5.定义一个学生类Student，包含三个属性姓名、年龄、性别，创建三个学生对象存入ArrayList集合中。

A：遍历集合遍历输出。

B：求出年龄最大的学生，让后将该对象的姓名变为：葫芦娃。

6.定义一个手机类Phone，包含三个属性品牌(brand)、价格(price)、颜色(color)。

创建四个手机("小米"、"华为"、"乐视"、"奇酷")对象存入ArrayList集合中，并遍历输出。

7.定义一个员工类Employee，包含三个属性姓名、工号、工资，创建三个员工对象存入ArrayList集合中，并遍历输出。

8.利用上面的三个案例，分别演示ArrayList集合中的以下常用方法：

\* public boolean add(元素类型 e) 直接向集合添加元素

\* public void add(int index,元素类型 e) 在指定索引处向集合添加元素

\* public 元素类型 remove(int index) 删除指定索引位置的元素

\* public 元素类型 set(int index,元素类型 e) 修改指定索引位置的元素

\* public int size() 获取集合长度

\* public 元素类型 get(int index) 根据指定索引获取集合元素

三个题目的方法一致

public class Student {

String name;

String grender;

int age;

public String toString() {

return "Student [name=" + name + ", age=" + age + ",grender="+grender+"]";

}

}

import java.util.ArrayList;

public class test {

public static void main(String[] args) {

ArrayList<Student> arr = new ArrayList<Student>();

addStudent(arr);

readMaxAge(arr);

}

public static void addStudent(ArrayList<Student> arr){

Student p1 = new Student();

p1.name="张三";

p1.age=18;

p1.grender="男";

Student p2 = new Student();

p2.name="李四";

p2.age=19;

p2.grender="女";

Student p3 = new Student();

p3.name="王二";

p3.age=20;

p3.grender="男";

arr.add(p1);

arr.add(p2);

arr.add(p3);

}

public static void readMaxAge(ArrayList<Student> arr){

Student p = arr.get(0);

int max = 0;

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

if(p.age<arr.get(i).age){//遍历集合比较找到最大值的地址

max = i;//取出最大值地址的索引 。

}

arr.get(max).name = "葫芦娃"; //.name 访问堆中赋给的值 “葫芦娃”

}

System.out.println("年龄最大的为"+arr.get(max).age); //访问堆中的age值

System.out.println("名字更改后"+arr.get(max).name);//访问堆中的name值

}

}

1.定义一个集合，初始化，然后完成如下功能：

（1）定义一个方法swap()传入集合和两个角标使集合中元素交换位置

（2）把集合中的元素反转

/\*

思路：

a,定义循环, 循环size()/2次

b,第一次交换第一个和最后一个, 第二次交换第二个和倒数第二个

\*/

(3)定义一个replaceAll方法,将传入的新值替换集合中的老值(list,old,new)

import java.util.ArrayList;

public class HomeWork01 {

/\*1.定义一个集合，初始化，然后完成如下功能：

（1）定义一个方法swap()传入集合和两个角标使集合中元素交换位置

（2）把集合中的元素反转

思路：

a,定义循环, 循环size()/2次

b,第一次交换第一个和最后一个, 第二次交换第二个和倒数第二个

(3)定义一个replaceAll方法,将传入的新值替换集合中的老值(list,old,new)\*/

public static void main(String[] args) {

/\*\*

int temp = arr[0];

arr[0] = arr[1];

arr[1] = temp;

array : 1,2,3,4,5,6,7

int temp = list.get(0); 1

list.get(0) = list.get(1); 1 = 2

list.get(1) = temp; 2 = 1

list.set(index,值) --> 改变集合中对应索引上边的值

\*/

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

list.add(1);

list.add(2);

list.add(3);

list.add(4);

list.add(5);

list.add(5);

list.add(6);

list.add(7);

//swapElement(list,1,3);

//reverserElement(list);

replaceAll(list,5,7); // 1 2 3 4 7 7 6 7

}

/\*\*

方法名：swap

返回值：打印： void

参数：一个集合， 两个需要交换位置的角标

\*/

public static void swapElement(ArrayList<Integer> list,int i , int j ){ // 0 1

// 0 1

// 1 2 3 4 5 6 7 8 2 1 3 4 ......

int temp = list.get(i); // 将0索引上的值缓存起来

list.set(i, list.get(j));// 将0索引上的值改变为1 索引

// 将1索引上的值改变成0 索引的值，因为在41 行，0 索引上的值已经被覆盖，但是！我已经把0索引上的值用temp缓存起来；

list.set(j, temp);

System.out.println(list);

}

/\*\*

方法名 ： 反转：reverserElement

返回值：打印，void

参数：集合

思路：

a,定义循环, 循环size()/2次

b,第一次交换第一个和最后一个, 第二次交换第二个和倒数第二个

\*/

public static void reverserElement(ArrayList<Integer> list){

// 1 2 3 4 5 6 7

for(int x = 0 ; x< list.size()/2 ;x++){

/\*\*

\* int temp = arr[i];

\* arr[i] = arr[arr.length -1 - i];

\* arr[arr.length -1 - i ] = temp ;

\* \*/

int temp = list.get(x);

list.set(x, list.get(list.size() - 1 - x));

list.set(list.size() - 1 - x, temp);

}

System.out.println(list);

}

/\*(3)定义一个replaceAll方法,将传入的新值替换集合中的老值(list,old,new)

list, 2 4

list : 1 1 2 2 3 3 ---> 1 1 4 4 3 3

\*/

/\*

方法名replaceAll

参数： list，old，new

返回值：打印，void

\*\*/

public static void replaceAll(ArrayList<Integer> list,int oldNum,int newNum){

// 第一步： 遍历

// 第二步： 拿着old 值到集合中去询问： 你等于不等于 old值？

// 第三步： 如果你是等于，我就把你换成new 替换值！ list.set

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

if( oldNum == list.get(i)){

list.set(i, newNum);

}

}

System.out.println(list);

}

}

2.把集合中存储多个Person对象,Person有姓名和年龄,找出年龄最大的

/\*思路：

a,定义一个Person类型的变量, 先记住第一个元素

b,循环遍历集合

c,用每一个元素和变量比较年龄, 如果集合中的元素比变量记住的年龄大

d,用变量记住这个年龄较大的元素

e,返回变量记住的元素

\*/

public class Person {

String name;

int age;

@Override

public String toString() {

return "Person [name=" + name + ", age=" + age + "]";

}

}

public class HomeWork02 {

/\*把集合中存储多个Person对象,Person有姓名和年龄,找出年龄最大的

思路：

a,定义一个Person类型的变量, 先记住第一个元素

b,循环遍历集合

c,用每一个元素和变量比较年龄, 如果集合中的元素比变量记住的年龄大

d,用变量记住这个年龄较大的元素

e,返回变量记住的元素

\*/

public static void main(String[] args) {

ArrayList<Person> list = new ArrayList<Person>();

addStudent(list);//帮我把数据准备好

// 如何把集合中有最大年龄的这哥们挑选出来

readMaxAgeForList(list);

}

public static void readMaxAgeForList(ArrayList<Person> list){

/\*

\* 先将数组中第一号元素取出来 int max = arr[0]

\* 拿着这个max 到数组中每一位去询问有没有哥们比我大呀

\* 如果有！说明max不是最大的

\* 将比我大的哥们赋值给max；

\* 如果没有！

\* \*\*/

Person p =list.get(0);

for(int x = 1 ; x < list.size(); x++){

if(p.age < list.get(x).age ){

//如果满足，说明集合中有人比我大

p.age = list.get(x).age; // list.get(1) --->杨幂 20

p.name = list.get(x).name;

}

}

/\* System.out.println(p.age);

System.out.println(p.name);

\*/

System.out.println(list);

}

public static void addStudent(ArrayList<Person> list){

/\* list.add(p1);

list.add(p2);

list.add(p3);\*/

Person p1 = new Person();

p1.name = "肖驰";

p1.age =17;

Person p2 = new Person();

p2.name = "杨幂";

p2.age = 20;

Person p3 = new Person();

p3.name = "马云";

p3.age = 10;

list.add(p1);

list.add(p2);

list.add(p3);

}

}

3.从键盘中录入3个人的信息（姓名、年龄、性别、电话、住址），并封装成对象，最后把对象存储在集合中。

并做简单的代码实现

[要求：使用两个方式 集合转为数组、迭代器]

import java.util.ArrayList;

import java.util.Scanner;

public class HomeWork03 {

/\*

\* 迭代器在迭代过程之中，不仅能用集合去修改元素

\* 还能用集合去删除元素

\* 限制： 能删除最后一位和倒数第二位

\*\*/

public static void main(String[] args) {

/\* 从键盘中录入3个人的信息（姓名、年龄、性别、电话、住址），并封装成对象，最后把对象存储在集合中。

并做简单的代码实现\*/

// 首先得有一个Person对象！ new Person Person对象存储到集合中

ArrayList<Person1> list = readListForScanner();

System.out.println(list);

}

public static ArrayList<Person1> readListForScanner(){

ArrayList<Person1> array = new ArrayList<Person1>();

// Person 的信息来源于键盘录入

Scanner sc = new Scanner(System.in);

//反复的录入Person信息

for(int x = 0 ; x< 3;x++){

Person1 p = new Person1();

System.out.println("请录入姓名");

p.name = sc.next();

System.out.println("请录入年龄");

p.age = sc.nextInt();

System.out.println("请录入性别");

p.gender = sc.next();

System.out.println("电话");

p.phone = sc.next();

System.out.println("地址");

p.address = sc.next();

array.add(p);

}

return array;

}

}

class Person1{

//虚拟机在运行时会找到一个方法的入口， main方法,我们默认有public在的地方，就有main方法，

String name;

int age;

String gender;//性别

String phone;

String address;

@Override

public String toString() {

return "Person1 [name=" + name + ", age=" + age + ", gender=" + gender + ", phone=" + phone + ", address="

+ address + "]";

}

}