今日学习目标

* 能够使用数组存储自定义类型并遍历
* 能够使用ArrayList集合的构造方法创建ArrayList集合对象
* 能够使用ArrayList集合存储数据
* 能够使用ArrayList集合中常用的方法
* 能够使用ArrayList集合存储字符串并遍历
* 能够使用ArrayList集合存储自定义对象并遍历
* 能够将数组中的元素添加到集合中，并按要求打印到控制台上
* 能够理解学生管理系统功能

# ArrayList集合

## 对象数组的案例

* 创建一个学生数组，存储三个学生对象并遍历

**public** **class** Student {

**private** String name;

**private** **int** age;

**public** Student() {

}

**public** Student(String name, **int** age) {

**this**.name = name;

**this**.age = age;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

}

**public** **class** Demo01Array {

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

//创建学生数组

Student[] students = **new** Student[3];

//创建学生对象

Student s1 = **new** Student("曹操",40);

Student s2 = **new** Student("刘备",35);

Student s3 = **new** Student("孙权",30);

//把学生对象作为元素赋值给学生数组

students[0] = s1;

students[1] = s2;

students[2] = s3;

//遍历学生数组

**for**(**int** x=0; x<students.length; x++) {

Student s = students[x];

//System.out.println(s);

System.***out***.println(s.getName()+"---"+s.getAge());

}

}

}

* 对象数组的内存图

## 为什么出现集合类

我们学习的是面向对象语言，而面向对象语言对事物的描述是通过对象体现的，为了方便对多个对象进行操作，我们就必须把这多个对象进行存储。

而要想存储多个对象，就不能是一个基本的变量，而应该是一个容器类型的变量，在我们目前所学过的知识里面，有哪些是容器类型的呢?数组和StringBuilder。

但是呢? StringBuilder的结果是一个字符串，不一定满足我们的要求，所以我们只能选择数组，这就是对象数组。

而对象数组又不能适应变化的需求，因为数组的长度是固定的，这个时候，为了适应变化的需求，Java就提供了集合类供我们使用，由此可见集合的长度是可变的。

## ArrayList类的构造方法和成员方法

* ArrayList<E>集合:大小可变数组的实现
  + <E>:是一种特殊的数据类型，泛型。
  + 在出现E的地方我们使用引用数据类型替换即可
  + 举例：ArrayList<String>,ArrayList<Student>
* 构造方法
  + ArrayList()
* 成员方法
  + 添加元素
    - public boolean add(E e):添加元素，在末端追加一个元素
    - public void add(int index,E element):在指定的索引处添加一个元素

**public** **class** Demo01ArrayList {

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

//创建集合对象

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

//add(E e):添加元素

list.add("hello");

list.add("world");

list.add("java");

//add(int index,E element):在指定的索引处添加一个元素

list.add(1, "android");

System.***out***.println("array:"+list);

}

}

* 更多成员方法
  + 获取元素
    - public E get(int index):返回指定索引处的元素
  + 集合长度
    - public int size():返回集合中的元素的个数
  + 删除元素
    - public boolean remove(Object o):删除指定的元素，返回删除是否成功
    - public E remove(int index):删除指定索引处的元素，返回被删除的元素
  + 修改元素
    - public E set(int index,E element):修改指定索引处的元素，返回被修改的元素

**public** **class** Demo02ArrayList {

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

//创建集合对象

ArrayList<String> array = **new** ArrayList<String>();

//添加元素

array.add("hello");

array.add("world");

array.add("java");

//public E get(int index):返回指定索引处的元素

System.***out***.println("get:"+array.get(0));

System.***out***.println("get:"+array.get(1));

System.***out***.println("get:"+array.get(2));

//public int size():返回集合中的元素的个数

System.***out***.println("size:"+array.size());

//public boolean remove(Object o):删除指定的元素，返回删除是否成功

System.***out***.println("remove:"+array.remove("world"));//true

System.***out***.println("remove:"+array.remove("world"));//false

//public E remove(int index):删除指定索引处的元素，返回被删除的元素

System.***out***.println("remove:"+array.remove(0));

//public E set(int index,E element):修改指定索引处的元素，返回被修改的元素

System.***out***.println("set:"+array.set(0, "android"));

//输出

System.***out***.println("array:"+array);

}

}

* 存储基本数据类型
  + byte Byte
  + short Short
  + int **Integer**
  + long Long
  + float Float
  + long Long
  + char **Character**
  + boolean Boolean

## ArrayList类案例

* 练习一

存储字符串并遍历

**public** **class** Demo03ArrayList {

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

//创建集合对象

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

//添加元素

list.add("hello");

list.add("world");

list.add("java");

//获取元素

//原始做法

System.***out***.println(list.get(0));

System.***out***.println(list.get(1));

System.***out***.println(list.get(2));

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

**for**(**int** i=0; i<3; i++) {

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

}

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

//如何知道集合中元素的个数呢?size()

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

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

}

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

//最标准的用法

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

String s = list.get(i);

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

}

}

}

* 练习二

给定一个字符串数组：{"张三丰","宋远桥","张无忌","殷梨亭","张翠山","莫声谷"};

，将数组中的元素添加到集合中，并把所有姓张的人员打印到控制台上。

**public** **class** Demo02ArrayListTest {

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

//定义字符串数组

String[] strArray = {"张三丰","宋远桥","张无忌","殷梨亭","张翠山","莫声谷"};

//创建集合对象

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

//遍历字符串数组，获取到每一个字符串元素

**for**(**int** x=0; x<strArray.length; x++) {

//把获取到的字符串元素添加到集合

list.add(strArray[x]);

}

//遍历集合

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

String s = list.get(x);

//要判断每一个字符串元素是否以"张"开头，如果是，就输出在控制台

**if**(s.startsWith("张")) {

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

}

}

}

}

* 练习三

存储自定义对象并遍历（也可以自定义Person类）

**public** **class** Student {

**private** String name;

**private** **int** age;

**public** Student() {

}

**public** Student(String name, **int** age) {

**this**.name = name;

**this**.age = age;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

}

**public** **class** Demo03ArrayListTest {

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

//创建集合对象

ArrayList<Student> array = **new** ArrayList<Student>();

//创建学生对象

Student s1 = **new** Student("赵丽颖",18);

Student s2 = **new** Student("唐嫣",20);

Student s3 = **new** Student("景甜",25);

Student s4 = **new** Student("柳岩",19);

//把学生对象作为元素添加到集合中

array.add(s1);

array.add(s2);

array.add(s3);

array.add(s4);

//遍历集合

**for**(**int** x=0; x<array.size(); x++) {

Student s = array.get(x);

System.***out***.println(s.getName()+"---"+s.getAge());

}

}

}

* 练习四

创建一个集合，存储学生对象，学生对象的数据来自于键盘录入，最后，遍历集合

**public** **class** Student {

**private** String name;

**private** String age;

**public** Student() {

}

**public** Student(String name, String age) {

**this**.name = name;

**this**.age = age;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getAge() {

**return** age;

}

**public** **void** setAge(String age) {

**this**.age = age;

}

}

**public** **class** Demo04ArrayListTest {

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

//创建集合对象

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

//为了提高代码的复用性，我把键盘录入数据给学生对象，并存储到集合中的动作用一个方法来实现

//调用方法

*addStudent*(list);

*addStudent*(list);

*addStudent*(list);

//遍历集合

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

Student s = list.get(x);

System.***out***.println(s.getName()+"---"+s.getAge());

}

}

/\*

\* 两个明确：

\* 返回值类型：void

\* 参数列表：ArrayList<Student> list

\*/

**public** **static** **void** addStudent(ArrayList<Student> list) {

//键盘录入数据,创建学生对象,把键盘录入的数据赋值给学生对象的成员变量

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

System.***out***.println("请输入学生姓名：");

String name = sc.nextLine();

System.***out***.println("请输入学生年龄：");

String age = sc.nextLine();

Student s = **new** Student();

s.setName(name);

s.setAge(age);

//把学生对象作为元素存储到集合中

list.add(s);

}

}

## 类名作为形式参数和返回值

* 类名作为形式参数案例:需要的其实是该类的对象

class Test {

public void show(Person p){

p.eat();

}

}

* 类名作为返回值案例:返回的其实是该类的对象

class Test {

public Person show(){

Person p = new Person("小王");

return p;

}

}

# 学生管理系统

* 学生管理系统项目演示
* 学生管理系统需求分析
* 学生管理系统之标准学生类

**public** **class** Student {

//学号

**private** String id;

//姓名

**private** String name;

//年龄

**private** String age;

//居住地

**private** String address;

**public** Student() {

}

**public** Student(String id, String name, String age, String address) {

**this**.id = id;

**this**.name = name;

**this**.age = age;

**this**.address = address;

}

**public** String getId() {

**return** id;

}

**public** **void** setId(String id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getAge() {

**return** age;

}

**public** **void** setAge(String age) {

**this**.age = age;

}

**public** String getAddress() {

**return** address;

}

**public** **void** setAddress(String address) {

**this**.address = address;

}

}

* 学生管理系统的主类

**public** **class** StudentManager {

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

//创建集合对象，用于存储学生数据

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

//为了让程序能够回到这里来，我们使用循环

**while**(**true**) {

//这是学生管理系统的主界面

System.***out***.println("--------欢迎来到学生管理系统--------");

System.***out***.println("1 查看所有学生");

System.***out***.println("2 添加学生");

System.***out***.println("3 删除学生");

System.***out***.println("4 修改学生");

System.***out***.println("5 退出");

System.***out***.println("请输入你的选择：");

//创建键盘录入对象

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

String choiceString = sc.nextLine();

//用switch语句实现选择

**switch**(choiceString) {

**case** "1":

//查看所有学生

*findAllStudent*(list);

**break**;

**case** "2":

//添加学生

*addStudent*(list);

**break**;

**case** "3":

//删除学生

*deleteStudent*(list);

**break**;

**case** "4":

//修改学生

*updateStudent*(list);

**break**;

**case** "5":

//退出

//System.out.println("谢谢你的使用");

//break;

**default**:

System.***out***.println("谢谢你的使用");

System.*exit*(0); //JVM退出

**break**;

}

}

}

/\*

\* 修改学生的方法

\* 修改学生的思路：键盘录入一个学号，到集合中去查找，看是否有学生使用的是该学号，如果有就修改该学生

\*/

**public** **static** **void** updateStudent(ArrayList<Student> list) {

//创建键盘录入对象

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

System.***out***.println("请输入你要修改的学生的学号：");

String id = sc.nextLine();

//定义一个索引

**int** index = -1;

//遍历集合

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

//获取每一个学生对象

Student s = list.get(x);

//拿学生对象的学号和键盘录入的学号进行比较

**if**(s.getId().equals(id)) {

index = x;

**break**;

}

}

**if**(index == -1) {

System.***out***.println("不好意思,你要修改的学号对应的学生信息不存在,请回去重新你的选择");

}**else** {

System.***out***.println("请输入学生新姓名：");

String name = sc.nextLine();

System.***out***.println("请输入学生新年龄：");

String age = sc.nextLine();

System.***out***.println("请输入学生新居住地：");

String address = sc.nextLine();

//创建学生对象

Student s = **new** Student();

s.setId(id);

s.setName(name);

s.setAge(age);

s.setAddress(address);

//修改集合中的学生对象

list.set(index, s);

//给出提示

System.***out***.println("修改学生成功");

}

}

/\*

\* 删除学生的方法

\* 删除学生的思路：键盘录入一个学号，到集合中去查找，看是否有学生使用的是该学号，如果有就删除该学生

\*/

**public** **static** **void** deleteStudent(ArrayList<Student> list) {

//创建键盘录入对象

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

System.***out***.println("请输入你要删除的学生的学号：");

String id = sc.nextLine();

//我们必须给出学号不存在的时候的提示

//定义一个索引

**int** index = -1;

//遍历集合

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

//获取到每一个学生对象

Student s = list.get(x);

//拿这个学生对象的学号和键盘录入的学号进行比较

**if**(s.getId().equals(id)) {

index = x;

**break**;

}

}

**if**(index == -1) {

System.***out***.println("不好意思,你要删除的学号对应的学生信息不存在,请回去重新你的选择");

}**else** {

list.remove(index);

System.***out***.println("删除学生成功");

}

}

//添加学生的方法

**public** **static** **void** addStudent(ArrayList<Student> list) {

//创建键盘录入对象

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

//为了让id能够被访问到，我们就把id定义在了循环的外面

String id;

//为了让代码能够回到这里，用循环

**while**(**true**) {

System.***out***.println("请输入学生学号：");

//String id = sc.nextLine();

id = sc.nextLine();

//判断学号有没有被人占用

//定义标记

**boolean** flag = **false**;

//遍历集合，得到每一个学生

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

Student s = list.get(x);

//获取该学生的学号，和键盘录入的学号进行比较

**if**(s.getId().equals(id)) {

flag = **true**; //说明学号被占用了

**break**;

}

}

**if**(flag) {

System.***out***.println("你输入的学号已经被占用,请重新输入");

}**else** {

**break**; //结束循环

}

}

System.***out***.println("请输入学生姓名：");

String name = sc.nextLine();

System.***out***.println("请输入学生年龄：");

String age = sc.nextLine();

System.***out***.println("请输入学生居住地：");

String address = sc.nextLine();

//创建学生对象

Student s = **new** Student();

s.setId(id);

s.setName(name);

s.setAge(age);

s.setAddress(address);

//把学生对象作为元素添加到集合

list.add(s);

//给出提示

System.***out***.println("添加学生成功");

}

//查看所有学生

**public** **static** **void** findAllStudent(ArrayList<Student> list) {

//首先来判断集合中是否有数据，如果没有数据，就给出提示，并让该方法不继续往下执行

**if**(list.size() == 0) {

System.***out***.println("不好意思,目前没有学生信息可供查询,请回去重新选择你的操作");

**return**;

}

//\t 其实就是一个tab键的位置

System.***out***.println("学号\t\t姓名\t年龄\t居住地");

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

Student s = list.get(x);

System.***out***.println(s.getId()+"\t"+s.getName()+"\t"+s.getAge()+"\t"+s.getAddress());

}

}

}