

## 題目 1uuuuu

一個簡單的遊戲，規則如下:玩家質兩個骰子，點數為 1~6，如果第一次點數和為 7 或 11，則玩家勝，如果點數和為 2,3 或 12，則玩家輸，如果和為其他點數，則記錄第一次的點數和，然後繼續擲骰，直到點數和等於第一次擲出的點數和，則玩家勝，如果在這之前擲出了點數和為 7，則玩家輸。

### 執行需求步驟

#### Step1

提示玩家按 “enter”開始遊戲

```
C:\UI>javac -encoding utf-8 CrapEx.java
C:\UI>java CrapEx
擲骰子遊戲，按enter鍵開始遊戲****
```

#### Step2

擲骰子，玩家輸，按任意鍵繼續玩

```
第一次玩家擲出的點數和:9
未分勝負，在擲一次****
第二次玩家擲出點數和:5
玩家輸
在玩一次?y
第一次玩家擲出的點數和:2
玩家輸
在玩一次?y
第一次玩家擲出的點數和:6
未分勝負，在擲一次****
第二次玩家擲出點數和:5
玩家輸
在玩一次?y
第一次玩家擲出的點數和:6
未分勝負，在擲一次****
第二次玩家擲出點數和:3
玩家輸
在玩一次?y
第一次玩家擲出的點數和:5
未分勝負，在擲一次****
第二次玩家擲出點數和:6
玩家輸
在玩一次?
```

#### Step3

擲篩子，一直到玩家勝為止，按 n 結束遊戲

```
第二次玩家擲出點數和:9
玩家勝
在玩一次?y
第一次玩家擲出的點數和:8
未分勝負，在擲一次****
第二次玩家擲出點數和:8
玩家勝
在玩一次?y
第一次玩家擲出的點數和:5
未分勝負，在擲一次****
第二次玩家擲出點數和:3
玩家輸
在玩一次?y
第一次玩家擲出的點數和:7
玩家勝
在玩一次?第一次玩家擲出的點數和:7
玩家勝
在玩一次?n
遊戲結束
```

參考解答:

```
import java.io.IOException;

class CrapEx
{
    public static void main(String[] args) throws IOException{

        final int WAN=0,Lost=1,Mid=2;
        boolean flag=true;
        int Outcome=Mid;
        int sum;
        int point=0;
        System.out.println("擲骰子遊戲，按 enter 鍵開始遊戲*****");
        while(true)
        {
            System.in.read();
            if(flag){
                sum=getPoint();
```

```
System.out.println("第一次玩家擲出的點數和:"+sum);
switch(sum)
{
    case 7:
    case 11:
        Outcome=WAN;
        break;
    case 2:
    case 3:
    case 12:
        Outcome=Lost;
        break;
    default:
        flag=false;
        Outcome=Mid;
        point=sum;
        break;
}

}
else
{
    sum=getPoint();

    System.out.println("\n 第二次玩家擲出點數和:"+sum);
    if(sum==point)
    {
        Outcome=WAN;
    }
    else
    {
        Outcome=Lost;
    }
}

if(Outcome==Mid)
{
```

```

        System.out.println("未分勝負，在擲一次****");
    }
    else
    {
        if(Outcome==WAN)
        {
            System.out.println("玩家勝");
        }
        else
        {
            System.out.println("玩家輸");
        }
        System.out.print("在玩一次?");

        if(System.in.read()=='n')
        {
            System.out.println("遊戲結束");
            break;
        }
        flag=true;
    }
}

}

public static int getPoint()
{
    int roll=((int)(Math.random()*6)+(int)(Math.random()*6));
    if(roll<2)
    {
        roll=2;
    }

    return roll;
}

}

```