**内部类与异常**

**类实 验 报 告**

**学 院** 信息科学与技术学院

**专 业** 计算机科学与技术

**班级/学号**  b22011108

**学生 姓名**  陈龙

**成 绩**

**成绩评定评语：**

**实 验 地 点**

**实 验 日 期 2022.10.27\_**

**指 导 教 师 马**永恒

上海杉达学院

电子商务

**实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验名称 | 内部类与异常类 | 实验地点 | 线上 | 实验时间 | 2022.10.27 |
| 1. 实验目的：   熟练掌握内部类与异常类的编码方法； | | | | | |
| 1. 实验内容：     **教材p191：五、编程题** | | | | | |
| 1. 实验要求：   截屏  提供关键代码  a.在钉钉提交作业  b.源码放附在报告后  c.实验报告为word文档  d.上交截止时间为10.31日23:59:59 | | | | | |
| 1. 实验准备：  * 内部类概念知识； * 异常类概念知识和实现； * Dev\_Tools:IDEA * JDK\_Ver:14.0.1 | | | | | |
| 1. 实验过程：（含实验具体内容、时间、实验过程描述及截图、遇到的问题截图及描述、解决方法等，如果有代码则需要附**代码**）：   （1）教材p191：五、编程题  **实验时间：**  2022年10月27日 星期四  **实验内容及过程：**   * 分析题目需求，程序能对异常输入及时反馈，并向用户提供异常信息。 * Assertion: assert (成绩>= 0 && 成绩 <= 100): "非法输入！"; * Scanner Class: Scanner scanner = new Scanner(System.in); * 当输入“-2”，表示输入结束；   **遇到的问题和解决方法：**  **问题：**   * 无   **代码（核心代码即可）：**   * Main Application(源码见附页):     **收获和体会：**  主要学习了内部类和异常类。对于内部类，只有外嵌类才能使用，同时protect和private两个访问修饰符也不能使用。内部类也可以通过Lambda表达式使用，大大减少代码的复杂度，提高可读性。  对于异常可以通过Assertion断言来调试，能够快速定位到问题所在。 | | | | | |
| 1. 实验总结（总结该实验的总体体会和收获，还可以对这部分内容的教学方法提出建议）：   在第七章的学习中，主要学习了内部类和异常，通过继承Exception，可以自定义异常，对于程序Bug的处理和用户非法输入等都可以应用。  TryCatch结构也是常用的，尤其在数据库获取，或者硬件链接等，都十分有用。我之前在做TCP开发的时候，就会遇到没有使用TryCatch结构，在网络波动时，出现获取不到数据，线程卡死，最终导致程序无响应。TryCatch配合Exception不仅能帮助程序员定位问题、分析问题，还能提高程序的健壮性。  本次学习收获颇丰，逐步深入Java学习。 | | | | | |

说明：

1. **实验名称、实验目的、实验内容、实验要求**由教师确定，实验前由教师事先填好，然后作为实验报告模版供学生使用；
2. **实验准备**由学生在实验或上机之前填写，教师应该在实验前检查；
3. **实验过程**由学生记录实验的过程，包括操作过程、遇到哪些问题以及如何解决等；
4. **实验总结**由学生在实验后填写，总结本次实验的收获、未解决的问题以及体会和建议等；
5. 源程序、代码、具体语句等，若表格空间不足时可作为**附录**另外附页。

**附页：**

源码如下：

1. package HM\_2022\_10\_27;
2. import java.util.\*;
3. public class Application {
4. public static void main(String[] args){
5. int score = 0, sum = 0, mount = 0;              *//Variable to make the result*
6. Scanner score\_scanner = new Scanner(System.in); *//instance of Scanner to get the input*
7. while(!score\_scanner.hasNext("-2")){    *//pattern:-2, is the end of Input*
8. score = score\_scanner.nextInt();          *//to assignment score*
9. assert (score >= 0 && score <= 100): "非法输入！";   *//Assertion : when the score is invalid*
10. sum +=score;                              *//Make the count*
11. ++mount;                                  *// amount of score*
12. }
13. System.out.println("当前总成绩为："+sum);
14. System.out.println("当前平均成绩为："+sum/mount);
15. }
16. }