Java07-集合与泛型

Task1

Q1. List 接口：有序、可重复。实现类有 ArrayList（动态数组，查询快）、LinkedList（链表，增删快）。 Set 接口：无序、不可重复。实现类有 HashSet（基于哈希表，查找快）、TreeSet（有序，基于红黑树）、LinkedHashSet（有序，基于链表+哈希表）。 Map 接口：键值对，无序，key 不可重复。实现类有 HashMap（常用，查找快）、TreeMap（有序）、LinkedHashMap（有序）。 **第三题已经有了一些集合的学习笔记** 与数组的区别：数组长度固定，集合长度可变。数组只能存同类型元素，集合可存对象且类型灵活。集合有丰富的操作方法（如增删查改、排序等），数组操作较为基础。集合更适合处理动态、复杂的数据结构，数组适合定长、简单数据。

Task2

Q2. 如图所示：

```java
import java.util.ArrayList;
import java.util.List;

public class abc {
    Run | Debug
    public static void main(String[] args) {
        List<Integer> list = new ArrayList<>();
list.add(e:1);
list.add(e:2);
list.add(e:3);
list.add(e:4);
for(Integer i:list){
    System.out.println(i);
    }
}
}
```

问题 83 输出 调试控制台 终端 端口

```
PS C:\Users\29867\Desktop\daima>  & 'D:\java\bin\java.exe' '-
XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\29867
hat.java\jdt_ws\daima_d46075f1\bin' 'abc'
1
2
3
4
```

Q3. 匿名内部类是没有名字的内部类，只用一次。函数式接口是只包含一个抽象方法的接口，可以用 lambda

表达式或方法引用来简化实现。 改写如下图（有两种）：

```java
import java.util.List;

public class abc {
    Run | Debug
    public static void main(String[] args) {
        List<Integer> list = new ArrayList<>();
list.add(e:1);
list.add(e:2);
list.add(e:3);
list.add(e:4);
list.forEach(System.out::println);
/*或者是list.forEach(e -> System.out.println(e)); */
}
}
```

问题 59 输出 调试控制台 终端 端口

```
PS C:\Users\29867\Desktop\daima>  & 'D:\java\bin\java.exe' '-agentlib
lsInExceptionMessages' '-cp' 'C:\Users\29867\AppData\Roaming\Code\User
' 'abc'
1
2
3
4
```

lambda 特点：语法：(参数) -> {方法体}，可省略参数类型和大括号（单语句时）。只能用于函数式接口。代码更简洁，易读。

Task3

Q4. 详情请见 seventhQuestion 文件夹。如图：

question > Ques.java > {} question

```java
package question;
public class Ques {
    Run | Debug
    public static void main(String[] args) {
        // User仓库
        MyRepository<User> repository = new MyRepository<>();
        repository.save(new User(name:"Lubenwei666", age:30));
        repository.save(new User(name:"Lubenwei777", age:31));
        repository.save(new User(name:"Lubenwei888", age:32));
        System.out.println(x:"User仓库: ");
        for(User u : repository.map.values()) {
            System.out.println(u);
        }

        // String仓库
        MyRepository<String> stringRepository = new MyRepository<>();
        stringRepository.save(data:"Lubenwei");
        stringRepository.save(data:"Lubenwei777");
        stringRepository.save(data:"Lubenwei888");
        System.out.println(x:"String仓库: ");
        for(String s : stringRepository.map.values()) {
            System.out.println(s);
        }

        // Integer仓库
        MyRepository<Integer> intRepository = new MyRepository<>();
        intRepository.save(data:100);
        intRepository.save(data:200);
        intRepository.save(data:300);
        System.out.println(x:"Integer仓库: ");
        for(Integer i : intRepository.map.values()) {
            System.out.println(i);
        }
    }
```

问题 82 输出 调试控制台 终端 端口

```
User仓库:
User{name='Lubenwei666', age=30}
User{name='Lubenwei777', age=31}
User{name='Lubenwei888', age=32}
String仓库:
Lubenwei
Lubenwei777
Lubenwei888
Integer仓库:
100
200
300
PS C:\Users\29867\Desktop\daima>
```

Task4

Q5. 如图所示：

```java
import java.util.ArrayList;
import java.util.List;
public class MockSongs {
    public static List<String> getSongStrings(){
        List<String> songs = new ArrayList<>();

        songs.add(e:"sunrise");
        songs.add(e:"thanks");
        songs.add(e:"$100");
        songs.add(e:"havana");
        songs.add(e:"114514");
        // 对歌曲列表排序：
        // 1. 先按字符串长度升序排列
        // 2. 长度相同则按字符类型优先级（字母<数字<其他符号）逐字符比较
        songs.sort((s1,s2)->{
           if (s1.length()!=s2.length()) {
               return s1.length()-s2.length();
           }else{
            for(int i=0;i<s1.length();i++){
                if (s1.charAt(i)!=s2.charAt(i)) {
                    return judgeType(s1.charAt(i))-judgeType(s2.charAt(i));
                }
            }   return 0;
           }

        });
        return songs;
    }
    public static int judgeType(char c){
        if (Character.isLetter(c)) return 1;
        if (Character.isDigit(c)) return 2;
        return 3;
    }
}
```

```
t_ws\daima_d46075f1\bin' 'ss'
[$100, thanks, havana, 114514, sunrise]
```