课程目标

1. 理解数组
2. 使用数组
3. 数组的应用

插入算法实现

1. 数组(Array)
2. 什么是数组

数组是一个变量，用来存储一组相同数据类型的数据

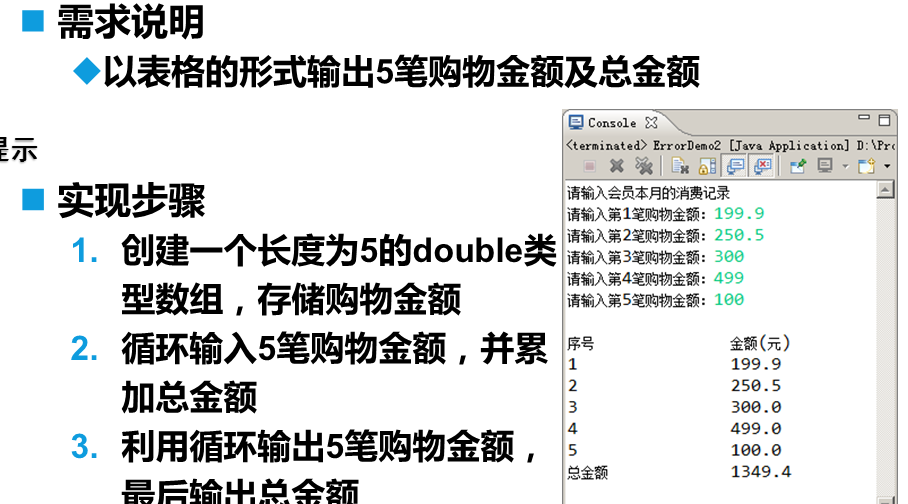
1. 使用数组

|  |
| --- |
| *//1.声明数组 //数据类型 [] 数组名称;* **int** [] maths; *//2.给数组分配内存空间 //数组名称=new 数据类型[数组长度];* maths=**new int**[4]; *//3.数组赋值 注意:索引位置从0开始  //数组名[索引位置]=值;* maths[0]=5;  maths[1]=10;  maths[2]=15;  maths[3]=20; *//4.获取数组中的值(元素)  //数组名[索引位置];* System.***out***.println(maths[0]);  *//5.使用循环控制数组输出* **for** (**int** i = 0; i < maths.length; i++) {  System.***out***.println(maths[i]);  }  //6.获取数组长度  数组名称.length  *//7定义数组并分配内存空间  //数据类型 []数组名=new 数据类型[长度];  //int [] maths=new int[4];*    *//8.定义数组并初始化  //数据类型 []数组名称=new 数据类型[]{元素1,元素2...}; //数据类型 []数组名称={元素1,元素2...}; //int [] maths=new int[]{5,10,15,20};* **int** [] maths={5,10,15,20}; //必需写成一行 |

注意:数组下标越界异常(不能超出数组长度)

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5

3.综合示例:



代码:

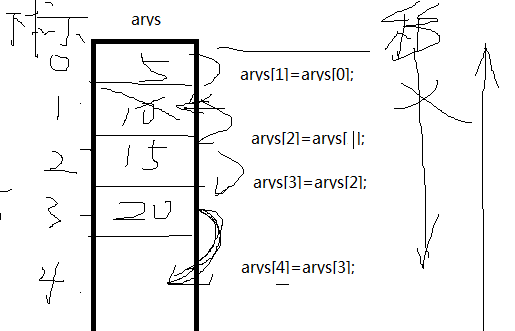
|  |
| --- |
| **public static void** main(String[] args) {  *//示例:循环输入5笔购物金额，进行显示* Scanner input=**new** Scanner(System.***in***);  *//1.定义数组 用于存购物金额* **int** [] moneys=**new int**[5];  *//2.循环输入内容存入数组* **for** (**int** i = 0; i < moneys.**length** ; i++) {  System.***out***.println(**"请输入第"**+(i+1)+**"笔购物金额"**);  **int** money=input.nextInt();  *//将每笔购物金额存入数组* moneys[i]=money;  }  *//显示数组* **int** sum=0; *//累加器* System.***out***.println(**"序号\t\t金额"**);  **for** (**int** i = 0; i < moneys.**length**; i++) {  System.***out***.println((i+1)+**"\t\t"**+moneys[i]);  sum=sum+moneys[i];  }  System.***out***.println(**"总金额:"**+sum); } |

1. 数组应用
2. *排序:使用Arrays的sort方法进行升序排列  
   //1.1导入Arrays类  
   //import java.util.Arrays;  
   //1.2使用sort方法进行升序排序*Arrays.*sort*(数组名称);

了解Arrays的其它方法:

|  |
| --- |
| **public static void** main(String[] args) {  **int** []ary={1,2};  **int** [] arys={1,2,1};  System.***out***.println(**"判断两数组是否相等:"**+Arrays.*equals*(ary,arys));  System.***out***.println(**"将数组转化为字符串"**+Arrays.*toString*(ary));  System.***out***.println(**"查找某个元素在数组中出现的位置，找不到则返回-3:"**+Arrays.*binarySearch*(ary,8));  ]  } }  (注意:使用*binarySearch的*前提条件数组要排序) |

1. 数组插入的算法实现



|  |
| --- |
| *//插入元素* **int** pos=2; *//指定插入的位置 //将插入位置的数依次往后移动位置* **for** (**int** i = 数组名.**length**-1; i >插入的位置; i--) {  temp[i]=temp[i-1]; } |

示例:

|  |
| --- |
| **public static void** main(String[] args) {  *//插入的算法* **int** [] temp=**new int**[10];  temp[0]=5;  temp[1]=10;  temp[2]=15;  temp[3]=20;   *//插入元素* **int** pos=2; *//指定插入的位置  //将插入位置的数依次往后移动位置* **for** (**int** i = temp.**length**-1; i >pos; i--) {  temp[i]=temp[i-1];  }  *//在插入的位置赋* temp[pos]=100;    System.***out***.println(**"插入元素后的数组内容:"**);  **for** (**int** i = 0; i <temp.**length**; i++) {  System.***out***.println(temp[i]);  }  } |

程序=数据结构+算法

1. 数据类型转换

|  |
| --- |
| 1. 数据类型由低到高的顺序  char->short->int->long->float->double 2. 数据类型转换的分类: 隐藏数据类型转换:数据类型由低向高自动转换称为隐藏转换 强制数据类型转换:数据类型由高到低转换称为强制转换 (目标类型)变量   *示例:* **char** c=**'A'**; **int** math=c; *//发生隐藏转换* System.***out***.println(math);  **int** math2=97; **char** c1=(**char**)math2; *//强制转换* System.***out***.println(c1); |