



Kotlin

Full Course

Bài 21



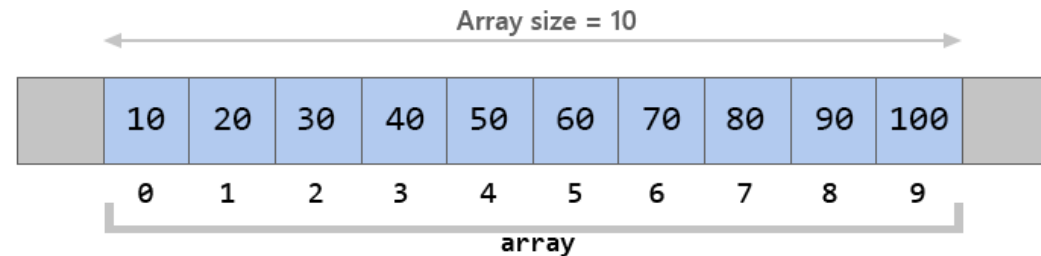
*Lập trình
Android*

1

Mảng 1 chiều Kotlin

2

Bài tập kotlin 23



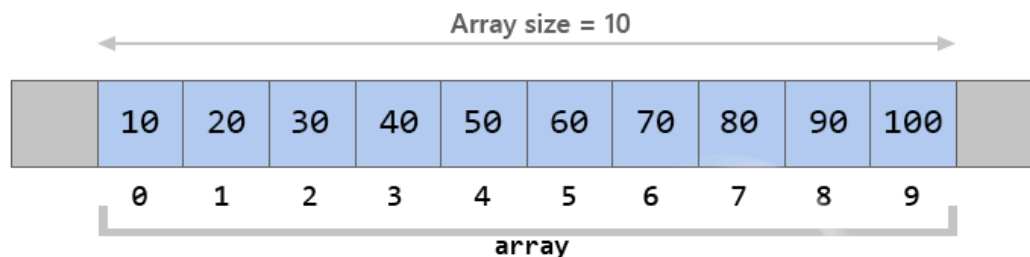
1

Mảng 1 chiều Kotlin

❑ 1. Khái niệm : *Mảng là tập hợp các đối tượng có cùng kiểu dữ liệu.*

Chúng có kích thước cố định không thể thay đổi

Mảng có index bắt đầu từ 0



❑ 2 . Tại sao phải dùng mảng:

Ví dụ : Chúng ta có khoảng 50 điểm của học sinh cần lưu, nếu không dùng mảng thì chúng ta phải khai báo 50 biến float

=> Gộp nhóm các đối tượng có chung tính chất lại với nhau giúp code gọn gàng hơn.



1

Mảng 1 chiều Kotlin

3. Khai báo mảng :

Tên mảng

Kiểu dữ liệu

Số phần tử của mảng

```
var M:IntArray = IntArray( size: 5) //mảng 5 pt là các số 0
var M2:FloatArray = FloatArray( size: 4) //mảng 4 pt là các số 0
println(M::class.java.typeName) // int[]
```

```
//duyet mảng
for (i in M)
    print("$i\t")
```



0 0 0 0 0

4. Khởi tạo và gán giá trị cho mảng:

```
//khai báo
var M3:IntArray = intArrayOf(0,2,-5,1,3)
//khai báo tắt (kotlin tự nội suy)
var M4 = intArrayOf(1,2,3,4)
var M5 = arrayOf("tuhoc","kotlin","python")
println(M4::class.java.typeName) // int[]
```



1

Mảng 1 chiều Kotlin

❑ 5. Truy xuất, thay đổi phần tử của mảng qua vị trí index :

```
//truy xuất ptu qua vị trí index
var M6= intArrayOf(9,8,7,6,5)
println(M6[0]) //9
println(M6[1]) //8
println(M6[2]) //7
```

```
//thay đổi giá trị của mảng qua index
M6[0]=1
M6[1]=1
M6[2]=3
println(M6[0]) //1
println(M6[1]) //1
println(M6[2]) //3
```

❑ 6. Khởi tạo mảng ngẫu nhiên :

```
var rd = Random
var M7:IntArray = IntArray( size: 7)
println(M7.indices) // trả về đoạn 0..6
//duyet từng phần tử mảng
for (i in M7.indices)
{
    //println(i)
    M7[i] = rd.nextInt( until: 100)
}
```

```
//xuất mảng cách 1
for (j in M7)
    print("$j\t")
```

```
//xuất mảng cách 2
for (j in M7.indices)
    print("${M7[j]}\t")
```

```
95 74 98 29 34 49 27
95 74 98 29 34 49 27
```



1

Mảng 1 chiều Kotlin

□7. Size : trả về số phần tử của mảng

```
println("Số phần tử của mảng 7 là:" + M7.size)
```



Số phần tử của mảng 7 là:7

□8. Phép gán mảng :

**** gán mảng mới = mảng đã tồn tại** (tham chiếu cùng vùng nhớ)

**chú ý : khi thay đổi giá trị của 1 mảng, thì mảng còn lại cũng thay đổi ,
// do mảng là kiểu tham chiếu, nó sẽ tham chiếu đến 1 vùng nhớ.**

```
var M9 = arrayOf(1,2,3,4,5)
var M10=M9
M9[0]=99
println(M9[0]) // 99
println(M10[0]) //99
```

□9. Clone (tạo ra mảng mới trên vùng nhớ mới)

```
var M11 = arrayOf(1,2,3,4,5)
var M12=M11.clone()
M11[0] =99 // thay đổi phần tử index 0 = 99
println(M11[0]) // 99
println(M12[0]) //1
```



1

Mảng 1 chiều Kotlin

❑ 10 . Array.reverse() : Đảo ngược mảng

```
var M13 = arrayOf(1,5,8,9)
M13.reverse()
println("Mảng 13 sau khi đảo: ")
for (i in M13)
    print("$i\t")
```



Mảng 13 sau khi đảo:
9 8 5 1

❑ 11. Array.Sort() (Sắp xếp mảng tăng dần)

```
//12.sắp xếp
var M14 = arrayOf(1,6,13,9)
M14.sort()
println("Mảng 14 sau sxep: ")
for (i in M14)
    print("$i\t")
println()
```



Mảng 14 sau sxep:
1 6 9 13



1

Mảng 1 chiều Kotlin

❑ 12. `arr.filter()` : Trích lọc mảng theo điều kiện=> trả về 1 ArrayList

```
var M15 = arrayOf(1,3,5,2,8,10)
var ds = M15.filter { x->x!=null }
println(ds) //[1, 3, 5, 2, 8, 10]
println(ds::class.java.typeName) // ArrayList
//lọc các số chẵn
var ds2 = M15.filter { y->y%2==0 }
println(ds2) // [2, 8, 10]
```

❑ 13. Tìm max, min

```
//tìm max min
val arr2: Array<Int> = arrayOf(6, 3, 2, 5, 10)
arr2.sort();
println("Minimum: ${arr2.first()}") // Minimum: 2
println("Minimum: ${arr2[0]}") // Minimum: 2
println("Maximum: ${arr2.last()}") // Maximum: 10
println("Maximum: ${arr2[arr2.size-1]}") // Maximum: 10
```



2

Bài tập kotlin 23

- ✓ 1. *Viết chương trình tạo 1 mảng 1 chiều gồm các phần tử là số nguyên, có n phần tử ngẫu nhiên, n do người dùng nhập từ bàn phím*
- ✓ 2. *Xuất các giá trị trong mảng*
- ✓ 3. *Đảo ngược mảng, và xuất mảng sau khi đảo ngược*
- ✓ 4. *Sắp xếp mảng tăng dần*
- ✓ 5. *Tính tổng các phần tử trong mảng*
- ✓ 6. *Cho người dùng nhập 1 số bất kỳ, kiểm tra số đó có tồn tại trong mảng hay không, nếu có xuất ra vị trí index của số đó trong mảng*

