**每日作业卷**

**JavaSE第08天**

# 关卡1

## 训练案例1

### 训练知识点

1. 栈和队列的特点

### 训练描述

请说出栈和队列的特点

### 操作步骤描述

1.说出栈特点: 先进后出

2.说出队列特点: 先进先出

## 训练案例2

### 训练知识点

1.数组和链表特点

### 训练描述

请说出数组和链表特点

### 操作步骤描述

1.说出数组的特点: 一块连续的存储区域

2.说出链表的特点: 每个元素指向下一个元素

## 训练案例3

### 训练知识点

1.LinkedList基本使用

### 训练描述

练习LinkedList基本方法: add, set, get, remove, clear, size方法

### 操作步骤描述

1. 创建LinkedList
2. 使用add方法添加元素
3. 使用add方法在指定索引添加元素
4. 使用set方法修改指定位置索引
5. 使用get方法获取指定索引的元素
6. 使用size方法获取集合大小
7. 使用remove方法删除指定索引的元素
8. 使用clear清空集合中的元素

**package** com.day08.exam01;

**import** java.util.LinkedList;

**public** **class** Test03 {

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

LinkedList<String> linkedList = **new** LinkedList<>();

linkedList.add("a");

linkedList.add("b");

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

linkedList.add(2, "c");

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

linkedList.set(1, "d");

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

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

System.***out***.println(linkedList.size());

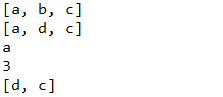
linkedList.remove(0);

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

linkedList.clear();

}

}



## 训练案例4

### 训练知识点

1.LinkedList特有方法

### 训练描述

练习LinkedList特有方法: addFirst, addLast, getFirst, getLast, removeFirst, removeLast方法

### 操作步骤描述

1. 创建LinkedList
2. 使用add方法添加元素
3. 使用addFirst添加元素到集合最前面
4. 使用addLast添加元素到集合最后面
5. 使用getFirst获取集合第一个元素
6. 使用getLast获取集合最后一个元素
7. 使用removeLast删除集合第一个元素
8. 使用removeLast删除集合最后一个元素

**package** com.day08.exam01;

**import** java.util.LinkedList;

**public** **class** Test04 {

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

LinkedList<String> linkedList = **new** LinkedList<>();

linkedList.add("a");

linkedList.add("b");

linkedList.addFirst("1");

linkedList.addLast("2");

System.***out***.println(linkedList.getFirst());

System.***out***.println(linkedList.getLast());

linkedList.removeFirst();

linkedList.removeLast();

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

}

}



## 训练案例5

### 训练知识点

1.HashSet基本使用

### 训练描述

往HashSet中添加字符串"zhangsan", "lisi", "wangwu", "zhangsan".使用迭代器获取HashSet中的元素.

### 操作步骤描述

1. 创建HashSet集合
2. 使用add方法往HashSet添加元素
3. 使用迭代器获取HashSet中的元素

**package** com.day08.exam01;

**import** java.util.HashSet;

**import** java.util.Iterator;

**public** **class** Test05 {

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

HashSet<String> hashSet = **new** HashSet<>();

hashSet.add("zhangsan");

hashSet.add("lisi");

hashSet.add("wangwu");

hashSet.add("zhangsan");

Iterator<String> iterator = hashSet.iterator();

**while** (iterator.hasNext()) {

String string = (String) iterator.next();

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

}

}

}



## 训练案例6

### 训练知识点

1.hashCode和equals方法

### 训练描述

请简述HashSet去除重复元素的原理

### 操作步骤描述

1. 先判断hashCode()
2. 再判断equals()

**public** **int** hashCode() {

**final** **int** prime = 31;

**int** result = 1;

result = prime \* result + age;

result = prime \* result + ((name == **null**) ? 0 : name.hashCode());

**return** result;

}

**public** **boolean** equals(Object obj) {

**if** (**this** == obj)

**return** **true**;

**if** (obj == **null**)

**return** **false**;

**if** (getClass() != obj.getClass())

**return** **false**;

Person other = (Person) obj;

**if** (age != other.age)

**return** **false**;

**if** (name == **null**) {

**if** (other.name != **null**)

**return** **false**;

} **else** **if** (!name.equals(other.name))

**return** **false**;

**return** **true**;

}

## 训练案例7

### 训练知识点

1.HashSet存储自定义类型

### 训练描述

定义人类,包含姓名和年龄属性.创建4个人存储到HashSet中.姓名和年龄相同的人看做同一人不存储

### 操作步骤描述

1. 定义Person类.包好姓名年龄属性,重写hashCode()和equals()方法
2. 创建HashSet用于存储Person类型
3. 添加多个Person到HashSet中
4. 遍历获取HashSet中的内容

**package** com.day08.exam01;

**public** **class** Person {

**private** String name;

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

**public** Person() {

**super**();

}

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

**super**();

**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;

}

@Override

**public** **int** hashCode() {

**int** thisHashCode = 0;

thisHashCode += name.hashCode() \* 17;

thisHashCode += age;

**return** thisHashCode;

}

@Override

**public** **boolean** equals(Object obj) {

Person otherP = (Person) obj;

**if** (!**this**.name.equals(otherP.name)) {

**return** **false**;

}

**if** (**this**.age != otherP.age) {

**return** **false**;

}

**return** **true**;

}

@Override

**public** String toString() {

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

}

}

**package** com.day08.exam01;

**import** java.util.HashSet;

**public** **class** Test07 {

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

HashSet<Person> hashSet = **new** HashSet<>();

Person person1 = **new** Person("张三", 13);

Person person2 = **new** Person("李四", 13);

Person person3 = **new** Person("张三", 14);

Person person4 = **new** Person("张三", 13);

hashSet.add(person1);

hashSet.add(person2);

hashSet.add(person3);

**for** (Person person : hashSet) {

System.***out***.println(person.toString());

}

}

}



## 训练案例8

### 训练知识点

1. LinkedHashSet基本使用

### 训练描述

使用LinkedHashSet存储以下元素:"王昭君","王昭君","西施","杨玉环","貂蝉".使用迭代器和增强for循环遍历LinkedHashSet

### 操作步骤描述

1. 创建LinkedHashSet
2. 使用add方法添加元素到LinkedHashSet
3. 使用迭代器获取LinkedHashSet中的元素
4. 使用增强for获取LinkedHashSet中的元素

**package** com.day08.exam01;

**import** java.util.Iterator;

**import** java.util.LinkedHashSet;

**public** **class** Test08 {

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

LinkedHashSet<String> linkedHashSet = **new** LinkedHashSet<>();

linkedHashSet.add("王昭君");

linkedHashSet.add("王昭君");

linkedHashSet.add("西施");

linkedHashSet.add("杨玉环");

linkedHashSet.add("貂蝉");

Iterator<String> iterator = linkedHashSet.iterator();

**while** (iterator.hasNext()) {

String string = (String) iterator.next();

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

}

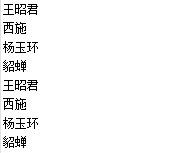
**for** (String string : linkedHashSet) {

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

}

}

}



# 关卡2

## 训练案例1

### 训练知识点

1.HashSet

### 训练描述

ArrayList有以下元素: "a","f","b","c","a","d"利用HashSet对ArrayList集合去重(最终结果: ArrayList中没有重复元素)

### 操作步骤描述

1. 创建ArrayList
2. 使用add方法往ArrayList添加元素
3. 创建HashSet.用于将ArrayList中重复的元素去除
4. 调用HashSet的addAll方法,将ArrayList中的元素添加到HashSet中
5. 清空list的所有元素
6. 将set集合中的元素再添加回ArrayList集合

**package** com.day08.exam02;

**import** java.util.ArrayList;

**import** java.util.HashSet;

**public** **class** Test01 {

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

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

arrayList.add("a");

arrayList.add("f");

arrayList.add("b");

arrayList.add("c");

arrayList.add("a");

arrayList.add("d");

HashSet<String> hashSet = **new** HashSet<>();

hashSet.addAll(arrayList);

arrayList.clear();

arrayList.addAll(hashSet);

**for** (String string : arrayList) {

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

}

}

}



## 训练案例2

### 训练知识点

1. HashSet使用

### 训练描述

向HashSet集合添加姓名{张三,李四,王五,二丫,钱六,孙七},将二丫删除,添加王小丫

### 操作步骤描述

1. 创建HashSet集合
2. 使用add方法往HashSet中添加元素
3. 使用remove方法删除HashSet中的二丫
4. 使用add方法往HashSet中添加王小丫

**package** com.day08.exam02;

**import** java.util.HashSet;

**public** **class** Test02 {

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

HashSet<String> hashSet = **new** HashSet<>();

hashSet.add("张三");

hashSet.add("李四");

hashSet.add("王五");

hashSet.add("二丫");

hashSet.add("钱六");

hashSet.add("孙七");

hashSet.remove("二丫");

hashSet.add("王小丫");

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

}

}



## 训练案例3

### 训练知识点

1. LinkedList使用
2. Collection的contains(Object o)方法
3. 增强for和迭代器

### 训练描述

已知数组存放一批QQ号码.QQ号码最长为11位,最短为5位

String[] strs = {"12345","67891",1"2347809933","98765432102","67891","12347809933"}

将该数组里面的所有qq号都存放在LinkedList中,将list中重复元素删除,将list中所有元素用两种方式打印出来

### 操作步骤描述

1. 定义QQ号码数组String[] strs
2. 创建LinkedList
3. 遍历strs获取每个qq号码
4. 判断LinkedList是否已经存在这个qq号码
5. 不存在这个qq号码则添加到LinkedList中
6. 增强for遍历LinkedList
7. 迭代器遍历LinkedList

**package** com.day08.exam02;

**import** java.util.Iterator;

**import** java.util.LinkedList;

**public** **class** Test03 {

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

String[] strs = { "12345", "67891", "12347809933", "98765432102", "67891", "12347809933" };

LinkedList<String> linkedList = **new** LinkedList<>();

**for** (String s : strs) {

**if** (!linkedList.contains(s)) {

linkedList.add(s);

}

}

**for** (String string : linkedList) {

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

}

Iterator<String> iterator = linkedList.iterator();

**while** (iterator.hasNext()) {

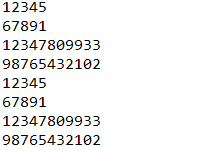
String string = (String) iterator.next();

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

}

}

}



## 训练案例4

### 训练知识点

1. HashSet

### 训练描述

定义一个Student类,包含名称,年龄,性别(姓名,年龄,性别完全相同视为同一学生)

创建10个Student对象,至少有两个学生姓名,年龄,性别完全相同. 把这10个学生添加到Set集合中,不可以重复,遍历Set集合打印学生信息,使用两种方式

### 操作步骤描述

1. 定义一个Student类,包含名称,年龄,性别重写hashCode()和equals()方法
2. 创建HashSet集合
3. 使用Student类创建10个学生,把这10个学生添加到Set集合中
4. 使用迭代器遍历HashSet
5. 使用增强for遍历集合

**package** com.day08.exam02;

**import** java.util.HashSet;

**import** java.util.Iterator;

**public** **class** Test04 {

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

HashSet<Student> hashSet = **new** HashSet<>();

Student student1 = **new** Student("张三", 18, "男");

Student student2 = **new** Student("李四", 18, "男");

Student student3 = **new** Student("张三", 19, "男");

Student student4 = **new** Student("张三", 18, "女");

Student student5 = **new** Student("张三", 18, "男");

Student student6 = **new** Student("李四", 18, "男");

Student student7 = **new** Student("王五", 18, "男");

Student student8 = **new** Student("王五", 19, "男");

Student student9 = **new** Student("王五", 18, "女");

Student student10 = **new** Student("李四", 18, "男");

hashSet.add(student1);

hashSet.add(student2);

hashSet.add(student3);

hashSet.add(student4);

hashSet.add(student5);

hashSet.add(student6);

hashSet.add(student7);

hashSet.add(student8);

hashSet.add(student9);

hashSet.add(student10);

Iterator<Student> iterator = hashSet.iterator();

**while** (iterator.hasNext()) {

Student student = (Student) iterator.next();

System.***out***.println(student.toString());

}

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

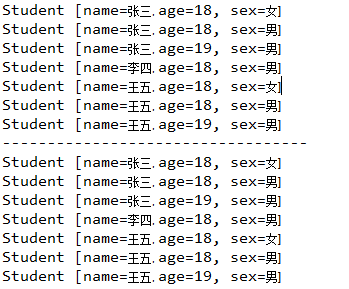
**for** (Student student : hashSet) {

System.***out***.println(student.toString());

}

}

}



## 训练案例5

### 训练知识点

1.HashSet

### 训练描述

有一个数组 String arr[]={"abc","bad","abc","aab","bad","cef","jhi"};创建一个ArrayList，将数组里面的元素添加进ArrayList，但元素不能重复(使用HashSet去除重复元素)

### 操作步骤描述

1. 创建arr数组
2. 创建HashSet集合
3. 遍历arr数组
4. 将arr数组中的元素添加进HashSet中.重复的就不会添加进去了
5. 创建ArrayList集合,将HashSet的元素添加到ArrayList中
6. 遍历ArrayList打印元素

**package** com.day08.exam02;

**import** java.util.ArrayList;

**import** java.util.HashSet;

**public** **class** Test05 {

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

String arr[] = { "abc", "bad", "abc", "aab", "bad", "cef", "jhi" };

HashSet<String> hashSet = **new** HashSet<>();

**for** (String s : arr) {

hashSet.add(s);

}

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

arrayList.addAll(hashSet);

**for** (String string : arrayList) {

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

}

}

}



## 训练案例6

### 训练知识点

1. HashSet
2. IO

### 训练描述

ArrayList集合有以下元素:{"张柏芝","刘德华","张亮","张靓颖","杨颖","黄晓明","张柏芝","刘德华"}.1.使用HashSet将重复的元素去除.2.编写一个方法:将去重后的人员写入到D:\\a.txt中.3.编写一个方法:将D:\\a.txt中所有姓张的人员信息读取出来并打印到控制台上

### 操作步骤描述

1. 定义ArrayList
2. 使用add方法往ArrayList添加指定的内容
3. 创建HashSet.将ArrayList的内容放入HashSet去除重复内容
4. 定义save(HashSet<String> hs)方法
5. 在save方法中创建BufferedWriter字符输出流
6. 在save方法中遍历获取每个名称
7. 在save方法中将名称写入指定文件中
8. 在save方法中关闭流
9. 调用save方法,传入去重后的名称
10. 定义read()方法读取指定文件内容
11. 在read()方法中创建BufferedReader字符输入流
12. 在read()方法中读取文件中的名称
13. 在read()方法中判断名称是否以张开头
14. 在read()方法中打印以张开头的名称
15. 调用read方法

**package** com.day08.exam02;

**import** java.io.BufferedReader;

**import** java.io.BufferedWriter;

**import** java.io.FileNotFoundException;

**import** java.io.FileReader;

**import** java.io.FileWriter;

**import** java.util.ArrayList;

**import** java.util.HashSet;

**public** **class** Test06 {

**public** **static** **void** save(HashSet<String> hs) **throws** Exception {

BufferedWriter bufferedWriter = **new** BufferedWriter(**new** FileWriter("a.txt"));

**for** (String string : hs) {

bufferedWriter.write(string);

bufferedWriter.write("\r\n");

}

bufferedWriter.close();

}

**public** **static** **void** read() **throws** Exception {

BufferedReader bufferedReader = **new** BufferedReader(**new** FileReader("a.txt"));

String line;

**while**((line= bufferedReader.readLine())!=**null**) {

**if**(line.charAt(0)=='张') {

System.***out***.println(line + " ");

}

}

}

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

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

arrayList.add("张柏芝");

arrayList.add("刘德华");

arrayList.add("张亮");

arrayList.add("张靓颖");

arrayList.add("杨颖");

arrayList.add("黄晓明");

arrayList.add("张柏芝");

arrayList.add("刘德华");

HashSet<String> hashSet = **new** HashSet<>();

**for** (String string : arrayList) {

hashSet.add(string);

}

*save*(hashSet);

*read*();

}

}





## 训练案例7

### 训练知识点

1.HashSet

### 训练描述

编写一个程序，获取10个1至20的随机数，要求随机数不能重复。使用迭代器把最终的结果输出到控制台。

### 操作步骤描述

1. 创建HashSet用于保存随机数
2. 创建Random用于产生随机数
3. 使用while循环判断hs的size是否小于10
4. hs的size小于10就生成一个随机数
5. 将随机数添加到HashSet,重复的随机数HashSet不会添加
6. 获取迭代器
7. 使用迭代器循环判断是否有下一个元素
8. 获取到下一个元素.打印出来

**package** com.day08.exam02;

**import** java.util.HashSet;

**import** java.util.Iterator;

**import** java.util.Random;

**public** **class** Test07 {

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

HashSet<Integer> hashSet = **new** HashSet<>();

Random random = **new** Random();

**while** (hashSet.size() < 10) {

**int** a = random.nextInt(20) + 1;

hashSet.add(a);

}

Iterator<Integer> iterator = hashSet.iterator();

**while** (iterator.hasNext()) {

Integer integer = (Integer) iterator.next();

System.***out***.print(integer+" ");

}

}

}



## 训练案例8

### 训练知识点

1. LinkedHashSet

### 训练描述

键盘录入一个字符串.去掉其中重复字符, 打印出不同的那些字符.必须保证顺序.例如输入: aaaabbbcccddd,打印结果为: abcd

### 操作步骤描述

1. 创建Scanner对象,用于键盘录入
2. 调用Scanner的nextLine()方法,让用户输入一个字符串
3. 创建LinkedHashSet.用于去除重复的字符串,并保证迭代顺序
4. 将字符串串转成char[]
5. 使用增强for循环遍历每个字符
6. 将每个字符添加到LinkedHashSet中
7. 使用增强for打印LinkedHashSet中的内容

**package** com.day08.exam02;

**import** java.util.LinkedHashSet;

**import** java.util.Scanner;

**public** **class** Test08 {

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

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

System.***out***.println("请输入一个字符串：");

String s = scanner.nextLine();

LinkedHashSet<Character> linkedHashSet = **new** LinkedHashSet<>();

**char**[] arr = s.toCharArray();

**for** (**char** c : arr) {

linkedHashSet.add(c);

}

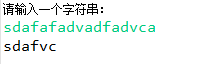
**for** (**char** c : linkedHashSet) {

System.***out***.print(c);

}

}

}



## 训练案例9

### 训练知识点

1.HashSet

### 训练描述

编写一个程序，键盘录入多个数字,直到录入-1结束.将录入的重复数据去除,计算这些不重复数据的总和和平均值

### 操作步骤描述

1. 创建Scanner对象,用于键盘录入
2. 创建HashSet.用于去除重复的数字
3. 不停循环,让用户输入数字
4. 调用Scanner的nextInt()方法,让用户输入一个数字
5. 如果用户输入-1,跳出循环
6. 不是-1将数字添加到HashSet
7. 定义求和变量sum
8. 使用增强for获取HashSet中的每个元素
9. 将每个数字添加到sum中
10. 输出求和的结果
11. 输出平均值,平均值 = 总数 / 个数

**package** com.day08.exam02;

**import** java.util.HashSet;

**import** java.util.Scanner;

**public** **class** Test09 {

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

HashSet<Integer> hashSet = **new** HashSet<>();

**int** a = 0;

**while** (a != -1) {

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

System.***out***.print("请输入数字：");

a = scanner.nextInt();

hashSet.add(a);

}

**int** sum = 0;

**for** (Integer integer : hashSet) {

sum += integer;

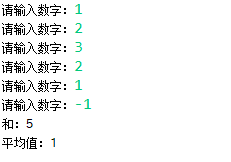
}

System.***out***.println("和：" + sum);

System.***out***.println("平均值：" + (sum / hashSet.size()));

}

}



## 训练案例10

### 训练知识点

1.HashSet

### 训练描述

有2个ArrayList,元素如下: array1 = "a","b","a","c","d",array2 = "e","f","a","d","g",将两个集合重复项去除后合并成一个集合

### 操作步骤描述

1. 创建ArrayList array1
2. 使用add方法为array1存放对应的元素
3. 创建ArrayList array2
4. 使用add方法为array2存放对应的元素
5. 创建HashSet用于存放去除重复的元素
6. 遍历array1,获取到每个元素,将每个元素添加到HashSet中
7. 遍历array2,获取到每个元素,将每个元素添加到HashSet中
8. 输出HashSet中去除重复后的元素

**package** com.day08.exam02;

**import** java.util.ArrayList;

**import** java.util.HashSet;

**public** **class** Test10 {

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

ArrayList<String> array1 = **new** ArrayList<>();

array1.add("a");

array1.add("b");

array1.add("a");

array1.add("c");

array1.add("d");

ArrayList<String> array2 = **new** ArrayList<>();

array2.add("e");

array2.add("f");

array2.add("a");

array2.add("d");

array2.add("g");

HashSet<String> hashSet = **new** HashSet<>();

**for** (String s : array1) {

hashSet.add(s);

}

**for** (String s : array2) {

hashSet.add(s);

}

**for** (String s : hashSet) {

System.***out***.print(s + " ");

}

}

}



## 训练案例11

### 训练知识点

1.HashSet

### 训练描述

自定义学生类:包含姓名,年龄,成绩属性.私有成员变量,生成无参,有参构造方法,生成get/set方法.创建5个学生放到HashSet中(姓名和年龄相同的认为是同一学生).使用迭代器获取每个学生信息.统计总分,平均分,最高分,最低分并输出

### 操作步骤描述

1. 创建HashSet,用于存放学生
2. 使用add方法添加5个学生
3. 使用迭代器获取一个学生.
4. 定义总分变量
5. 定义最高分变量,初始化值为迭代出来的学生的成绩
6. 定义最低分变量,初始化值为迭代出来的学生的成绩
7. 使用增强for循环获取遍历HashSet,获取到每个学生
8. 在增强for中将学生的分数添加到总分
9. 在增强for中判断最高分是否小于这个学生的分数,如果最高分小于这个学生的分数,将这个学生的分数设置为最高分
10. 在增强for中判断最低分是否大于这个学生的分数,如果最低分大于这个学生的分数,将这个学生的分数设置为最低分
11. 计算平均分(平均分 = 总分 / 人数)
12. 按照格式打印

**package** com.day08.exam02;

**import** java.util.HashSet;

**import** java.util.Iterator;

**public** **class** Test11 {

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

HashSet<Person> hashSet = **new** HashSet<>();

Person person1 = **new** Person("张三", 18, 60);

Person person2 = **new** Person("李四", 17, 61);

Person person3 = **new** Person("王五", 16, 62);

Person person4 = **new** Person("李四", 17, 59);

Person person5 = **new** Person("王五", 16, 58);

hashSet.add(person1);

hashSet.add(person2);

hashSet.add(person3);

hashSet.add(person4);

hashSet.add(person5);

Iterator<Person> iterator = hashSet.iterator();

**while** (iterator.hasNext()) {

Person person = (Person) iterator.next();

System.***out***.println(person.toString());

}

**int** allGrade = 0;

**int** maxGrade = 0;

**int** minGrade = 100;

**for** (Person person : hashSet) {

allGrade += person.getGrade();

**if** (maxGrade < person.getGrade()) {

maxGrade = person.getGrade();

}

**if** (minGrade > person.getGrade()) {

minGrade = person.getGrade();

}

}

**int** avgGrade = allGrade / hashSet.size();

System.***out***.println("总分：" + allGrade);

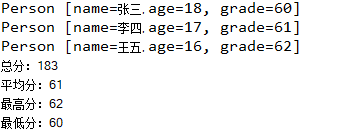
System.***out***.println("平均分：" + avgGrade);

System.***out***.println("最高分：" + maxGrade);

System.***out***.println("最低分：" + minGrade);

}

}



## 训练案例12

### 训练知识点

1. ArrayList
2. 正则表达式

### 训练描述

定义一个方法filterChars(ArrayList<String> list)将传入的ArrayList<String>集合中的每个元素中的字母过滤掉,比如list有2个元素{"hello123java", "11world555java"}，返回结果{"123", "11555"}

### 操作步骤描述

1. 定义ArrayList strs存放元素
2. 使用add方法往ArrayList中添加元素,元素包含字母和数字
3. 定义filterChars(ArrayList<String> list)方法
4. 在filterChars方法中使用for循环拿出每个元素
5. 替换掉字符串中所有的字母
6. 使用替换后的字符串覆盖原来的字符串
7. 调用filterChars方法

**package** com.day08.exam02;

**import** java.util.ArrayList;

**public** **class** Test13 {

**public** **static** **void** filterChars(ArrayList<String> list) {

String s = "";

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

s = list.get(i).replaceAll("[a-zA-Z]", "");

list.set(i, s);

}

}

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

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

strs.add("hello123java");

strs.add("11world555java");

*filterChars*(strs);

**for** (String s : strs) {

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

}

}

}



# 关卡3

## 训练案例

### 训练知识点

1.HashSet

### 训练描述

双色球规则:双色球每注投注号码由6个红色球号码和1个蓝色球号码组成。红色球号码从1—33中选择；蓝色球号码从1—16中选择.请随机生成一注双色球号码

**package** com.day08.exam03;

**import** java.util.HashSet;

**import** java.util.Random;

**public** **class** Test01 {

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

HashSet<Integer> hashSet = **new** HashSet<>();

Random random = **new** Random();

**while** (hashSet.size() < 6) {

hashSet.add(random.nextInt(33) + 1);

}

**while** (hashSet.size() == 6) {

hashSet.add(random.nextInt(16) + 1);

}

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

}

}



## 训练案例2

### 训练知识点

1. HashSet
2. 增强for

### 训练描述

一个学科中有若干班级，每一个班级又有若干学生。整个学科一个大集合，若干个班级分为每一个小集合(集合嵌套之HashSet嵌套HashSet)。要求如下

1、 学生类有两个属性，姓名和年龄，并定义有参构造、无参构造和getter/setter方法.姓名和年龄相同的视为同一学生

2、向班级集合中添加若干个学生（至少两个学生）。

2、向学科集合中添加所有班级(至少创建两个班级)。

3、使用两种方法遍历学科集合，并打印出所有学生。

**package** com.day08.exam03;

**import** java.util.HashSet;

**import** java.util.Iterator;

**public** **class** Test02 {

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

HashSet<HashSet<Student>> hashSet = **new** HashSet<HashSet<Student>>();

HashSet<Student> firstClass = **new** HashSet<Student>();

firstClass.add(**new** Student("张三", 16));

firstClass.add(**new** Student("李四", 17));

HashSet<Student> SecondClass = **new** HashSet<Student>();

firstClass.add(**new** Student("王五", 16));

firstClass.add(**new** Student("赵六", 17));

hashSet.add(firstClass);

hashSet.add(SecondClass);

**for** (HashSet<Student> hashSet2 : hashSet) {

**for** (Student student : hashSet2) {

System.***out***.println(student.toString());

}

}

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

Iterator<HashSet<Student>> iterator = hashSet.iterator();

**while** (iterator.hasNext()) {

HashSet<Student> hashSet2 = (HashSet<Student>) iterator.next();

Iterator<Student> iterator2 = hashSet2.iterator();

**while** (iterator2.hasNext()) {

Student student = (Student) iterator2.next();

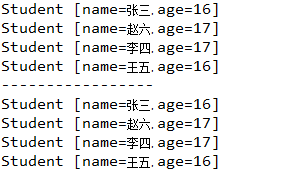
System.***out***.println(student.toString());

}

}

}

}



## 训练案例3

### 训练知识点

1.循环

### 训练描述

有一堆零件数量不超过1000个,如果4个零件一组最后剩余2个,如果7个零件一组最后剩余3个,如果9个零件一组最后剩余5个,请计算这堆零件个数有几种可能

**package** com.day08.exam03;

**import** java.util.HashSet;

**public** **class** Test03 {

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

HashSet<Integer> hashSet = **new** HashSet<>();

**for** (**int** i = 1; i <= 1000; i++) {

**if** (i % 4 == 2 && i % 7 == 3 && i % 9 == 5) {

hashSet.add(i);

}

}

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

}

}



## 训练案例4

### 训练知识点

1.嵌套循环

### 训练描述

将50元兑换成5元,2元和1元.每种面额的钱不能为0张.共有几种兑换方法.每种兑换方法的钱是怎么样的?

**package** com.day08.exam03;

**public** **class** Test04 {

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

**for** (**int** x = 1; x < 10; x++) {

**for** (**int** y = 1; y < 25; y++) {

**for** (**int** z = 1; z < 50; z++) {

**if** (x \* 5 + y \* 2 + z == 50) {

System.***out***.println("5元：" + x + "张 " + "2元：" + y + "张 " + "1元：" + z + "张");

}

}

}

}

}

}