Kotlin

集合

- Java集合

世上本无集合(只有数组)

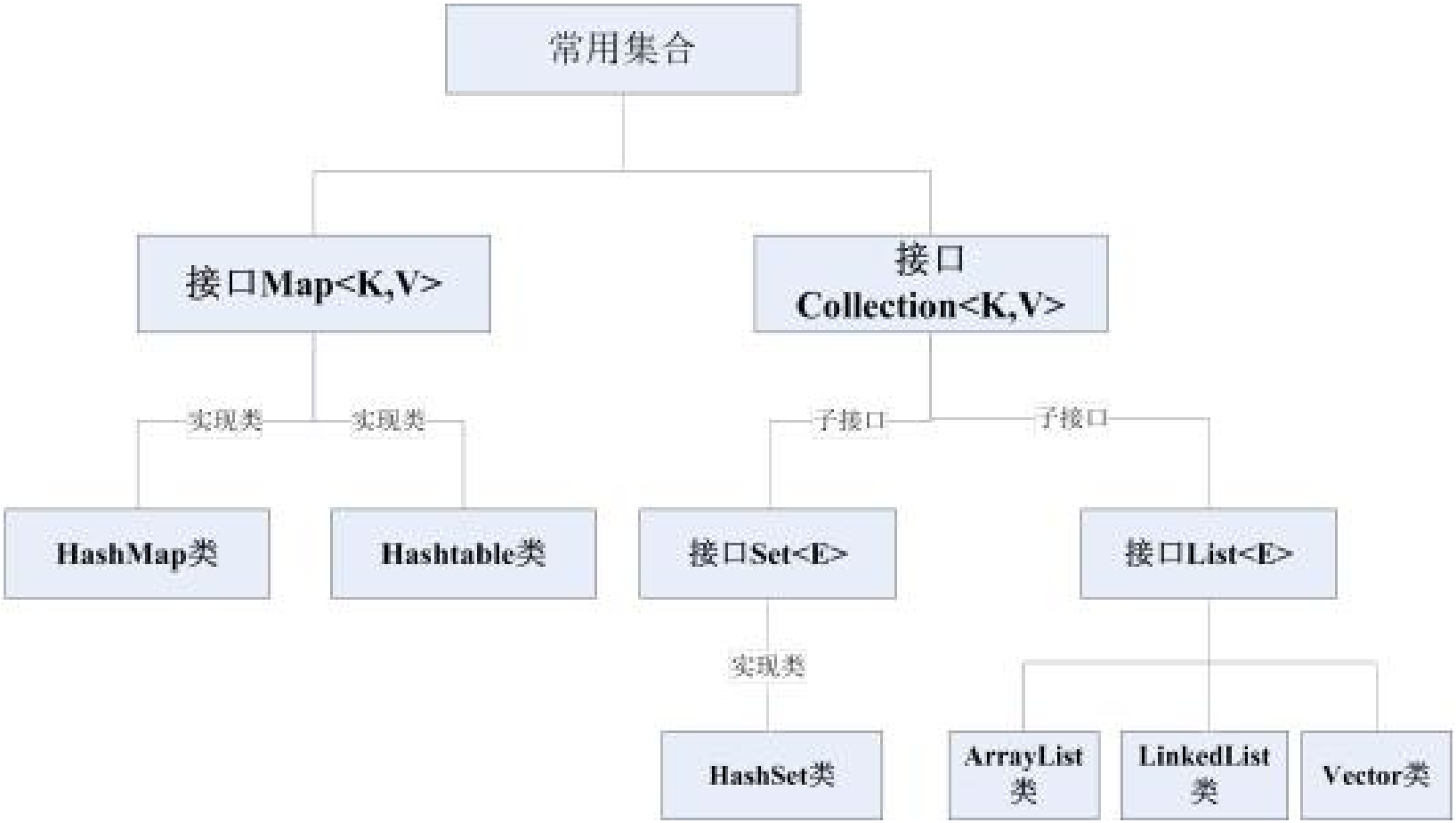
但有人想要,于是就用数组创造了集合类

有人想要可以自动扩展容量的数组,于是有了List

有人想要元素不重复的数组,于是有了Set

有人想要有序的数组,于是有了TreeSet

有人想要通过复杂对象(而不仅仅是简单的下标)来查找另一个对象的关联数组,于是有了Map


常用集合
接口Map<K,V>
接口
Collection<K,V>
实现类
实现类
子接口
子接口
HashMap类
Hashtable类
接口Set<E>
接口List<E>
实现类
HashSet类
ArrayList
类
LinkedList
类
Vector类

- Set集合和List集合实现了Collection接口
- List元素可以重复,Set集合元素不能重复
- ArrayList线程不安全 效率高,Vector线程安全 效率低
- HashMap线程不安全 效率高,HashTable线程安全 效率低

函数式编程

函数式编程

- 函数式编程是现代编程语言一个重要的特性JDK1.8
- “函数式编程”又称”泛函编程” ,是一种”编程范式”
- 面向对象编程 面向过程编程 函数式编程
- OOP(Object Oriented Programming)
- FP(Functional Programming)

- OOP编程:一切皆对象
- FP编程:一切皆是函数
- Kotlin的函数和对象一样都是属于”一等公民”,可以像对象一样作为函数的输入和输出
- FP编程是OOP编程的有效补充,不能代替OOP编程

闭包

闭包

- Python:一个函数返回了一个内部函数，该内部函数引用了外部函数的相关参数和变量，我们把该返回的内部函数称为**闭包**
- **在kotlin里面我们通常说的闭包就是lambda表达式**

高阶函数

高阶函数

- 将函数作为参数或者刷新返回一个函数叫高阶函数

```
fun cacl(a:Int,b:Int,action:(Int,Int)->Int):Int{
        return action(a,b)
}
```

Lambda表达式

Lambda表达式

- Lambda就是匿名函数

  调用全面的高阶函数

```
sum = newCacl(a,b,{m,n->
        m+n
})
```

去()的Lambda表达式

调用全面的高阶函数

```
sum = newCacl(a,b){m,n->
        m+n
}
```

无参的Lambda表达式

```
{
    println("执行了lambda表达式")
}()
```

有参的Lambda表达式

```
{a:Int,b:Int->
    println("a+b=$a+b")
}(10,20)
```

Lambda表达式返回值

```
val result1 = {
        println("其他语句")
        println("其他语句")
        "字符串“
}()
```

Lambda表达式

- 通过变量保存lambda

  val function ={a:Int,b:Int->a+b}

  调用 function(10,20)

- 如果Lambda参数只有一个,可以使用it表示

```
var result = add(a){
        it + 10
}
```

- Lambda访问外部变量

```
var a = 10
{
   println(a)
}()
```

常见的lambda表达式

- Foreach
- Array.indexof

非诚勿扰练习

第一位男嘉宾

- 俺是河南里,俺只找河南的妹子

第二位男嘉宾

- 我只喜欢30岁以下的女生

第三位男嘉宾

- 我喜欢广东的,身高162以上,25岁以下的妹子

第四位男嘉宾

- 我喜欢广东的,身高160以上,35岁以上的妹子

List集合

- 过滤
- 排序
- 分组
- 最值
- 去重
- 拆分
- joinToString

```
val list = listOf("张三","李四","王五","赵六","张四","李五","李六")
val list2 = listOf("周芷若","张无忌","张五","李善长","林青霞","李寻欢")
```

过滤(find和filter)

- 找到第一个姓张的
- 把第一个集合中所有姓张的找出来
- 把两个集合中所有姓张的找到并存放在同一个集合中
- 把第一个集合中角标为偶数的元素找出来

排序

- 正序排序
- 倒序排序
- 按字段排序

分组

- 姓张的一组 姓李的一组 其他一组

最値

- 最大値
- 最小値
- 対象最大値
- 対象最小値

去重复(*distinctBy或者toSet*)

- 把重复的张三去掉
- 把重复的姓张的去掉

集合拆分(*partition*)

- 姓张的一部分,另外的一部分

集合重新组合(map)

- 将Person里面每一个name获取

元素相加(sumby/sumbydouble)

- 求出所有人的年龄之和

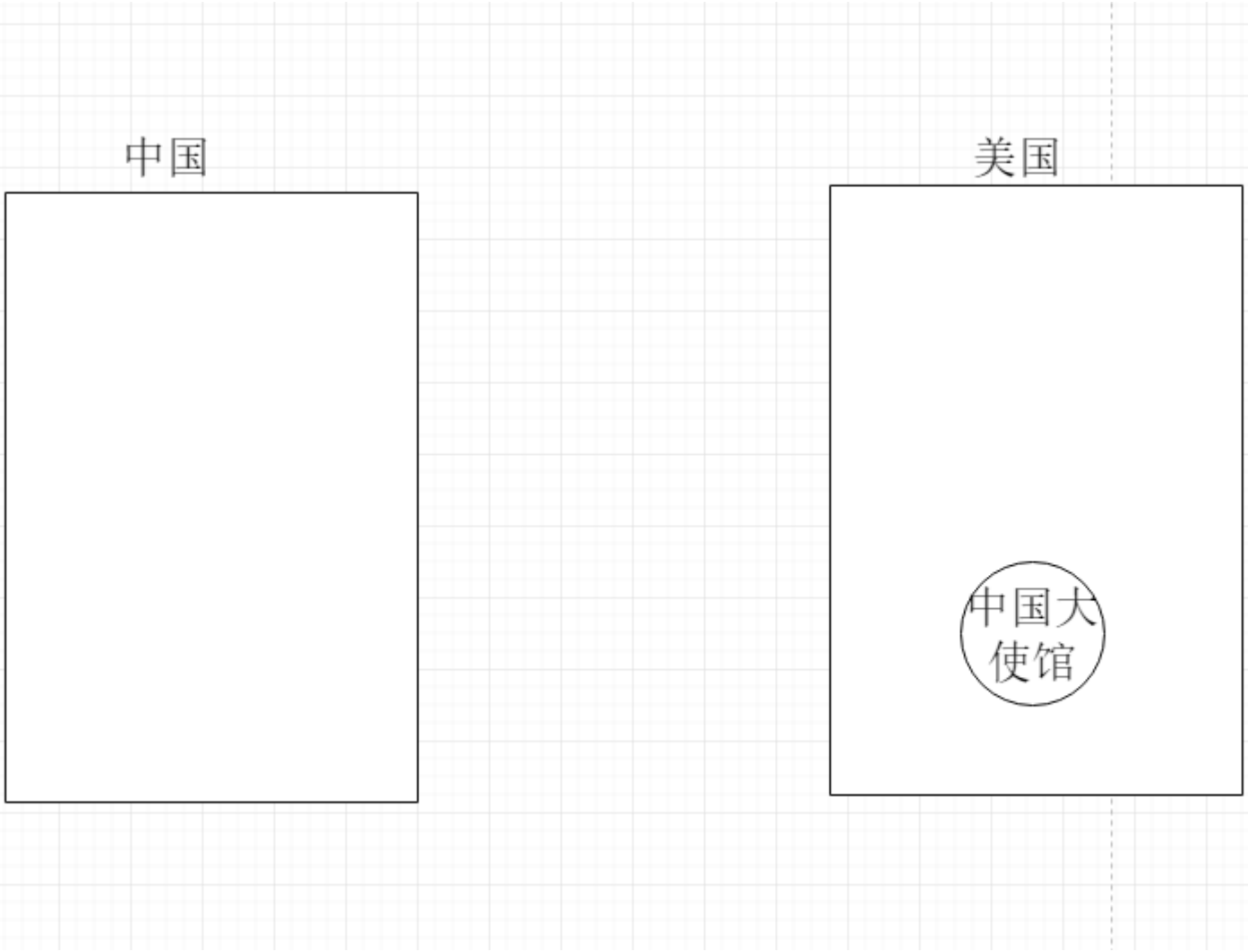
四大表达式

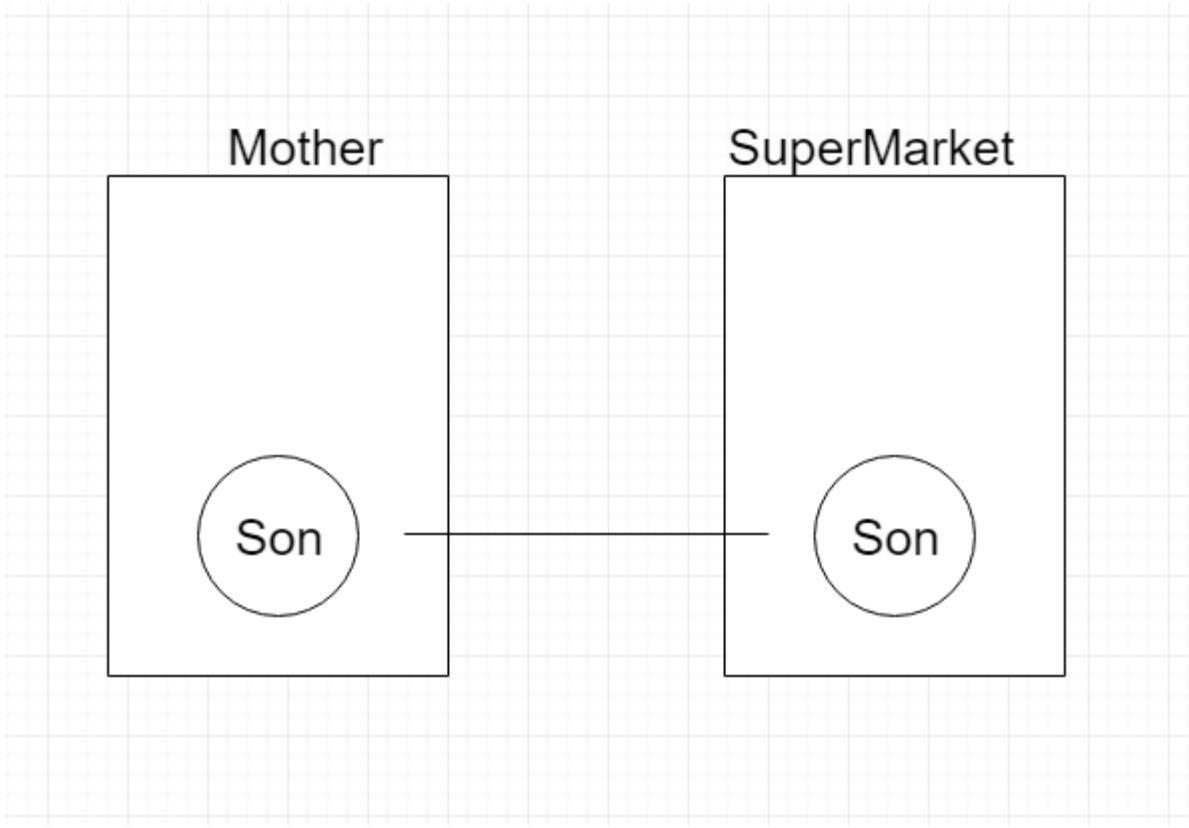
- Apply(使用this代表当前对象,返回值是当前对象)
- Let(太用it代表当前对象,返回值是表达式最后一行)
- With(使用this代表当前对象,返回值是最后一行)
- Run(代码块)

接口回调

面向对象回顾

- 妈妈中午做饭要去超市买一瓶酱油


中国
美国
中国大
使馆


Mother
SuperMarket
Son
Son

接口回调实现

- 在Mother中定义接口类型
- 在Mother中创建接口对象,并传递给SuperMarket
- 在SuperMarket中调用Mother传递的接口对象

函数回调

- 函数代替接口实现对象间通信