**每日作业卷**

**java基础第6天基础强化**

# 基础案例

## 训练案例1

### 训练描述

随机产生5个范围是[1,100]的整数,把这些数字中所有个位或十位为3的数字打印出来

### 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5个范围是[1,100]的随机数把这五个数添加到数组中
3. 遍历数组,获取元素,判断一旦满足个位或者十位为3就打印这个元素

**import** java.util.Random;

**public** **class** Test01 {

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

// **TODO** Auto-generated method stub

Random random = **new** Random();

**int**[] arr = **new** **int**[5];

**for**(**int** i = 0;i<arr.length;i++){

arr[i] = random.nextInt(100)+1;

}

**for**(**int** i : arr){

System.***out***.print(i+" ");

}

System.***out***.println();

*demo*(arr);

}

**private** **static** **void** demo(**int**[] arr) {

**for**(**int** i : arr){

**if**((i)%10 == 3 ||(i/10)%10 == 3 ){

System.***out***.println(i);

}

}

}

}

28 51 37 59 47

37

## 训练案例2

### 训练描述

计算出5位数的回文数和7位数的回文数相差了多少个

### 操作步骤描述

1. 创建一个测试类,
2. 定义两个方法分别统计出5位数的回文数个数和7位数的回文数个数
3. 在主方法中分别调用者两个方法 计算出相差的个数

**public** **class** Test02 {

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

// **TODO** Auto-generated method stub

**int** a = *fif*();

**int** b = *sev*();

System.***out***.println((b-a));

}

**private** **static** **int** fif(){

**int** a = 0;

**for**(**int** i =10000;i<=99999;i++){

**if**(i%10 == i/10000%10 && i/10%10 == i/1000%10){

a++;

}

}

System.***out***.println(a);

**return** a;

}

**private** **static** **int** sev(){

**int** b = 0;

**for**(**int** i =1000000;i<=9999999;i++){

**if**(i%10 == i/1000000%10 && i/10%10 == i/100000%10 && i/100%10 == i/10000%10){

b++;

}

}

System.***out***.println(b);

**return** b;

}

}

900

9000

8100

## 训练案例3

### 训练描述

定义一个方法,求出给定的数字在给定int型数组中出现的次数,如果一次没有出现则返回0

如:给定数字3 求出3在数组 int[] arr = {3,4,3,5,7,9};中出现的次数

### 操作步骤描述

1.创建一个测试类

2.定义个方法可以接受一个int数字key和int型数组arr

3.在方法的内部统计key在arr中出现的次数(即统计arr中有多少个元素与key相等)

4.在主方法中调用这个方法传入一个数字,一个数组,得到数组的返回结果并打印

**import** java.util.Scanner;

**public** **class** Test03 {

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

// **TODO** Auto-generated method stub

**int** key = 1;

**int** arr[] = {0,5,1,6,1,3,1,4,5,6,3};

*demo*(key, arr);

}

**private** **static** **void** demo(**int** key,**int**[] arr) {

**int** count = 0;

**for** (**int** i = 0; i<arr.length;i++) {

**if**(key == arr[i]){

count++;

}

}

System.***out***.println("共出现"+count+"次");

}

}

共出现3次

## 训练案例4

### 训练描述

定义一个方法，查找指定数字在数组中出现的位置(若出现多次，多次打印)

如: 数组[1232]要查找的数是2 则方法内部会打印索引值 1 ,3

数组[1232] 要查找的数是5 则方法每部会打印 “数组中没有这个数字”

### 操作步骤描述

1. 创建一个测试类,在测试类中书写上述代码
2. 定义方法可以接受一个int[] brr和一个int key
3. 在方法中定义标记 int count = 0 ;
4. 方法中遍历brr 对每个元素进行判断是否等于key,如果等于就打印索引值,并count++
5. 遍历brr完毕之后 判断count的值,如果还为默认值0,则表示brr中没有key,直接打印”数组中没有这个数字”
6. 在主方法中传入数组 arr 和数字 2 或者5 进行测试

**import** java.util.Scanner;

**public** **class** Test03 {

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

// **TODO** Auto-generated method stub

Scanner scanner = **new** Scanner(System.***in***);

**int** key = scanner.nextInt();

**int** arr[] = {0,5,1,6,1,3,1,4,5,6,3};

*demo*(key, arr);

}

**private** **static** **void** demo(**int** key,**int**[] arr) {

**int** count = 0;

**for** (**int** i = 0; i<arr.length;i++) {

**if**(key == arr[i]){

count++;

System.***err***.println("arr["+i+"]:"+arr[i]);

}

}

System.***out***.println("共出现"+count+"次");

}

}

1

共出现3次

arr[2]:1

arr[4]:1

arr[6]:1

## 训练案例5

### 训练描述

定义一个方法,实现同时求出两个整数的加、减、乘、除的结果,并同时把这个四个结果返回(把四个数放入到一个数组中并返回)

### 操作步骤描述

参见讲义扩展内容 数组作为方法的返回值

**import** java.util.Scanner;

**public** **class** Test05 {

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

// **TODO** Auto-generated method stub

Scanner scanner = **new** Scanner(System.***in***);

**int** a = scanner.nextInt();

**int** b = scanner.nextInt();

scanner.close();

**for** (**int** i : *demo*(a, b)) {

System.***out***.println(i);

}

}

**private** **static** **int**[] demo(**int** a, **int** b) {

**int**[] arr = **new** **int**[4];

arr[0] = a+b;

arr[1] = a-b;

arr[2] = a\*b;

arr[3] = a/b;

**return** arr;

}

}

4

6

10

-2

24

0

# 扩展案例

## 训练案例1

### 训练描述

1.键盘录入10个整数存入数组中

2.定义一个方法将奇数放在数组的左侧,偶数放在数组的右侧

3.定义一个方法打印原数组和处理后的数组

### 操作步骤描述

**import** java.security.Principal;

**import** java.util.Scanner;

**public** **class** TTest01 {

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

Scanner scanner = **new** Scanner(System.***in***);

**int**[] arr = **new** **int**[10];

**for**(**int** i = 0;i<10;i++){

arr[i] = scanner.nextInt();

}

scanner.close();

*prin*(arr);

System.***out***.println();

*prin*(*demo*(arr));

}

**private** **static** **int**[] demo(**int**[] arr) {

**int**[] brr = **new** **int**[10];

**int** j = 0;

**int** k = 0;

**for**(**int** i = 0; i< arr.length;i++){

**if**(arr[i]%2 != 0){

brr[j++]=arr[i];

}**else** {

brr[brr.length-1-(k++)]=arr[i];

}

}

**return** brr;

}

**private** **static** **void** prin(**int**[] arr) {

**for** (**int** i : arr) {

System.***out***.print(i + " ");

}

}

}

10

12

11

9

7

5

6

2

1

0

10 12 11 9 7 5 6 2 1 0

11 9 7 5 1 0 2 6 12 10

## 训练案例2

### 训练描述

定义一个方法传入一个int类型数组，输出这个数组中每一个数字及其出现的个数

例如 传入数组[1,2,2,2,3,3,4,4,4,4] 打印结果：

数字1出现了1次

数字2出现了3次…

### 操作步骤描述

**import** java.util.HashMap;

**import** java.util.Map;

**import** java.util.Map.Entry;

**public** **class** TTest02 {

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

**int** [] arr = {1,2,2,2,3,3,4,4,4,4};

*testaaa*(arr);

}

**public** **static** **void** testaaa(**int** [] str) {

Map<Integer, Integer> map=**new** HashMap<>();

**for** (**int** i = 0; i < str.length; i++) {

**if**(map.containsKey(str[i])) {

map.put(str[i], map.get(str[i])+1);//原来有则+1

}**else** {

map.put(str[i], 1); //原来没有则放一个

}

}

**for** (Entry<Integer, Integer> entry : map.entrySet()) {

System.***out***.println("数字"+entry.getKey()+"出现了"+entry.getValue()+"次");

}

}

}

数字1出现了1次

数字2出现了3次

数字3出现了2次

数字4出现了4次

## }训练案例3

### 训练描述

分析以下需求，并用代码实现

1.键盘录入6个int类型的数据存数数组arr中

2.将arr数组中的内容反转

3.将翻转后的数组角标为奇数的互相交换 1和3换, 3和5换,以此类推

4.最后将数组最后一个角标为奇数的元素 和数组中第一个角标为奇数的元素交换

5.打印最终的数组(实现了1-4步之后的数组)

6.如：用户输入的6个int数字为[1,2,3,4,5,6],最后输出的结果为[6, 5, 4, 1, 2, 3]

### 操作步骤描述

**import** java.util.Scanner;

**public** **class** TTest03 {

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

// **TODO** Auto-generated method stub

**int**[] arr = {1,2,3,4,5,6,7,8};

*demo*(arr);

**for** (**int** i : arr) {

System.***out***.print(i+"");

}

}

**private** **static** **void** demo(**int**[] arr) {

**for**(**int** startIndex=0,endIndex=arr.length-1;startIndex<=endIndex;startIndex++,endIndex--) {

**int** temp = arr[startIndex];

arr[startIndex] = arr[endIndex];

arr[endIndex] = temp;

}

**if**(arr.length%2 == 0){

**for**(**int** i = 1; i< arr.length-1; i=i+2){

**int** temp = arr[i];

arr[i] = arr[i+2];

arr[i+2] = temp;

}

**int** d = 0;

d = arr[arr.length-1];

arr[arr.length-1] = arr[1];

arr[1] = d;

}

**else**{

**for**(**int** i = 1; i< arr.length-2; i=i+2){

**int** temp = arr[i];

arr[i] = arr[i+2];

arr[i+2] = temp;

}

**int** d = 0;

d = arr[arr.length-2];

arr[arr.length-2] = arr[1];

arr[1] = d;

}

}

}

87634125

## 训练案例4

### 训练描述

分析以下需求，并用代码实现

1.求1-500之内能同时被2，5，7整除的所有数及个数并5个一行打印

2.求100以内能被3整除的数加上能被5整除的数的和

### 操作步骤描述

**public** **class** TTest04 {

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

// **TODO** Auto-generated method stub

*demo*();

*demo2*();

}

**private** **static** **void** demo() {

**int** count = 0;

**for**(**int** i =1;i<=500;i++ ){

**if**(i%2 == 0 && i%5 == 0 && i%7 ==0){

count++;

System.***out***.print(i+" ");

**if**(count%5 == 0){

System.***out***.println();

}

}

}

System.***out***.println();

}

**private** **static** **void** demo2() {

**int** b = 0;

**for**(**int** i = 1;i<100;i++){

**if**(i%3 == 0 || i%5 == 0){

b += i;

}

}

System.***out***.println(b);

}

70 140 210 280 350

420 490

2318

## }训练案例5

### 训练描述

分析以下需求，并用代码实现

有一分数序列：2/1，3/2，5/3，8/5，13/8，21/13... 定义一个方法求出这个数列的前20 项之和。

### 操作步骤描述

**public** **class** TTest05 {

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

// **TODO** Auto-generated method stub

*demo*();

}

**private** **static** **void** demo() {

**int**[] arr = *feibo*(2,3);

**int**[] arr2 = *feibo*(1,2);

**double** sum = 0;

**for**(**int** i = 0; i<20;i++){

sum += (**double**)(1.0\*arr[i]/arr2[i]);

}

System.***out***.println(sum);

}

**private** **static** **int**[] feibo(**int** a,**int** b) {

**int**[] arr = **new** **int**[20];

arr[0] = a;

arr[1] = b;

**for**(**int** i = 2;i<20;i++){

arr[i] = arr[i-1]+arr[i-2];

}

**return** arr;

}

}

32.66026079864164

## 训练案例6??????????????????????????????????????????

### 训练描述

分析以下需求，并用代码实现

1.创建两个长度为10的数组，数组内元素为随机生成的、不重复的 1-100之间的整数， [1,2][1,2]

2.定义一个方法，传入两个数组，方法中将两个数组不同的元素拼接成一个字符串，并且将该字符串以及字符串的长度输出到控制台上；

如果没有则输出"对不起两个数组的所有元素均相同"

**import** java.util.ArrayList;

**import** java.util.ListResourceBundle;

**import** java.util.Random;

**public** **class** TTest06 {

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

// **TODO** Auto-generated method stub

ArrayList<Integer> arr1= *demo*();

**for** (Integer i : arr1) {

System.***out***.print(i+" ");

}

System.***out***.println();

ArrayList<Integer> arr2= *demo*();

**for** (Integer i : arr2) {

System.***out***.print(i+" ");

}

*qqq*(arr1, arr2);

}

**private** **static** ArrayList<Integer> demo() {

ArrayList<Integer> list = **new** ArrayList<>();

Random random = **new** Random();

**while**(list.size()<10){

Integer item = random.nextInt(100)+1;

**if**(!list.contains(item)){

list.add(item);

}

}

**return** list;

}

**private** **static** **void** qqq(ArrayList<Integer> list1,ArrayList<Integer> list2) {

ArrayList<Integer> lista = **new** ArrayList<>();

ArrayList<Integer> listb = **new** ArrayList<>();

**int** count = 0;

**for** (**int** i = 0; i<list1.size();i++) {

**for** (**int** j = 0; j<list2.size();j++) {

**if**(list1.get(i) != list2.get(j)){

}**else** {

listb.add(list1.get(i));

count++;

}

}

}

System.***out***.println();

**if**(count == 0){

**for** (**int** i = 0; i<list1.size();i++) {

lista.add(list1.get(i));

}

**for** (**int** i = 0; i<list1.size();i++) {

lista.add(list2.get(i));

}

System.***out***.println("相同的数有0组");

**for** (Integer integer : lista) {

System.***out***.print(integer+" ");

}

}

**else** {

ArrayList<Integer> demo = **new** ArrayList<>();

**for** (**int** i = 0; i<list1.size();i++) {

demo.add(list1.get(i));

}

**for** (**int** i = 0; i<list1.size();i++) {

demo.add(list2.get(i));

}

**for** (**int** i = 0; i<demo.size();i++) {

**for** (**int** j = 0; j<listb.size();j++) {

**if**(demo.get(i) != listb.get(j)){

lista.add(demo.get(i));

**break**;

}

}

}

**if**(count == 10){System.***out***.println("数组全部相同");}

**else** {

System.***out***.println("相同数有"+count+"组");

}

System.***out***.print("相同的数是：");

**for** (Integer integer : listb) {

System.***out***.print(integer+" ");

}

System.***out***.println();

**for** (Integer integer : lista) {

System.***out***.print(integer+" ");

}

}

}

}

43 77 72 8 30 6 89 44 2 80

67 15 33 76 93 60 45 87 6 34

相同数有1组

相同的数是：6

43 77 72 8 30 89 44 2 80 67 15 33 76 93 60 45 87 34

55 75 87 28 13 35 31 16 51 74

72 28 51 95 92 100 24 6 62 53

相同数有2组

相同的数是：28 51

55 75 87 28 13 35 31 16 51 74 72 28 51 95 92 100 24 6 62 53

### 操作步骤描述

## 训练案例7

### 训练描述

分析以下需求，并用代码实现

1.提示用户输入10个整数，存入到int数组中；根据以下要求实现相关功能

2.提示用户输入一个数字作为数组索引，查找数组中该索引对应的元素，

（需要判断输入的数字是否在0-9之间包含临界值，如果该数字不符合要求则随机生成一个符合要求的索引）

3.将a中得到的索引对应的元素的左右两边元素进行互换，若对应的元素是第一个或最后一个则不做任何操作

### 操作步骤描述

**import** java.util.Random;

**import** java.util.Scanner;

**public** **class** TTest07 {

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

**int**[] arr = **new** **int**[10];

**for** (**int** i = 0; i < arr.length; i++) {

System.***out***.println("请输入第"+(i+1)+"个数：");

Scanner scanner = **new** Scanner(System.***in***);

arr[i] = scanner.nextInt();

}

System.***out***.println("请输入一个数字：");

Scanner scanner = **new** Scanner(System.***in***);

**int** a = scanner.nextInt();

**int** y;

**int** num;

**if**(a>=0&&a<=9){

num = arr[a];

y=a;

}**else**{

Random random = **new** Random();

**int** s = random.nextInt(10);

num = arr[s];

y=s;

}

**if**(y!=0||y!=9){

**int** x = arr[y-1];

arr[y-1] = arr[y+1];

arr[y+1] = x;

}

**for** (**int** i : arr) {

System.***out***.print(i+" ");

}

}

}

请输入第1个数：

1

请输入第2个数：

2

请输入第3个数：

3

请输入第4个数：

4

请输入第5个数：

5

请输入第6个数：

6

请输入第7个数：

7

请输入第8个数：

8

请输入第9个数：

9

请输入第10个数：

10

请输入一个数字：

5

1 2 3 4 7 6 5 8 9 10