**深 圳 大 学 实 验 报 告**

**课程名称： Java 程序设计**

**实验项目名称： 实验3 常用集合类和线程**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术（创新班）**

**指导教师： 姚俊梅**

**报告人： 李文俊 学号： 2023150001 班级：高性能班**

**实验时间：2024年10月29日 ~ 2024年11月23日**

**实验报告提交时间： 2024年11月22日**

**教务部制**

|  |
| --- |
| **实验目的与要求：**  **实验目的：**掌握Java程序设计开发环境的搭建，掌握编译、运行等基本步骤和命令；掌握基础知识和基本类型应用，以及类的初级应用。  **实验要求：**  1、String类的应用以及正则表达式的使用（20分，每题10分）  (1) 编写一个Java应用程序，读取一个以空格分隔的字符串，并将其解析为一个包含单词数量的统计信息。每个单词的统计信息包括单词本身和该单词出现的次数。具体要求如下：编写一个方法 countWords(String input)，对输入input字符串进行解析并返回另外一个格式为"word1: count1; word2: count2; ...; wordN: countN"的字符串。请用如下字符串进行测试：  "Tom and Jerry is an American animated media franchise and series of comedy short films created in 1940 by William Hanna and Joseph Barbera. Best known for its 161 theatrical short films by Metro-Goldwyn-Mayer, the series centers on the enmity between the titular characters of a cat named Tom and a mouse named Jerry. Many shorts also feature several recurring characters."  (2) 编写一个Java应用程序，每次从键盘读入一个包含有大写英文字母、小写英文字母和数字混杂的字符串(例如Aa123bEFGaaa49023)，要求按反转顺序输出大写英文字母（例如GFEA）、小写英文字母（aaaba）和数字（32094321）。要求循环连续测试10次，在报告中附上程序截图、完整的运行结果截图和简要文字说明。  2、集合类的应用（40分，每题10分）  (1) 编写一个Java应用程序，读取一个CSV格式的字符串并将其解析为一个包含学生信息的对象列表。每个学生的信息包括姓名、年龄和成绩。要求如下：i) 首先定义一个 Student 类，包含name（String，表示学生的姓名）、age（int，表示学生的年龄）及score（float，表示学生的成绩）；ii) 编写一个 StudentParser 类，包含方法List<Student> parseStudents(String csvData)，该方法接收一个以逗号分隔的字符串（CSV格式），并返回一个 Student 对象的列表。请使用以下CSV格式的字符串进行测试：  Alice,20,88.5  Bob,22,75.0  Charlie,19,90.0  (2) 编写Java应用程序，使用集合类管理图书信息。该系统能够存储图书信息、查找特定图书以及列出所有图书。要求如下：  i) 创建Book类，包含属性title（String，图书标题）、author（String，作者姓名）及isbn（String，ISBN号）。  ii) 创建BookManager类，使用 HashSet<Book> 来存储图书信息，提供的方法包括void addBook(Book book)用于将图书添加到集合中，Book findBookByTitle(String title) 根据标题查找并返回对应的图书，方法 void listAllBooks() 输出所有存储的图书信息。  iii) 在主类的 main 方法中创建 BookManager 实例，并演示添加、查找和列出图书的功能。请使用如下两种图书信息进行测试：  《Java编程思想》，作者：Bruce Eckel，ISBN：978-7-111-23866-3  《Effective Java》，作者：Joshua Bloch，ISBN：978-7-121-29968-0  (3) 编写一个Java应用程序，按照GDP从小到大排序输出10个国家的信息，要求：i) 创建一个Country类，包括属性name、gdp2023和population，分别表示国家名称、2023年的国内生产总值（单位：百万美元）和人口总数。ii) 使用TreeMap<K,V>进行排序，通过Comparator接口和Comparable接口两种方式实现。  请自行查找中国、美国、德国、法国、日本、英国、意大利、韩国、印度、泰国这10个国家的GDP和人口信息，将这10个国家的信息添加倒TreeMap中，并输出排序后的国家信息。  (4) 编写Java应用程序。新建一个ArrayList对象、一个LinkedList对象，一个HashSet对象、一个LinkedHashSet对象、以及一个TreeSet对象。每个对象都包含1到50000的所有整数，顺序随机。现进行如下测试：i) 产生10000个随机整数，每个对象调用contains方法判断这些整数是否在该对象中；ii）计算每个对象执行这10000次操作所需要的时间，并进行对比分析。  3、线程同步（30分，每题10分）  (1) 运行以下三个程序（每个程序运行10次），并对输出结果给出分析。在报告中附上程序截图和详细的文字说明。（10分）  **程序1:**  **程序2:**  **程序3:**  (2) 编写Java应用程序实现如下功能：第一个线程输出数字1,2,..,12，第二个线程输出英文单词数字和月份One January, Two February, …, Twelve December，输出的顺序和格式为1OneJanuary2TwoFebruary...12TwelveDecember，即每1个数字紧跟着2个英文单词的方式。要求线程间实现通信。要求采用实现Runnable接口和Thread类的构造方法的方式创建线程，而不是通过Thread类的子类的方式。在报告中附上程序截图、运行结果截图和详细的文字说明。  (3) 编写Java应用程序实现如下功能：创建工作线程，模拟银行现金账户取款操作。多个线程同时执行取款操作时，如果不使用同步处理，会造成账户余额混乱，要求使用syncrhonized关键字同步代码块，以保证多个线程同时执行取款操作时，银行现金账户取款的有效和一致。要求采用实现Runnable接口和Thread类的构造方法的方式创建线程，而不是通过Thread类的子类的方式。在报告中附上程序截图、运行结果截图和详细的文字说明。  报告写作。要求：主要思路有明确的说明，重点代码有详细的注释，行文逻辑清晰可读性强，报告整体写作较为专业。（10）  **说明：**  （1）本次实验课作业满分为100分。  （2）报告正文：请在指定位置填写，本次实验不需要单独提交源程序文件。  （4）个人信息：WORD文件名中的“姓名”、“学号”，请改为你的姓名和学号；实验报告的首页，请准确填写“学院”、“专业”、“报告人”、“学号”、“班级”、“实验报告提交时间”等信息。  （5）提交方式：截至时间前，请在Blackboard平台中提交。  （6）发现抄袭（包括复制&粘贴整句话、整张图），**抄袭者和被抄袭者的成绩记零分（含抄袭往届同学的作业）。**  （7）延迟提交将酌情扣分；期末考试阶段补交无效。  1、String类的应用以及正则表达式的使用（20分，每题10分）  (1) 编写一个Java应用程序，读取一个以空格分隔的字符串，并将其解析为一个包含单词数量的统计信息。每个单词的统计信息包括单词本身和该单词出现的次数。具体要求如下：编写一个方法 countWords(String input)，对输入input字符串进行解析并返回另外一个格式为"word1: count1; word2: count2; ...; wordN: countN"的字符串。请用如下字符串进行测试：  Tom and Jerry is an American animated media franchise **and** series of comedy short films created in 1940 by William Hanna and Joseph Barbera. Best known for its 161 theatrical short films by Metro-Goldwyn-Mayer, the series centers on the enmity between the titular characters of a cat named Tom and a mouse named **Jerry**. Many shorts also feature several recurring characters.  1.读取字符串  导包java.util,声明字符串input和Scanner对象scanner，调用scanner的nextLine方法，可以读取**有空格的完整字符串**  2.删去逗号和句号  String创建的对象是无法修改的，因此如果想要删去逗号和句号，需要用到StringBuffer类，运用构造方法StringBuffer(String str)，然后遍历所有元素，调用方法CharAt访问每个元素，如果是逗号和句号，调用方法delete(int startindex,int endindex)可以删除startindex-endindex-1的所有字符。  3.分离并统计所有单词  运用StringTokenizer类可以将空格当做分隔符分离所有单词，注意：因为StringTokenizer st需要String变量作为构造参数，所以需要把StringBuffer对象调用toString方法转化为String变量，创造临时words和count数组，调用st.hasMoreTokens，将st.NextTokens()赋给给个words[i]，count全赋为1  4.删除重复部分  如果把count[i]=1当做没遍历过，那么count[i]=-1表示前面已经有这个单词，当再次读到这个单词时就可以通过count[i]是否等于-1判断是否要跳过这个单词。  创建新的String [] newwords,int[] count2来存储最后的结果。Cnt表示实际单词计数器 。注意**字符串相等内容是用str1.equals(str2) ，==表示两个字符串有同一个地址**  注意这里已经不能再用st.countTokens()为两个数组声名大小，因为前面调用nextTokens()已经改变了st的位置。  5.最后输出  6.运行结果  7.完整代码  8.后续可优化  可以将大写字母改成小写，可以解决首字母大写导致单词分开统计的问题。  (2) 编写一个Java应用程序，每次从键盘读入一个包含有大写英文字母、小写英文字母和数字混杂的字符串(例如Aa123bEFGaaa49023)，要求按反转顺序输出大写英文字母（例如GFEA）、小写英文字母（aaaba）和数字（32094321）。要求循环连续测试10次，在报告中附上程序截图、完整的运行结果截图和简要文字说明。  1.设置测试10次循环，输入  2.存储翻转结果  For从右往左循环遍历所有字符串，创建三个StringBuffer类对象来存储最后结果，通过调用Character的isUpperCase,isLowerCase和isDigit(char ch)可以判断字符的类型，str.charAt(index)访问字符  然后动态地将字符通过StringBuffer的append方法接入到对应对象后面  3.输出结果  4.测试结果  5.完整代码  **2、集合类的应用（40分，每题10分）**  (1) 编写一个Java应用程序，读取一个**CSV格式**的字符串并将其解析为一个包含学生信息的**对象列表**。每个学生的信息包括**姓名、年龄和成绩**。要求如下：i) 首先定义一个 Student 类，包含name（String，表示学生的姓名）、age（int，表示学生的年龄）及score（float，表示学生的成绩）；ii) 编写一个 StudentParser 类，包含方法List<Student> parseStudents(String csvData)，该方法接收一个以逗号分隔的字符串（CSV格式），并返回一个 Student 对象的列表。请使用以下CSV格式的字符串进行测试：  Alice,20,88.5  Bob,22,75.0  Charlie,19,90.0  1.定义一个 Student 类  包含name（String，表示学生的姓名）、age（int，表示学生的年龄）及score（float，表示学生的成绩），有参构造方法（用于后面List元素创建），重写了 toString 方法，用于格式化输出图书信息。  2.定义StudentParser类  包含方法List<Student> parseStudents(String csvData)，该方法接收一个以逗号分隔的字符串（CSV格式），并返回一个 Student 对象的列表。具体实现：根据测试，csvDate每一行有\n分离，每一行中又由‘,’分离，所以需要使用两次String [] split(string redix)，注意转义字符要用两个\\表达，最后用list.add(Student std)将元素存储到表中。  3.编写主方法  根据题目要求, List<Student> parseStudents(String csvData)参数是不可修改的String类型，所以先用StringBuffer存储主方法中的csvData，每次输入再把每一行输入和换行符通过append()存储到csvData中，最后只需调用toString传入到方法。最后输出list中每一个元素  4．测试结果  程序可满足多次输入，通过Boolean isEmpty()来判断输入是否结束，如果读到\n，程序就会结束。  5.完整代码  (2) 编写Java应用程序，使用集合类管理图书信息。该系统能够存储图书信息、查找特定图书以及列出所有图书。要求如下：  i) 创建Book类，包含属性title（String，图书标题）、author（String，作者姓名）及isbn（String，ISBN号）。  ii) 创建BookManager类，使用 HashSet<Book> 来存储图书信息，提供的方法包括void addBook(Book book)用于将图书添加到集合中，Book findBookByTitle(String title) 根据标题查找并返回对应的图书，方法 void listAllBooks() 输出所有存储的图书信息。  iii) 在主类的 main 方法中创建 BookManager 实例，并演示添加、查找和列出图书的功能。请使用如下两种图书信息进行测试：  《Java编程思想》，作者：Bruce Eckel，ISBN：978-7-111-23866-3  《Effective Java》，作者：Joshua Bloch，ISBN：978-7-121-29968-0  1.创建Book类  包含属性title（String，图书标题）、author（String，作者姓名）及isbn（String，ISBN号），有参构造方法（用于后面Set元素添加），重写的toString方法，方便后面输出结果。  2. 创建BookManager类  使用 HashSet<Book> 来存储图书信息，所以HashSet<Book> books所为成员，提供的方法包括void addBook(Book book)用于将图书添加到集合中，调用HashSet的add()方法；Book findBookByTitle(String title) 根据标题查找并返回对应的图书，如果没找到，则返回null；方法 void listAllBooks() 输出所有存储的图书信息。  3.编写主类和main方法  创建 BookManager 实例，并演示添加、查找和列出图书的功能。使用两种图书信息进行测试。分别对应对象book1,book2，调用BookManager中添加，查找和列出图书方法进行测试。  4.测试结果  5.完整代码  6.改善  当添加相同的书籍时，Hashset无法判断两个书籍是否相同，和集合不能有相同的元素矛盾，所以需要重写Book类的equals 和 hashCode。当向 HashSet 中添加元素时：首先调用对象的 hashCode 方法确定哈希集的位置。如果该位置已有元素，HashSet 会通过 equals 方法比较这些元素是否与新元素相等，从而决定是否接受新元素。  7.测试结果  只输出了一次说明重复元素未被加入  (3) 编写一个Java应用程序，按照GDP从小到大排序输出10个国家的信息，要求：i) 创建一个Country类，包括属性name、gdp2023和population，分别表示国家名称、2023年的国内生产总值（单位：百万美元）和人口总数。ii) 使用TreeMap<K,V>进行排序，通过Comparator接口和Comparable接口两种方式实现。  请自行查找中国、美国、德国、法国、日本、英国、意大利、韩国、印度、泰国这10个国家的GDP和人口信息，将这10个国家的信息添加倒TreeMap中，并输出排序后的国家信息。   1. 创建一个Country类   包括属性name、gdp2023和population，分别表示国家名称、2023年的国内生产总值（单位：百万美元）和人口总数。有参构造方法（用于后面TreeMap元素添加），重写的toString方法，方便后面格式化输出结果。执行了Comparable接口进而重写compareTo方法，用来TreeMap按照GDP从小到大排序。   1. 创建十个国家对象   国家名称、2023年的国内生产总值（单位：百万美元）和人口总数创建中国、美国、德国、法国、日本、英国、意大利、韩国、印度、泰国。  3.使用TreeMap<K,V>进行Comparable排序  Comparable实现的排序调用TreeMap的无参构造方法，将十个元素通过put()添加到映射中，最后遍历并输出TreeMap的Key即可。  4. 使用TreeMap<K,V>进行Comparator排序  Comparator需要调用TreeMap的有参构造方法TreeMap<K,V>(Comparator<K> comp)，而Comparator<K> comp可以使用*comparingDouble*(c->c.gdp2023)快捷来表示比较元素的gdp2023，而不用再写一个comparator类，然后重写compare方法。  **Comparator.comparingDouble** 是 Comparator 的静态方法，Lambda 表达式 c -> c.gdp2023，c 是参数，表示传入的 Country 对象。c.gdp2023 是返回值，表示从 Country 对象中提取 gdp2023 的值。  5.测试结果  (4) 编写Java应用程序。新建一个ArrayList对象、一个LinkedList对象，一个HashSet对象、一个LinkedHashSet对象、以及一个TreeSet对象。每个对象都包含1到50000的所有整数，顺序随机。现进行如下测试：i) 产生10000个随机整数，每个对象调用contains方法判断这些整数是否在该对象中；ii）计算每个对象执行这10000次操作所需要的时间，并进行对比分析。  1.创建对象  建一个ArrayList对象、一个LinkedList对象，一个HashSet对象、一个LinkedHashSet对象、以及一个TreeSet对象，泛型类对应类型为Integer  2.创建50000个随机整数  首先创建一个List存储五万个整数，从1到50000，然后调用Collections的静态方法shuffle打乱五万个整数的顺序，将这些数都存到五个对象中  3.生成随机数  运用Random类产生1-50000的随机数放入测试List中，用于后续contain测试  注意：random.nextInt默认0-index，所以需要加1.  4．测试时间  编写一个测试时间方法，参数为(集合名称，需要测试的集合，测试元素集合)，方法内部：记录开始时间（用到了System.nanoTime（），单位是纳秒，更加精准），循环进行一万次contain()方法，记录结合时间，输出时间差。主方法中调用测试时间方法。  5.测试结果  根据集合特性：  **ArrayList**: 顺序存储，contains 需要线性查找，时间复杂度为 O(n)  **LinkedList**: 链表存储，contains 也需要线性查找，时间复杂度为 O(n)。  **HashSet**: 基于哈希表，contains 时间复杂度为 O(1)（理论上）。  **LinkedHashSet**: 继承自 HashSet，保留插入顺序，contains 时间复杂度为 O(1)。  **TreeSet**: 基于红黑树，contains 时间复杂度为 O(logn)。  测试结果基本符合集合特性，存在差异可能与硬件相关，总体来看：线性表性能较差，哈希表最快，红黑树次之。  6.完整代码  **3、线程同步（30分，每题10分）**  (1)运行以下三个程序（每个程序运行10次），并对输出结果给出分析。在报告中附上程序截图和详细的文字说明。（10分）  1.程序1截图：  2.输出结果  输出结果都是先是ab混杂输出：开始输出a，然后ab交替输出，ab输出完之后开始输出从1到100的数字。输出结果不一致在于ab交替输出的位置不同。  3.分析输出结果  **PrintChar类**：实现Runnable接口，负责打印指定的字符若干次。run()方法中通过循环打印字符。**PrintNum类**：实现Runnable接口，负责打印从1到lastNum的数字。run()方法中通过循环打印数字。**主类**在main方法中创建了3个线程thread1-3。  输出结果依赖当前CPU资源的使用情况，由于Java中线程的调度是由线程调度器管理的，线程的执行顺序和交替输出是不可预测的。因此，每次运行的结果可能会不同。3个线程在CPU之间竞争，它们的输出可能会交替出现或部分重叠。  对于程序1的三个线程，从输出结果上可以看到thread3在thread1和thread2死亡后才开始执行输出语句，说明线程1和线程2的输出优先完成，原因在于它们的任务简单，只需单一输出字符，执行速度快，而且JVM调度线程的时间片分配使得它们优先被执行，输出缓冲区处理字符的开销低于处理数字的开销，使得程序先输出完两个字符，再从1到100输出数字。  这种方法对于单个任务执行很方便，但是对于大量任务来说效率不高，因为必须为每个任务创建一个线程。为每个任务启动一个新线程可能会限制吞吐量并导致性能下降。  程序2:  1.程序截图  2.输出结果  输出结果和程序1总体类似,出现字符和数字交替输出的情况。  3.结果分析  使用线程池是管理并发执行任务数量的理想方法。Java为执行线程池中的任务提供了Executor接口，为管理和控制任务提供了ExecutorService接口。ExecutorService是Executor的子接口。  该程序使用ExecutorService大小固定为 3 的线程池并发执行三个任务：任务 1：打印字符'a'100 次，任务 2：打印字符'b'100 次，任务 3 ：打印从1到100的数字。  使用ExecutorService执行任务，因为与直接使用Thread相比，它更高效、可扩展且更易于管理。线程被重复使用，与每次创建新线程相比减少了开销  程序3:  1.程序截图  2.输出结果  3.结果分析  这个程序模拟了多线程并发更新共享资源（Account类中的balance变量）的场景，但由于缺乏适当的同步机制，可能导致**竞态条件**问题。竞态条件会导致多个线程同时访问和修改共享数据时出现不可预测的结果。  **理论上：**每个线程向账户存入,100个线程最终结果应该是100.  **实际上：**多个线程可能在尚未更新balance之前读取相同的值，最终多个线程将更新后相同的newbalance赋值给balance，导致账户可能只增加了1次。  **深层原因：**deposit方法没有被同步，导致多个线程同时访问balance变量。即使Thread.sleep(5)模拟了延迟，也无法保证线程的执行顺序，因此可能导致balance值被覆盖。为了保证线程安全，应该在deposit方法上添加同步。  (2) 编写Java应用程序实现如下功能：第一个线程输出数字1,2,..,12，第二个线程输出英文单词数字和月份One January, Two February, …, Twelve December，输出的顺序和格式为1OneJanuary2TwoFebruary...12TwelveDecember，即每1个数字紧跟着2个英文单词的方式。要求**线程间实现通信**。要求采用实现Runnable接口和Thread类的构造方法的方式创建线程，而不是通过Thread类的子类的方式。在报告中附上程序截图、运行结果截图和详细的文字说明。  1.编写打印类  我的想法是将两个线程绑定在一个目标对象上，Print类实现了Runnable接口，定义两个String name1,name2，将两个线程与名字绑定，从而能够通过名字来区分线程。  2.编写线程同步的run方法  程序在0到11的循环中实现，表示循环十二次，同时对应1-12，通过当前线程获取名字来匹配是哪个线程，如果是名字为Number，就输出数字，如果是Month，就输出月份。  关键的是怎么将两个线程实现通信，如果不实现，程序就会先把数字输出完，再输出月份，这满足的是一个线程，所以这里我运用了wait() 方法来控制两个线程，只有当另一个线程输出完执行notifyAll()，当前线程才会继续执行。什么时候才执行等待呢？那就要再增加一个状态变量isNumber，如果isNumber为true，说明当前要输出数字，月份则需要等待。  3.编写main方法  创建Print类目标对象，Thread类创建两个线程绑定在一个目标对象上，将两个线程绑定到两个不同的名字，最后启动线程。  4.运行结果截图  5.完整代码  (3) 编写Java应用程序实现如下功能：创建工作线程，模拟银行现金账户取款操作。多个线程同时执行取款操作时，如果不使用同步处理，会造成账户余额混乱，要求使用syncrhonized关键字同步代码块，以保证多个线程同时执行取款操作时，银行现金账户取款的有效和一致。要求采用实现Runnable接口和Thread类的构造方法的方式创建线程，而不是通过Thread类的子类的方式。在报告中附上程序截图、运行结果截图和详细的文字说明。  1.编写账户类  一个成员变量balance表示账户余额，这是一个目标对象共用的全局变量，有参构造方法设置余额。Synchronized修饰的取款方法，计算取款后的newBalance，如果余额小于0，输出取款失败，否则将值赋值给balance，显示新的余额。  2.编写退款类  成员由一个Account变量和要取款的amount组成，有参构造方法赋值，重写的run方法执行account变量的取款方法，因为withdraw已有synchronized修饰，所以run方法可以不用重复修饰，保证每个线程执行时对应的是一个账户余额。如果只有run方法被synchronized修饰，不同线程执行withdraw时还是存在竞争。  3.编写main方法  创造一个全局对象成员account，设置余额为1000,以创建3个线程为例，首先创建3个Withdraw对象，设置不同的取款金额，以这三个为参数分别构造3个Thread线程，最后启动。  4.运行结果  **输出结果符合实际取款**  5.完整代码 |

|  |
| --- |
| ++++++++++++++++++++++++++++++++++++++++++++++++++++++  **其他（例如感想、建议等等）。**  **感想：**本月的学习内容涵盖了字符串操作、常用类、泛型集合框架和线程编程。这些知识点非常实用，也为后续深入学习和项目开发打下了坚实基础。通过学习String类及相关工具，更加熟悉了字符串的操作方法，提高了代码处理文本的效率。泛型集合框架的学习让我理解了如何高效管理和操作数据结构，而线程编程让我体会到了并发处理的重要性以及同步的复杂性。在学习的过程中，完成了不少实践题目，帮助我加深了对概念的理解，也提升了编码的熟练度。尤其是线程同步相关的题目，虽然有些挑战，但解决后成就感十足。  总体来说，这个月的学习让我对 Java 编程有了更系统的认识，也让我感受到不断提高编程能力的乐趣和意义。希望在接下来的学习中，能够继续巩固所学知识，并进一步挑战更复杂的编程任务！  **建议：**对于线程同步和集合框架，还可以学习它们的底层实现原理，比如 synchronized 的机制、HashMap 的哈希算法等，这将对编写高效、稳定的代码非常有帮助。还可以进一步熟练正则表达式，这种简洁高效的表达方式。对于写过的代码，可以继续思考怎么优化，将复杂的表达简单化，关注代码的可读性和可维护性。 |

深圳大学学生实验报告用纸

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  2024年 月 日 |
| 备注： |

注：1、报告内的项目或内容设置，可根据实际情况加以调整和补充。

2、教师批改学生实验报告时间应在学生提交实验报告时间后10日内。