**第46届世界技能大赛移动应用开发项目中国集训队**

**集训日志（选手） 朱姚飞**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **9-28** | **指导老师** | **叶重涵** |
| **训练任务** | **List Set Map 迭代器 OOP** | | | |
| **训练内容：**  List Set Map 迭代器的使用 和 Kotlin 中接口 的继承和抽象方法的重写 按钮的实现  **训练要求：**  使用Kotlin 编写代码  **过程记录：**  **import java.util.\*  fun code1() {  val listA = *listOf*<String>("1", "2", "3")  listA.*plus*("123")  System.*out*.println(listA);  val mapA = *mapOf*<String, String>("a" *to* "b", "c" *to* "d")  val a: (it: Int) -> Int = {  it \* 2  }  System.*out*.println(a(2))  val s = *mutableListOf*<String>("1", "2", "3", "4")  val sl: List<String> = s  System.*out*.println(sl)  s.add("5");  System.*out*.println(sl)  val sll = sl.*plus*("123")  System.*out*.println(sll)  System.*out*.println(s) }  fun code2() {   val listA = *MutableList*<Int>(100) { Math.abs((Random().nextInt() % 100 + 1)) }  val listB = *MutableList*<Int>(100) { (1..100).*random*() }  System.*out*.println(listA)  System.*out*.println(listB)   val big10s = *mutableListOf*<Int>();   for (i in listA) {  big10s.add(i)  }  System.*out*.println(big10s)   val qqs = *listOf*<String>("12345", "123456", "12345678901", "123456789", "123456", "12345678")  val qqsNew = qqs.*toSet*().*toList*();  val iterator = qqsNew.iterator()  while (iterator.hasNext()) {  System.*out*.println(iterator.next())  }  for (s in qqsNew) {  System.*out*.println(s)  }  val ks= *listOf*<String>("1","2","3","4","5")  val vs= *listOf*<String>("a","b","c","d","e")   val resultMap= *mutableMapOf*<String,String>()   for (index in 0 *until* ks.*count*()){  resultMap[ks[index]]=vs[index];  }  System.*out*.println(resultMap); }  fun code3(){  System.*out*.println(*mapOf*("10" *to* "2","30" *to* "3").*toList*().*unzip*())  System.*out*.println( *listOf*("123","234","3451","123","1","2","345234","12341234","1").*groupingBy*(keySelector = { it.length }).*eachCount*()) }  interface InternalA<T> {  var state:T;  fun onClick():T }  abstract class Click<A> :InternalA<A> {  fun click():A{  return this.onClick()  } }  fun main(args: Array<String>) {   val a=object: Click<Int>() {  override var state=1  override fun onClick():Int {  state+=1  System.*out*.println("you click me !");  return state  }  }  System.*out*.println(a.click())  System.*out*.println(a.click())  System.*out*.println(a.click())  System.*out*.println(a.click())  System.*out*.println(a.click()) }**  **分析总结：**  Kotlin 强大的语法糖 | | | | |

**填写人：朱姚飞**