

Day 16 集合框架

Key Point :

- Collection 接口、List 接口基本操作
- List 接口及其实现类
- for-each遍历
- 泛型

问题：

1.填空

Collection 接口的特点是元素是_；
List 接口的特点是元素（有|无）顺序，_（可以|不可以）重复；
Set接口的特点是元素（有|无）顺序，_（可以|不可以）重复；

2.填空

Collection中定义的添加方法声明是_
删除方法的声明__
获取元素个数的方法声明__
List中定义的与下标相关的添加方法声明是____
与下标相关的删除方法声明是____
获取元素的方法声明____
获取指定元素的下标方法声明____

3. (List) 有如下代码

```

import java.util.*;
public class TestList{
    public static void main(String args[]){
        List<String> list = new ArrayList<String>();
        list.add("Hello");
        list.add("World");
        list.add(1,"Learn");
        list.add(1,"Java");
        printList(list);
    }
    public static void printList(List<String> list){
        //1
    }
    public static void printCollection(Collection<String> collection){
        //2
    }
}

```

要求：

- I. 把//1处的代码补充完整，要求使用下标遍历输出list 中所有元素的内容
- 把//2处的代码补充完整，要求使用for-each遍历输出list 中所有元素的内容
- II. 写出程序执行的结果
- III. 如果要把实现类由ArrayList换为LinkedList，应该改哪里？ArrayList和LinkedList使用上有什么区别？实现上有什么区别？
- IV. 如果要把实现类由ArrayList换为Vector，应该改哪里？ArrayList 和Vector使用上有什么区别？实现上有什么区别？

4. (List) 写出下面程序的运行结果

```
import java.util.*;
public class TestList{
    public static void main(String args[]){
        List<String> list = new ArrayList<String>();
        list.add("Hello");
        list.add("World");
        list.add("Hello");
        list.add("Learn");
        list.remove("Hello");
        list.remove(0);
        for(int i = 0; i<list.size(); i++){
            System.out.println(list.get(i));
        }
    }
}
```

5. (List) 已知有一个Worker 类如下：

```

public class Worker {
    private int age;
    private String name;
    private double salary;
    public Worker (){}
    public Worker (String name, int age, double salary){
        this.name = name;
        this.age = age;
        this.salary = salary;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public double getSalary(){
        return salary;
    }
    public void setSalary(double salary){
        this.salary = salary;
    }
    public void work(){
        System.out.println(name + " work");
    }
}

```

完成下面的要求：

I. 创建一个List，在List 中增加三个Worker对象，基本信息如下

姓名	年龄	工资
zhang3	18	3000
li4	25	3500
wang5	22	3200

注意：将每一行信息封装成一个Worker对象，并添加到List中

II. 在li4 之前插入一个Worker对象，信息为：姓名：zhao6，年龄：24，工资3300

III. 删除下标为2的Work对象

IV. 利用for 循环遍历，打印List 中所有工人的信息

V. 利用for-each遍历，对List 中所有的工人调用work 方法。

VI. 为Worker 类添加equals 方法

6、(List)

```
class Animal{ }
class Dog extends Animal{
    public void play(){
        System.out.println("Dog play with you");
    }
}
class Cat extends Animal{ }
public class TestAnimal{
    public static void main(String args[]){
        List<Animal> as = new ArrayList<Animal>();
        as.add(new Dog());
        as.add(new Cat());
        as.add(new Dog());
        as.add(new Cat());

        List<Dog> dogs = getAllDog(as);
        //使用2种方式遍历dogs集合，并调用每一个Dog对象的paly方法

    }

    public static List<Dog> getAllDog(List<Animal> animals){

    }
}
```

程序填空：

完成getAllDog 方法，从一个Animal集合中挑选出所有的Dog对象，并把这些对象放在一个Dog 集合中返回。