

Table of Contents

1.Before you begin	2
2.Roll random number	2
3.Create a dice class.....	2
4. Return your dice roll's value.....	3
5. Change the number of sides on your dice	3
6.Customize your dice	3
7. Adopt good coding practices	4
8.Solution code	4
9.Summary.....	4
10.Learn more.....	4
11.Practice on your own	4

1.Before you begin

2.Roll random number

```
fun main() {  
    val diceRange = 1..6  
    val randomNumber = diceRange.random()  
    println("Random number: ${randomNumber}")  
}
```

Random number: 3

In ra một số bất kì trong khoảng

3.Create a dice class

```
class Dice {  
    var sides = 6  
  
    fun roll() {  
        val randomNumber = (1..6).random()  
        println(randomNumber)  
    }  
}
```

1

4. Return your dice roll's value

```
println("Xúc xắc có ${myFirstDice.sides} mặt của bạn đã tung ra số 5!")
}
```

```
class Dice {
    var sides = 6
    fun roll(): Int {
        val randomNumber = (1..6).random()
        return randomNumber
    }
}
```

Xúc xắc có 6 mặt của bạn đã tung ra số 5!

Chương trình nhận về giá trị random và in ra màn hình

5. Change the number of sides on your dice

```
fun main() {
    val myFirstDice = Dice()
    val diceRoll = myFirstDice.roll()
    println("Xúc xắc có ${myFirstDice.sides} mặt của bạn đã tung ra số ${diceRoll}!")
    myFirstDice.sides = 20
    println("Xúc xắc có ${myFirstDice.sides} mặt của bạn đã tung ra số ${myFirstDice.roll()}!")
}
```

```
class Dice {
    var sides = 6
    fun roll(): Int {
        val randomNumber = (1..sides).random()
        return randomNumber
    }
}
```

Xúc xắc có 6 mặt của bạn đã tung ra số 5!
Xúc xắc có 20 mặt của bạn đã tung ra số 18!

Thay đổi số mặt bất kì cho 1 xúc sắc

6. Customize your dice

```
fun main() {
    val myFirstDice = Dice(6)
    println("Xúc xắc có ${myFirstDice.numSides} mặt của bạn đã tung ra số ${myFirstDice.roll()}!")
    val mySecondDice = Dice(20)
    println("Xúc xắc có ${mySecondDice.numSides} mặt của bạn đã tung ra số ${mySecondDice.roll()}!")
}
```

```
class Dice(val numSides: Int) {
    fun roll(): Int {
        val randomNumber = (1..numSides).random()
        return randomNumber
    }
}
```

Xúc xắc có 6 mặt của bạn đã tung ra số 4!
Xúc xắc có 20 mặt của bạn đã tung ra số 2!

Thay đổi lớp Dice để có thể chỉ định số mặt khi tạo một thể hiện mới. Thay đổi định nghĩa của lớp Dice để chấp nhận số mặt.

7. Adopt good coding practices

```
println("Xúc xắc có ${myFirstDice.numSides} mặt của bạn đã tung ra số ${myFirstDice.roll()}!")
val mySecondDice = Dice(20)
println("Xúc xắc có ${mySecondDice.numSides} mặt của bạn đã tung ra số ${mySecondDice.roll()}!")
}

class Dice(val numSides: Int) {
    fun roll(): Int {
        return (1..numSides).random()
    }
}

Xúc xắc có 6 mặt của bạn đã tung ra số 5!
Xúc xắc có 20 mặt của bạn đã tung ra số 7!
```

Việc thay đổi mã để làm cho nó ngắn gọn hơn, hiệu quả hơn, hoặc dễ đọc và hiểu hơn được gọi là tái cấu trúc mã (refactoring). Nó giống như việc viết một tài liệu, bạn viết bản nháp đầu tiên có đầy đủ thông tin, sau đó chỉnh sửa và cải tiến các câu từ của mình.

8.Solution code

```
fun main() {
    val myFirstDice = Dice(6)
    println("Xúc xắc có ${myFirstDice.numSides} mặt của bạn đã tung ra số ${myFirstDice.roll()}!")
    val mySecondDice = Dice(20)
    println("Xúc xắc có ${mySecondDice.numSides} mặt của bạn đã tung ra số ${mySecondDice.roll()}!")
}

class Dice(val numSides: Int) {
    fun roll(): Int {
        return (1..numSides).random()
    }
}
```

9.Summary

10.Learn more

11.Practice on your own