
  }  
}
```

```

        false
    }

    def +(z: mySet) = {
        var resStr: String = set(0)
        if (str.indexOf(z.str) != -1)
            resStr = z.str
        if (z.str.indexOf(str) != -1)
            resStr = str
        var ls: List[String] = List()
        for (el <- set)
            if (in(el) || z.in(el))
                ls = ls :+ el
        for (el <- z.set)
            if (in(el) || z.in(el))
                ls = ls :+ el
        ls = removeDups(ls)
        var sumSet = new mySet(resStr, ls)
        sumSet
    }

    def *(z: mySet) = {
        var tmp: String = ""
        var len: Int = 999
        for (el <- set)
            if (z.in(el)) {
                if (el.length() < len) {
                    len = el.length()
                    tmp = el
                }
            }
        var ls: List[String] = List()
        for (el <- set)
            if (in(el) && z.in(el))
                ls = ls :+ el
        ls = removeDups(ls)
        var inSet = new mySet(tmp, ls)
        inSet
    }
}

object Main {
    def main(args: Array[String]) = {
        var st = new mySet("ab")
        println(s"Set generated by \"${st.str}\"\\n\\nin method testing:")
        println(s"abracadabra: ${st.in("abracadabra")}")
    }
}

```

```

println(s"alibaba: ${st.in("alibaba"}}")
println(s"ali: ${st.in("ali"}}")
println("\nunion and intersection testing:")
var s1 = new mySet("sun", List("sun", "sunny"))
var s2 = new mySet("sunny", List("sunny", "sunnyDays"))
println(s"Set1: ${s1.set}, formed from string \"${s1.str}\"")
println(s"Set2: ${s2.set}, formed from string \"${s2.str}\"")
var sum = s1 + s2
println(s"Set1 + Set2: ${sum.set}, formed from string \"${sum.str}\"")
var inter = s1 * s2
println(s"Set1 * Set2: ${inter.set}, formed from string \"${inter.str}\"")
s1 = new mySet("a", List("a", "ab", "cad"))
s2 = new mySet("b", List("b", "ab", "dab"))
println(s"\nSet1: ${s1.set}, formed from string \"${s1.str}\"")
println(s"Set2: ${s2.set}, formed from string \"${s2.str}\"")
sum = s1 + s2
println(s"Set1 + Set2: ${sum.set}, formed from string \"${sum.str}\"")
inter = s1 * s2
println(s"Set1 * Set2: ${inter.set}, formed from string \"${inter.str}\"")
s1 = new mySet("a", List("a", "aa", "cad"))
s2 = new mySet("b", List("b", "bb", "dab"))
println(s"\nSet1: ${s1.set}, formed from string \"${s1.str}\"")
println(s"Set2: ${s2.set}, formed from string \"${s2.str}\"")
sum = s1 + s2
println(s"Set1 + Set2: ${sum.set}, formed from string \"${sum.str}\"")
inter = s1 * s2
println(s"Set1 * Set2: ${inter.set}, formed from string \"${inter.str}\"")
}
}

```

Тестирование

Set generated by "ab"

in method testing:

abracadabra: **true**

alibaba: **true**

ali: **false**

union and intersection testing:

Set1: List(sun, sunny), formed from string "sun"

Set2: List(sunny, sunnyDays), formed from string "sunny"

Set1 + Set2: List(sun, sunny, sunnyDays), formed from string "sun"

Set1 * Set2: List(sunny), formed from string "sunny"

Set1: List(a, ab, cad), formed from string "a"

```
Set2: List(b, ab, dab), formed from string "b"
Set1 + Set2: List(cad, a, ab, dab, b), formed from string "a"
Set1 * Set2: List(ab), formed from string "ab"

Set1: List(a, aa, cad), formed from string "a"
Set2: List(b, bb, dab), formed from string "b"
Set1 + Set2: List(cad, a, dab, b, aa, bb), formed from string "a"
Set1 * Set2: List(), formed from string ""
```

Вывод

В ходе данной лабораторной работы было проведено ознакомление с базовыми объектно-ориентированными возможностями языка программирования Scala, получен опыт разработки классов. Также были изучены возможности контейнерных классов, доступных в стандартной библиотеке языка Scala.