（Collection、迭代器、泛型）

1. **知识点：Collection接口**

**要求：**

1. **掌握Collection接口中的常用方法：**

题目：

1. 请写出Collection接口的两个子接口？
2. 请写出几个Collection接口的子类的名字。
3. 请说明List接口有什么特点；
4. 请说明Set接口有什么特点；
5. 请写出Collection接口中与下述操作相关的方法：
6. 添加元素：
7. 删除元素：
8. 判断元素：
9. 获取集合大小；
10. 获取元素：
11. 定义类，定义main()方法，按以下步骤编写代码：
12. 使用Collection多态，实例化一个ArrayList对象；里面存储String类型；
13. 向集合中添加以下数据：

章子怡

范冰冰

柳岩

李冰冰

杨颖

1. 判断“柳岩”是否存在
2. 删除“柳岩”；
3. 打印当前集合大小；
4. 获取Object[]数组，遍历并打印元素；
5. 获取迭代器(需要指定泛型)，遍历并打印元素；
6. 按以下步骤编写代码：
7. 定义类：Cat，包含以下成员：

成员属性(私有)：

名称：

颜色：

年龄：

构造方法：

无参

全参

成员方法：

1).get/set方法；

2).重写toString()方法；内部打印所有属性的值；

b) 在main()方法中，按以下步骤编写代码：

* 向集合中添加以下cat对象：

“波斯猫”，“白色”，2

“折耳猫”，“灰色”，1

“加菲猫”，“红色”，3

“机器猫”，“蓝色”，5

* 分别使用数组，迭代器(需要指定泛型)，增强for三种方式遍历集合；

编写一个程序的模拟班级学生的成绩库, 包含多项成绩,英语,数学,Java.

实现如下功能：

1.可以修改某个学生的某项分数.

2.可以打印全班同学成绩

要求使用集合来完成.

|  |
| --- |
| **public** **class** Test2 {  **public** **static** **void** main(String[] args) {  ArrayList<Student> student\_list = **new** ArrayList<>();  student\_list.add(**new** Student("张三", 99, 90, 100));  student\_list.add(**new** Student("李四", 89, 80, 100));  **for** (Student student : student\_list) {  System.***out***.println(student);  }  Scanner scanner = **new** Scanner(System.***in***);  // 请输入学生姓名  System.***out***.println("请输入学生姓名 ");  String name = scanner.nextLine();  **for** (Student student : student\_list) {  String n = student.getName();  **if** (n.equals(name)) {  // 可以继续录入, 修改  System.***out***.println("请输入科目 ");  String kemu = scanner.nextLine();  System.***out***.println("请输入分数 ");  **double** score = scanner.nextDouble();  // 根据不同的科目,修改不同的分数.  **switch** (kemu) {  **case** "数学":  student.setMath(score);  **break**;  **case** "英语":  student.setEnglish(score);  **break**;  **case** "java":  student.setJava(score);  **break**;  **default**:  System.***out***.println("查无此课, 对不起");  **return** ;  }  System.***out***.println("恭喜您, 修改成功");  System.***out***.println(student);  **return**;  }  }  System.***out***.println("对不起 ,查无此人");  }  }  **class** Student {  **private** **double** english;  **private** **double** math;  **private** **double** java;  **private** String name;  **public** Student() {  **super**();  // **TODO** Auto-generated constructor stub  }  **public** Student(String name, **double** english, **double** math, **double** java) {  **super**();  **this**.english = english;  **this**.math = math;  **this**.java = java;  **this**.name = name;  }  **public** **double** getEnglish() {  **return** english;  }  **public** **void** setEnglish(**double** english) {  **this**.english = english;  }  **public** **double** getMath() {  **return** math;  }  **public** **void** setMath(**double** math) {  **this**.math = math;  }  **public** **double** getJava() {  **return** java;  }  **public** **void** setJava(**double** java) {  **this**.java = java;  }  **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }  @Override  **public** String toString() {  **return** "Student [english=" + english + ", math=" + math + ", java=" + java + ", name=" + name + "]";  }    @Override  **public** **boolean** equals(Object obj) {  **if** (**this** == obj)  **return** **true**;  **if** (obj == **null**)  **return** **false**;  **if** (getClass() != obj.getClass())  **return** **false**;  Student other = (Student) obj;  **if** (Double.*doubleToLongBits*(english) != Double.*doubleToLongBits*(other.english))  **return** **false**;  **if** (Double.*doubleToLongBits*(java) != Double.*doubleToLongBits*(other.java))  **return** **false**;  **if** (Double.*doubleToLongBits*(math) != Double.*doubleToLongBits*(other.math))  **return** **false**;  **if** (name == **null**) {  **if** (other.name != **null**)  **return** **false**;  } **else** **if** (!name.equals(other.name))  **return** **false**;  **return** **true**;  }  } |