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2 类的高级应用、包、继承和接口回调

学院： 计算机与软件学院

专业： 软件工程

指导教师： 蔡树彬

报告人： 江浩 学号： 2022150083 班级： 软工2A

实验时间： 2024年10月1日-10月22日

实验报告提交时间： 2024年10月28日

教务部制

|  |
| --- |
| 实验目的与要求：  实验目的：   1. 熟悉面向对象编程中类的编写及应用。 2. 熟悉集合类的应用，掌握接口的定义、实现类的编写和接口回调等技术。 3. 熟悉异常和IO的应用，掌握异常的抛出与处理、文件输入输出处理等技术。   实验要求：   1. 观看优课上Java程序设计课程视频的第3-8章 2. 完成优课上Java程序设计课程第3-8章的课后习题 3. 完成蓝桥云课上Java简明教程的面向对象、常用类、泛型和集合、异常、IO和Lambda（可选）实验 4. 使用Java解决“二叉搜索树的中序遍历”问题   二叉搜索树是一棵二叉树，树中每个节点左侧各后代节点的值均严格小于节点值，右侧各后代节点的值均严格大于节点值。现需按中序遍历（左根右）的方式，输出二叉搜索树各节点的值。如右图，应输出1 5 7 8 10 12  可以使用逐渐添加左右子节点（子树）的方式构建输入二叉树，再进行输出。测试用例的输入、输出与设计、使用可查阅https://www.kdocs.cn/l/ct3xHtMjbRGa。   1. “学生得分”问题的迭代    1. OOP课程老师小欧创建了一个学生数组students，students[i]表示一个学生对象，并且该对象具有姓名（name，String）和考试分数（score，Float）2个属性。现需按考试分数降序返回对应学生姓名，并且若考试分数相同，原下标小的同学排在前面。该如何解？    2. 小欧老师给学生类增加了更多的属性，包括学号、身高、体重、出生日期等信息，还设计了BMI(体重kg÷身高m的平方)等计算方法；也增加了更多的如按年龄、身高、BMI等排序进行输出的方式。并且，为了测试前述代码，小欧老师还构思了一个可以随机生成多个同学信息的“班级学生生成器”（ClassStudentGenerator)，请帮小欧老师实现前述代码并进行测试输出。    3. 因为学生可以选多个课程，小欧老师现在又增加了Course类（具有课程名、修读学期等属性）,而且课程的计分方式允许有两种，一种是分数（0-100），一种是等级（ABCDF），不同课程可以有不同的计分方式，但一门课程只能选择一种计分方式。那又多了2种得分排序输出的需求，1种是按每门课程的成绩降序输出对应学生姓名和成绩，并且若成绩相同，学号小的同学排在前面；另1种是对每个学生，按课程修读学期、成绩降序排序（先计分再等级），输出所修读的各个课程及其成绩？你又该如何实现前述功能？    4. 在课程成绩登记表中，按学号从小到大，依次登记了每个同学的课程得分。汇总的登记表格式如下：第一行是课程名、修读学期、课程学分、学生人数n和计分方式，接下来n行，每行是学号、姓名和成绩，然后是下一课程，例如：  Java程序设计 第3学期 3学分 1人 分数制  2023150XXX 张三 99  Web程序设计 第5学期 3学分 1人 等级制  2023180XXX 李四 B   现输入汇总登记CourseGrade.txt，请输出汇总个人成绩StudentGrade.txt，文件中先按学号升序，对每个学生，按课程修读学期升序、成绩降序排序（先计分再等级），输出学生修读的学分和各课程的成绩。然后是下一学生，例如：  2023150XXX 张三 选修10学分 获得9学分 //60（或D）及以上获得学分  Java程序设计 第3学期 3学分 99  Web程序设计 第5学期 3学分 A  你又该如何实现前述功能？（还可以改进“生成器”进行自测） |
| 实验过程：   1. 优课上Java程序设计课程视频第3-8章观看完成的截图      1. 优课上Java程序设计课程第3-8章课后习题的完成情况及得分截图      1. 蓝桥云课上Java简明教程面向对象、常用类、泛型和集合、异常、IO和Lambda（可选）实验完成情况截图        1. 二叉搜索树的中序遍历问题测试用例的设计和选用说明 2. 二叉搜索树的中序遍历问题的解题思路（使用流程图或伪代码）和核心代码说明 3. 学生得分问题的首次设计思路（使用流程图或伪代码）、核心代码说明和具体测试结果展示 4. 学生得分问题的第2次设计思路（使用流程图或伪代码）、核心代码说明和具体测试结果展示 5. 学生得分问题的第3次设计思路（使用流程图或伪代码）、核心代码说明和具体测试结果展示 6. 学生得分问题的第4次设计思路（使用流程图或伪代码）、核心代码说明和具体测试结果展示 |
| 4.使用Java解决“二叉搜索树的中序遍历”问题  二叉搜索树是一棵二叉树，树中每个节点左侧各后代节点的值均严格小于节点值，右侧各后代节点的值均严格大于节点值。现需按中序遍历（左根右）的方式，输出二叉搜索树各节点的值。如右图，应输出1 5 7 8 10 12  可以使用逐渐添加左右子节点（子树）的方式构建输入二叉树，再进行输出。测试用例的输入、输出与设计、使用可查阅https://www.kdocs.cn/l/ct3xHtMjbRGa。    测试1：按照题目给的测试用例进行了二叉树的构建：  */\*\*  \** ***@BelongsProject:*** *test0924  \** ***@BelongsPackage:*** *PACKAGE\_NAME  \** ***@ClassName*** *BinarySearchTreeInorderTraversal  \** ***@Author:*** *bill  \** ***@CreateTime:*** *2024-11-26 09:34  \** ***@Description:*** *TODO  \** ***@Version:*** *1.0  \*/  /\*\*  \*  \** ***@author*** *dell  \*/ // 定义二叉树的节点* class TreeNode {  int val;  TreeNode left;  TreeNode right;   TreeNode(int val) {  this.val = val;  this.left = null;  this.right = null;  } }  public class BinarySearchTreeInorderTraversal {   *// 中序遍历函数* public void inorderTraversal(TreeNode root) {  if (root == null) {  return;  }  inorderTraversal(root.left); *// 递归遍历左子树* System.out.print(root.val + " "); *// 输出当前节点值* inorderTraversal(root.right); *// 递归遍历右子树* }   *// 主函数，构建二叉树并执行中序遍历* public static void main(String[] args) {  BinarySearchTreeInorderTraversal tree = new BinarySearchTreeInorderTraversal();   *// 构建二叉搜索树* TreeNode root = new TreeNode(8);  root.left = new TreeNode(5);  root.right = new TreeNode(10);  root.left.left = new TreeNode(1);  root.left.right = new TreeNode(7);  root.right.right = new TreeNode(12);   *// 执行中序遍历* System.out.print("Inorder Traversal: ");  tree.inorderTraversal(root);  } }  输出正确：     1. 测试2：极端情况测试，树形非常倾斜（例如只有右子树或左子树），验证代码能正确处理这种情况。例如，测试树为单链式结构：   */\*\*  \** ***@BelongsProject:*** *test0924  \** ***@BelongsPackage:*** *PACKAGE\_NAME  \** ***@ClassName*** *BinarySearchTreeInorderTraversal  \** ***@Author:*** *bill  \** ***@CreateTime:*** *2024-11-26 09:34  \** ***@Description:*** *TODO  \** ***@Version:*** *1.0  \*/  /\*\*  \*  \** ***@author*** *dell  \*/ // 定义二叉树的节点* class TreeNode {  int val;  TreeNode left;  TreeNode right;   TreeNode(int val) {  this.val = val;  this.left = null;  this.right = null;  } }  public class BinarySearchTreeInorderTraversal {   *// 中序遍历函数* public void inorderTraversal(TreeNode root) {  if (root == null) {  return;  }  inorderTraversal(root.left); *// 递归遍历左子树* System.out.print(root.val + " "); *// 输出当前节点值* inorderTraversal(root.right); *// 递归遍历右子树* }   *// 主函数，构建二叉树并执行中序遍历* public static void main(String[] args) {  BinarySearchTreeInorderTraversal tree = new BinarySearchTreeInorderTraversal();   *// 构建二叉搜索树  // 只有一个节点* TreeNode root = new TreeNode(5);     *// 执行中序遍历* System.out.print("Inorder Traversal: ");  tree.inorderTraversal(root);  } }  输出正确：  实验结论：   1. 二叉搜索树的中序遍历问题是否通过自行或他人设计的测试用例？ 2. 4次迭代的学生得分问题能否通过自行设计的测试用例？ 3. 回顾整个实验过程中，你是否有遇到什么困难，你是如何解决的？ 4. 做完实验后，你最大的收获是什么？ |
| 心得体会：  通过本次实验，你是否能够使用Java语言解决一些有一定难度的编程问题？是否还有其他的感想和建议？ |
| 指导教师批阅意见：  成绩评定：  指导教师签字：蔡树彬  2024年11月2日 |