**每日作业卷**

**java基础5天方法**

传智播客.黑马程序员

# 基础案例

## 训练案例1

### 训练描述

定义一个方法 能接受一个整数(这个整数大于3)打印0到这个整数(包含)之间的所有的偶数

如 接受的数字是6则调用完方法打印出来的偶数是 0 2 4 6

如 接受的数字是 5则调用完方法打印出来的偶数是 0 2 4

### 操作步骤描述

1. 创建一个测试类,在测试类中创建上述方法
2. 定义一个无返回值,有参数的方法
3. 在方法内部把符合条件的数字打印
4. 在主方法中调用这个方法,并传入数字15进行测试

**public static void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);  
 System.***out***.print(**"请输入一个大于3的整数："**);  
 **int** num = sc.nextInt();  
 *m1*(num);  
 System.***out***.println();  
 System.***out***.println(**"------------------------------"**);  
}  
  
**public static void** m1(**int** a){  
 **for** (**int** i = 0; i <= a; i++) {  
 **if** (i % 2 == 0){  
 System.***out***.print(i + **" "**);  
 }  
 }

}

## 训练案例2

### 训练描述

随机产生两个整数,随机数的范围均是[1,100],定义方法求这两个整数的和并打印和值

### 操作步骤描述

1. 创建一个测试类
2. 在主方法中使用Random产生两个范围是[1,100]的随机数
3. 定义一个求两个整数和的方法
4. 在主方法中调用这个求和方法得到结果值,并打印

**public static void** main(String[] args) {  
 Random r = **new** Random();  
 **int** m = r.nextInt(100)+1;  
 **int** n = r.nextInt(100)+1;  
 System.***out***.println(m);  
 System.***out***.println(n);  
 *m2*(m, n);  
 System.***out***.println(**"\n"** + **"------------------------------"**);  
}  
**public static void** m2(**int** a, **int** b){  
 System.***out***.println(**"两个整数的和是:"** + (a + b));  
}

## 训练案例3

### 训练描述

有如下数组int[] arr={10,20};

定义一个方法求两个数字的和(方法参数为两个int类型),使用这个方法求出数组这两个元素相加的结果并打印

### 操作步骤描述

1. 创建一个测试类
2. 定义个方法可以求两个整数的和
3. Arr数组写在主方法中,在主方法中调用求和方法求出两个元素的和并打印

**public static void** main(String[] args) {  
 **int**[] arr = {10, 20};  
 *m3*(arr[0], arr[1]);  
 System.***out***.println(**"\n"** + **"------------------------------"**);  
}  
  
**public static void** m3(**int** a, **int** b){  
 **int** sum = a + b;  
 System.***out***.println(**"两个元素的和是："** + sum);  
}

## 训练案例4

### 训练描述

主方法中给定数组int[] arr= {10,20,30,40,50,60};定义一个方法可以接受这个给定的数组

并返回这个数组中元素的最小值

### 操作步骤描述

1. 创建一个测试类,在测试类中写该方法
2. 定义一个方法可以接收一个整型的一维数组

3.在方法内部求出数组元素的最小值,并把这个最小值返回

4.在主方法中调用上述方法传递数组arr,得到数组中元素的最小值并打印

**public static void** main(String[] args) {  
 **int**[] arr1 = {10, 20, 30, 40, 50, 60};  
 **int** min = *m4*(arr1);  
 System.***out***.println(**"数组中元素的最小值是："** + min);  
 System.***out***.println(**"\n"** + **"------------------------------"**);  
}  
**public static int** m4(**int**[] arr) {  
 **int** temp = 0;  
 **for** (**int** i = 0; i < arr.**length**-1; i++) {  
 **if** ( arr[i] < arr[i+1]){  
 temp = arr[i];  
 arr[i] = arr[i+1];  
 arr[i+1] = temp;  
 }  
 }  
 **return** arr[arr.**length**-1];  
}

## 训练案例5

### 训练描述

有定义一个方法,可以接受一个整数,

如果为[90,100]这个范围方法返回’A’

如果为[80,90)这个范围方法返回’B’

如果为[70,80)这个范围方法返回’C’

如果为[60,70)这个范围方法返回’D’

如果为[0,60)这个范围方法返回’E’

整数不在以上范围的返回’F’

### 操作步骤描述

1.创建一个测试类

2.定义个方法根据传递进来的数字返回对应的字母

3.在主方法中使用键盘录入一个整数,调用这个方法得到这个整数对应的字母并打印

**public static void** main(String[] args) {  
 Scanner sc = **new** Scanner(System.***in***);  
 System.***out***.print(**"请输入一个(0-100)整数："**);  
 **int** choice = sc.nextInt();  
 System.***out***.print(*m5*(choice));  
 System.***out***.println(**"\n"** + **"------------------------------"**);  
}  
**public static char** m5(**int** a){  
 **if** (a<=100&&a>=90){  
 **return 'A'**;  
 } **else if** (a>=80 && a<90){  
 **return 'B'**;  
 } **else if** (a>=70 && a<80){  
 **return 'C'**;  
 } **else if** (a>=60 && a<70){  
 **return 'D'**;  
 } **else if** (a>=0 && a<60){  
 **return 'E'**;  
 } **else  
 return 'F'**;  
}

# 扩展案例

## 训练案例1

### 训练描述

主方法中给定数组int[] arr= {10,20,30,40,50,60};这个数组没有重复元素.定义一个方法可以接受这个给定的数组并返回这个数组中最大元素值的索引值

**public static void** main(String[] args) {  
 **int**[] arr2= {10,20,30,40,50,60};  
 System.***out***.println(*m6*(arr2));  
 System.***out***.println(**"------------------------------"**);  
}  
**public static int** m6(**int**[] a){  
 **int** max = a[0], index = 0;  
 **for** (**int** i = 1; i < a.**length**; i++) {  
 **if** (a[i] > max){  
 max = a[i];  
 index = i;  
 }  
 }  
 **return** index;  
}

### 操作步骤描述

略

## 训练案例2

### 训练描述

分析以下需求，并用代码实现

1.键盘录入长方形的长和宽

定义方法计算该长方形的周长,并在main方法中打印周长

2.键盘录入长方形的长和宽

定义方法计算该长方形的面积,并在main方法中打印面积

3.键盘录入圆的半径

定义方法计算该圆的周长,并在main方法中打印周长

4.键盘录入圆的半径

定义方法计算该圆的面积,并在main方法中打印面积

**public static void** main(String[] args) {  
 Scanner sc = **new** Scanner(System.***in***);  
 System.***out***.println(**"(0：计算长方形的周长;1:计算长方形的面积;2:计算圆形的周长;3:计算圆的面积)"**);  
 System.***out***.print(**"请输入您要计算的类型："**);  
 **int** choice1 = sc.nextInt();  
 **if** (choice1 == 0 || choice1 == 1) {  
 System.***out***.print(**"请输入长方形的长:"**);  
 **int** length = sc.nextInt();  
 System.***out***.print(**"请输入长方形的宽:"**);  
 **int** width = sc.nextInt();  
 System.***out***.println(*m7*(choice1, length, width));  
 } **else if** (choice1 == 2 || choice1 == 3) {  
 System.***out***.print(**"请输入圆形的半径:"**);  
 **int** radius = sc.nextInt();  
 System.***out***.println(*m8*(choice1, radius));  
 }  
 System.***out***.println(**"------------------------------"**);  
 }  
  
**public static int** m7(**int** c, **int** len, **int** wid) {  
 **int** result = 0;  
 **if** (c == 0) {  
 result = (len + wid) \* 2;  
 } **else if** (c == 1) {  
 result = len \* wid;  
 }  
 **return** result;  
 }  
  
 **public static double** m8(**int** c, **int** r) {  
 **double** result = 0;  
 **if** (c == 2){  
 result = 2 \* 3.14 \* r;  
 } **else if** (c == 3){  
 result = 3.14 \* r \* r;  
 }  
 **return** result;  
 }

### 操作步骤描述

略

## 训练案例3

### 训练描述

分析以下需求，并用代码实现

1.定义一个方法equals(int[] arr1,int[] arr2),功能:比较两个数组是否相等(长度和内容均相等则认为两个数组是相同的)

2.定义一个方法fill(int[] arr,int value),功能:将数组arr中的所有元素的值改为value

3.定义一个方法fill(int[] arr,int fromIndex,int toIndex,int value),功能:将数组arr中的元素从索引fromIndex开始到toIndex(不包含toIndex)对应的值改为value

4.定义一个方法copyOf(int[] arr, int newLength),功能:将数组arr中的newLength个元素拷贝到新数组中,并将新数组返回,从索引为0开始

5.定义一个方法copyOfRange(int[] arr,int from, int to),功能:将数组arr中从索引from(包含from)开始到索引to结束(不包含to)的元素复制到新数组中,并将新数组返回

**public static void** main(String[] args) {  
 **int**[] arr3 = {1, 2, 3, 4};  
 **int**[] arr4 = {1, 2, 3, 4, 5};  
 *equals*(arr3, arr4);  
 System.***out***.println(**"------------------------------"**);  
  
 **int**[] arr5 = {1, 3, 5, 7, 9};  
 *fill*(arr5, 9);  
 System.***out***.println(**"\n"** + **"------------------------------"**);  
  
 **int**[] arr6 = {2, 4, 6, 8, 10};  
 *fill1*(arr6, 2, 4, 6);  
 System.***out***.println(**"\n"** + **"------------------------------"**);  
  
 *copyOf*(arr6, 3);  
 System.***out***.println(**"\n"** + **"------------------------------"**);  
  
 **int**[] arr7 = *copyOfRange*(arr6, 0, 3);  
 System.***out***.println(**"按索引拷贝之后的数组是："**);  
 **for** (**int** i = 0; i < arr7.**length**; i++) {  
 System.***out***.print(arr7[i] + **" "**);  
 }  
}  
  
*/\*  
 定义一个方法equals(int[] arr1,int[] arr2)  
 功能:比较两个数组是否相等(长度和内容均相等则认为两个数组是相同的)  
\*/***public static void** equals(**int**[] arr1, **int**[] arr2) {  
 **if** (arr1.**length** == arr2.**length**) {  
 **for** (**int** i = 0; i < arr1.**length**; i++) {  
 **if** (arr1[i] != arr2[i]) {  
 System.***out***.println(**"两个数组不相等"**);  
 **return**;  
 }  
 }  
 System.***out***.println(**"两个数组相等"**);  
 } **else** System.***out***.println(**"两个数组不相等"**);  
}  
  
*/\*  
 定义一个方法fill(int[] arr,int value)  
 功能:将数组arr中的所有元素的值改为value  
 \*/***public static void** fill(**int**[] arr, **int** value) {  
 **for** (**int** i = 0; i < arr.**length**; i++) {  
 arr[i] = value;  
 }  
 System.***out***.println(**"替换之后的数组是："**);  
 **for** (**int** i = 0; i < arr.**length**; i++) {  
 System.***out***.print(arr[i] + **" "**);  
 }  
}  
  
*/\*  
 定义一个方法fill(int[] arr,int fromIndex,int toIndex,int value)  
 功能:将数组arr中的元素从索引fromIndex开始到toIndex(不包含toIndex)对应的值改为value  
 \*/***public static void** fill1(**int**[] arr,**int** fromIndex,**int** toIndex,**int** value) {  
 **for** (**int** i = 0; i < arr.**length**; i++) {  
 **if** (i >= fromIndex && i <= toIndex) {  
 arr[i] = value;  
 }  
 }  
 System.***out***.println(**"替换之后的数组是："**);  
 **for** (**int** i = 0; i < arr.**length**; i++) {  
 System.***out***.print(arr[i] + **" "**);  
 }  
}  
  
*/\*  
 定义一个方法copyOf(int[] arr, int newLength)  
 功能:将数组arr中的newLength个元素拷贝到新数组中,并将新数组返回,从索引为0开始  
 \*/***public static void** copyOf(**int**[] arr, **int** newLength) {  
 **int**[] newArr = **new int**[newLength];  
 **for** (**int** i = 0; i < newLength; i++) {  
 newArr[i] = arr[i];  
 }  
 System.***out***.println(**"拷贝之后的数组是："**);  
 **for** (**int** i = 0; i < newArr.**length**; i++) {  
 System.***out***.print(newArr[i] + **" "**);  
 }  
}  
  
*/\*  
 定义一个方法copyOfRange(int[] arr,int from, int to)  
 功能:将数组arr中从索引from(包含from)开始到索引to结束(不包含to)的元素复制到新数组中,并将新数组返回  
 \*/***public static int**[] copyOfRange(**int**[] arr,**int** from, **int** to) {  
 **int**[] newArr = **new int**[to-from];  
 **for** (**int** i = 0; i < to; i++) {  
 newArr[i] = arr[i];  
 }  
 **return** newArr;  
}

### 操作步骤描述

略

## 训练案例4

### 训练描述

定义一个方法,去打印指定两个年份之间所有的闰年年份

如:将2000和2005传入方法,方法执行完毕会打印出这段之间的所有闰年年份

2000和2004

提示:

计算公历年闰年的算法: 四年一闰，百年不闰，四百年再闰

翻译:

满足如下任意一条即使如年

1. 年份是整百数的必须是400的倍数才是闰年(能被100整除并且能被400整除)
2. 其他年份能被4的是闰年

实例: 2000 是整百数,并且是400的倍数所以是闰年; 2004年是4的倍数是闰年

2100 是整百的倍数但不是400的倍数,所以不是闰年

**public static void** main(String[] args) {  
 *leap*(2000, 2005);   
}**public static void** leap(**int** a, **int** b) {  
 **int** temp = 0;  
 **if** (a < b) {  
 temp = a;  
 a = b;  
 b = temp;  
 }  
 System.***out***.println(a + **"和"** + b + **"之间所有的闰年年份是："**);  
 **for** (**int** i = b; i <= a; i++) {  
 **if** (i % 4 == 0 || (i % 400 == 0 && i % 100 == 0)) {  
 System.***out***.print(i + **" "**);  
 }  
 }  
}

### 操作步骤描述

略

## 训练案例5

### 训练描述

1.键盘录入一个整数(正数或者负数都可以,但是符号位不算有效的数字位)

2.定义一个方法,该方法的功能是计算该数字是几位数字,并将位数返回

3.在main方法中打印该数字是几位数

4.演示格式如下:

(1)演示一:

请输入一个整数:1234

控制台输出:1234是4位数字

(2)演示二:

请输入一个整数:-34567

控制台输出:-34567是5位数字

**public static void** main(String[] args) {  
 Scanner sc = **new** Scanner(System.***in***);  
 System.***out***.print(**"请输入一个整数："**);  
 **int** num = sc.nextInt();  
 System.***out***.println(num + **"是"** + *count*(num) + **"位数字"**);  
 System.***out***.println(**"\n"** + **"------------------------------"**);  
}  
  
*/\*  
 1.键盘录入一个整数(正数或者负数都可以,但是符号位不算有效的数字位)  
 2.定义一个方法,该方法的功能是计算该数字是几位数字,并将位数返回  
 3.在main方法中打印该数字是几位数  
\*/***public static int** count(**int** a) {  
 **if** (a < 0) {  
 a \*= -1;  
 }  
 **int** count = 0;  
 **while** (a != 0) {  
 a = a / 10;  
 count++;  
 }  
 **return** count;  
}

### 操作步骤描述

略

## 训练案例6

### 训练描述

分析以下需求，并用代码实现(每个小需求都需要封装成方法)

1.求两个数据之和(整数和小数)

2.判断两个数据是否相等(整数和小数)

3.获取两个数中较大的值(整数和小数)

4.获取两个数中较小的值(整数和小数)

5.否能用一个方法实现3和4的两个功能

### 操作步骤描述

1.求两个数据之和(整数和小数)

(1)参数类型：int或者double

(2)返回值类型：int或者double

所以需要定义两个重载的方法，一个为求两个int数据的和，一个为求两个double数据的和

2.判断两个数据是否相等(整数和小数)

(1)参数类型：int或者double

(2)返回值类型：boolean

所以需要定义两个重载的方法，一个为判断两个int数据是否相等，一个为判断两个double数据是否相等

3.获取两个数中较大的值(整数和小数)

(1)参数类型：int或者double

(2)返回值类型：int或者double

所以需要定义两个重载的方法，一个为求两个int数据的最大值，一个为求两个double数据的最大值

4.获取两个数中较小的值(整数和小数)

(1)参数类型：int或者double

(2)返回值类型：int或者double

所以需要定义两个重载的方法，一个为求两个int数据的最校值，一个为求两个double数据的最小值

5.否能用一个方法实现3和4的两个功能

(1)参数类型：int或者double，但必须有个flag标记用于表示到底是求最大值还是最小值(boolean类型)

(2)返回值类型：int或者double

所以需要定义两个重载的方法，一个为求两个int数据的最校值，一个为求两个double数据的最小值

**public static void** main(String[] args) {  
 System.***out***.println(**"10 + 20 = "** + *sum1*(10, 20));  
 System.***out***.println(**"13.14 + 10.23 = "** + *sum2*(+13.14, 10.23));  
 System.***out***.println(**"------------------------------"**);  
  
 System.***out***.println(**"10等于10？ "** + *equal1*(10, 10));  
 System.***out***.println(**"10.12等于9.81? "** + *equal2*(10.12, 9.81));  
 System.***out***.println(**"------------------------------"**);  
  
 System.***out***.println(**"3和90中，较大的是： "** + *max1*(3, 90));  
 System.***out***.println(**"1.23和5.68中，较大的是： "** + *max2*(1.23, 5.68));  
 System.***out***.println(**"------------------------------"**);  
  
 System.***out***.println(**"3和90中，较小的是： "** + *min1*(3, 90));  
 System.***out***.println(**"1.23和5.68中，较小的是： "** + *min2*(1.23, 5.68));  
 System.***out***.println(**"------------------------------"**);  
  
 System.***out***.println(*mostValue1*(3, 4, **true**));  
 System.***out***.println(*mostValue1*(3, 4, **false**));  
 System.***out***.println(*mostValue2*(1.23, 3.54, **true**));  
 System.***out***.println(*mostValue2*(1.23, 3.54, **false**));  
}  
  
*/\*  
 求两个数据之和(整数和小数)  
\*/***public static int** sum1(**int** a, **int** b) {  
 **int** sum = a + b;  
 **return** sum;  
}  
  
**public static double** sum2(**double** a, **double** b) {  
 **double** sum = a + b;  
 **return** sum;  
}  
  
*/\*  
 判断两个数据是否相等(整数和小数)  
 \*/***public static boolean** equal1(**int** a, **int** b) {  
 **boolean** result = a == b;  
 **return** result;  
}  
  
**public static boolean** equal2(**double** a, **double** b) {  
 **boolean** result = a == b;  
 **return** result;  
}  
  
*/\*  
 获取两个数中较大的值(整数和小数)  
 \*/***public static int** max1(**int** a, **int** b) {  
 **int** result = a > b ? a : b;  
 **return** result;  
}  
  
**public static double** max2(**double** a, **double** b) {  
 **double** result = a > b ? a : b;  
 **return** result;  
}  
  
*/\*  
 获取两个数中较小的值(整数和小数)  
 \*/***public static int** min1(**int** a, **int** b) {  
 **int** result = a < b ? a : b;  
 **return** result;  
}  
  
**public static double** min2(**double** a, **double** b) {  
 **double** result = a < b ? a : b;  
 **return** result;  
}  
  
*/\*  
 用一个方法实现3和4的两个功能  
 \*/***public static int** mostValue1(**int** a, **int** b, **boolean** flag) {  
 **int** result = 0;  
 **if** (flag) {  
 result = a < b ? a : b;  
 } **else** result = a > b ? a : b;  
 **return** result;  
}  
  
**public static double** mostValue2(**double** a, **double** b, **boolean** flag) {  
 **double** result = 0;  
 **if** (flag) {  
 result = a < b ? a : b;  
 } **else** result = a > b ? a : b;  
 **return** result;  
}