

一. **参数**: 方法体中哪些部分是经常改变的量

**返回值**: 调用方法时, 是否需要获得方法体中的某个值.

**return**: 1. 返回值 2. 跳出

方法 (return 后的代码是不会执行的)

```
//练习:定义方法实现,交换两个点 的坐标
//创建2个point对象
Point point1=new Point();
point1.x=3;
point1.y=4;
point1.print();

Point point2=new Point();
point2.x=5;
point2.y=6;
point2.print();

exemcise.swapPoiny(point1,point2);
```

二. 练习:随机为数组的元素赋值[m, n]

```
1  int[] a=new int[10];
2  exemcise.randomArray(a,10,20);
3  exemcise.pp(a);
```

```
1  对数组进行排序,升序降续
2  exemcise.array(a,false);
3  exemcise.pp(a);
```

```
1  求n个数的最大值
2  exemcise.ee(a);
3  System.out.println();
```

三.冒泡排序

```

*/
public void randomArray(int[] b, int m, int n) {
    Random random = new Random();
    for (int i = 0; i < b.length; i++) {
        b[i] = random.nextInt( bound: n - m + 1) + m;
    }
}

```

```

/**
 * 对数组进行冒号排序
 *
 * @param c    要排序的数组
 * @param asc  true代表升序, false代表降序
 */

```

```

public void array(int[] c, boolean asc) {
    for (int i = 0; i < c.length - 1; i++) {
        for (int j = 0; j < c.length - 1 - i; j++) {
            if (asc) {
                if (c[j] > c[j + 1]) {
                    int temp = c[i];
                    c[i] = c[j + 1];
                    c[j + 1] = temp;
                }
            } else {
                if (c[j] < c[j + 1]) {
                    int temp = c[i];
                    c[i] = c[j + 1];
                    c[j + 1] = temp;
                }
            }
        }
    }
}

```

```
    }  
    } else {  
        if (c[j] < c[j + 1]) {  
            int temp = c[j];  
            c[j] = c[j + 1];  
            c[j + 1] = temp;  
        }  
    }  
}  
}  
}
```

练习:求2个整数的和

练习:求3个整数的和

练习:求n个整数的和

//...代表可选参数,可以传,也可以不传,也可以传多个值  
//a就是数组

```
public int maxNPro(int... nums) {  
    int max = nums[0];  
    for (int i = 0; i < nums.length; i++) {  
        if (nums[i] > max) {  
            max = nums[i];  
        }  
    }  
    return max;  
}
```

//可选参数必须放在参数列表的最后

//public void test(int... a, int b) {}

//public void test(int b, int... a) {}

```
public int he(int... n) {  
    int result = 0;  
    for (int i = 0; i < n.length; i++) {  
        result += n[i];  
    }  
    return result;  
}
```

练习:使用递归完成, 1-n的和

练习:使用递归完成, 求费纳妾数列第n项的值

$f(1)=1;$

$f(2)=1;$

$f(n)=f(n-1)+f(n-2);$

System.out.println(exemcise.sunpro1(20));

练习:九九乘法表

## 四. 递归

解答:

//递归:在方法的内部调用本身

//注:递归一定要有出口!!!

```
// public void test(){  
////     System.out.println("辉哥最帅");  
////     test();  
//// }
```

```
public int sumpro(int n) {  
    if (n == 1) {  
        return 1;  
    } else {  
        return sumpro(n - 1) + n;  
    }  
}
```

```
public int sunpro1(int n) {  
    if (n == 1 || n == 2) {  
        return 1;  
    } else {  
        return sunpro1(n - 1) + sunpro1(n - 2)  
    }  
}
```

```
public void m(int i) {  
    if (i == 1) {  
        System.out.println("1x1=1");  
    } else {  
        m(i - 1);  
        for (int i1 = 1; i1 < i; i1++) {  
            System.out.printf("%dx%d=%d\t", i1, i, i1 * i);  
        }  
        System.out.println();  
    }  
}
```