第三章 Java的循环语句

1、while循环语句

```
import java.lang.Math;
import java.util.Scanner;

public class Test1
{
    public static void main(String[] args)
    {
        int i = 0;
        int sum1 = 0;
        while(i<=100)
        {
            sum1+=i;
            i+=3;
        }
        System.out.print(sum1);
    }
}
```

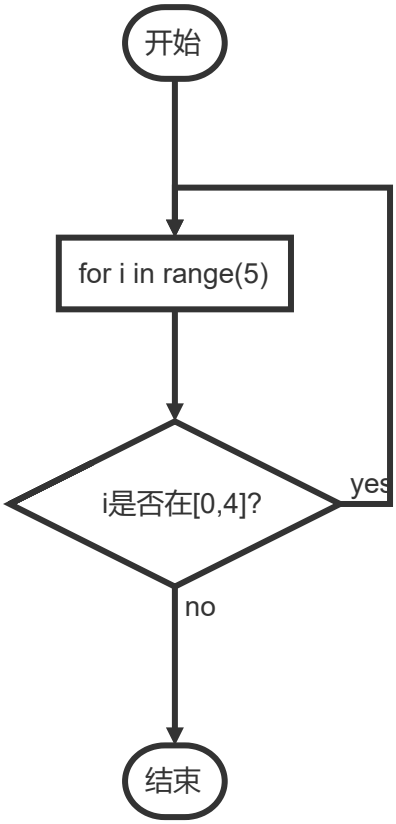
2、for循环语句

1、判断1~100之间自然数，哪些数能够被7整除。

```
import java.lang.Math;
import java.util.Scanner;

public class Test2
{
    public static void main(String[] args)
    {
        int sum1 = 0;
        for(int i=0;i<=100;i+=3)
        {
            sum1+=i;
        }
        System.out.print(sum1);
    }
}
```

3、do-while循环语句




```
import java.lang.Math;
import java.util.Scanner;

public class Test3
{
    public static void main(String[] args)
    {
        int i =0;
        int sum1 = 0;
        do
        {
            sum1+=i;
            i++;
        }while(i<=100);
        System.out.print(sum1);
    }
}
```

4、Math.random()产生随机数。

```
import java.lang.Math;
import java.util.Scanner;

public class Test4
{
    public static void main(String[] args)
    {
        int number = (int)(Math.random()*51+50); //[50,100]
        int i = 0;
```

```
        int guess;
        Scanner sc = new Scanner(System.in);
        while(i<=8)
        {
            guess = sc.nextInt();
            if(number==guess)
            {
                System.out.print(number);
                break;
            }
            else
            {
                System.out.print("Error");
                i++;
            }
        }
    }
}
```

5、猜数字游戏(对半法log_2^n)。

```
import java.util.Scanner;

public class Test5
{
    public static void main(String[] args)
    {
        int number = (int)(Math.random()*51+50); //[50,100]
        int i = 0;
        int guess;
        Scanner sc = new Scanner(System.in);
        while(i<=8)
        {
            guess = sc.nextInt();
            if(number>guess)
            {
                System.out.print("small\n");
                i++;
            }
            else if(number<guess)
            {
                System.out.print("big\n");
                i++;
            }
            else
            {
                System.out.print(number);
                break;
            }
        }
    }
}
```

6、静态函数的定义

```java
import java.lang.Math;
import java.util.Scanner;

public class Test6
{
    public static int max_number(int a, int b)
    {
        if(a>b)
        {
            return a;
        }
        else
        {
            return b;
        }
    }

    public static int max_number2(int a, int b)
    {
        return (a>b)?a:b;
    }
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int number1 = sc.nextInt();
        int number2 = sc.nextInt();
        System.out.print(max_number2(number1,number2));
    }
}
```

7、定义静态函数（递归方法）Fibonacci数列

$$F(n)=\begin{cases}1 & n=1\\ 1 & n=2\\ F(n-1)+F(n-2) & (n\geqslant 3)\end{cases}$$

```java
import java.lang.Math;
import java.util.Scanner;

public class Test6
{
    public static int fibonacci(int n)
    {
        if(n==1)
        {
            return 1;
        }
        else if(n==2)
        {
            return 1;
        }
        else
        {
            return fibonacci(n-1)+fibonacci(n-2);
        }
    }
```

```
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int number1 = sc.nextInt();
        System.out.print(fibonacci(number1));
    }
}
```

8、利用递归方法定义

$$\prod_{i=1}^{10}$$

$$\sum_{i=1}^{10}$$

```
import java.lang.Math;
import java.util.Scanner;

public class Test6
{
    public static int fibonacci(int n)
    {
        if(n==1)
        {
            return 1;
        }
        else if(n==2)
        {
            return 1;
        }
        else
        {
            return fibonacci(n-1)+fibonacci(n-2);
        }
    }

    public static int product(int n)
    {
        if(n==1)
        {
            return 1;
        }
        else
        {
            return product(n-1)*n;
        }
    }


    public static int sigma(int n)
    {
        if(n==1)
        {
            return 1;
        }
        else
```

```
        {
            return sigma(n-1)+n;
        }
    }

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int number1 = sc.nextInt();
        System.out.println(product(number1));
        System.out.println(sigma(number1));
    }
}
```

9、Switch-case

```
import java.lang.Math;
import java.util.Scanner;

public class Test7
{
    public static char grade(int n)
    {
        char grade;
        switch(n)
        {
            case 10:
                grade ='A';
            case 9:
                grade = 'A';
            case 8:
                grade = 'B';
                break;
            case 7:
                grade = 'C';
                break;
            case 6:
                grade = 'D';
                break;
            default:
                grade = 'E';
        }
        return grade;
    }


    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        double grade = sc.nextDouble();
        System.out.print(grade((int)(grade/10)));
    }
}
```

10、最大公约数和最小公倍数

```
import java.lang.Math;
import java.util.Scanner;

public class Test8
{
    public static int gcd(int m, int n)
    {
        int gcd =1;
        int k=2;
        while(k<=m && k<=n)
        {
            if(m%k==0 && n%k==0)
            {
                gcd =k;
            }
            k++;
        }
        return gcd;
    }

    public static int lcm(int m, int n)
    {
        return (m*n)/gcd(m,n);
    }

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int a = sc.nextInt();
        int b = sc.nextInt();
        System.out.println(gcd(a,b));
        System.out.println(lcm(a,b));
    }
}
```

11、最大公约数和最小公倍数（循环取余数法）

```
import java.lang.Math;
import java.util.Scanner;

public class Test9
{
    public static int gcd(int m, int n)
    {
        int r = m%n;
        while(r!=0)
        {
            m = n;
            n = r;
            r = m%n;
        }
        return n;
    }
    //m=48,n=18;
    //r=12;
    //m =18;
    //n=12;
```

```
    //r=6;
    //m=12;
    //n=6;
    //r=0;
    //return n=6;

    public static int lcm(int m, int n)
    {
        return (m*n)/gcd(m,n);
    }

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int a = sc.nextInt();
        int b = sc.nextInt();
        System.out.println(gcd(a,b));
        System.out.println(lcm(a,b));
    }
}
```