实验课程名称：Java语言程序设计A

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **实验项目名称** | **实验2：控制流程** | | | **实验成绩** |  |
| **实 验 者** |  | **专业班级** | **信管1502** | **组 别** | **1人1组** |
| **同 组 者** | **无** | | | **开始日期** | **2018年3月22日** |
| 第一部分：实验预习报告（包括实验目的及意义，实验基本原理与方法，主要仪器设备及耗材，实验内容及要求，实验方案与技术路线等）  一．实验目的及意义  控制流程（选择、循环）的应用。  二．实验基本原理与方法  1．选择。  2．循环。  三．主要仪器设备及耗材  1．PC及其兼容机。  2．计算机操作系统。  3．程序编辑器EditPlus。  4．Java开发工具JDK。  四．实验内容及要求  实验2-1：近些年，国际非常流行BMI（Body Mass Index，身体质量指数），即体重（单位：公斤）除以身高的平方（单位：米）。  世界卫生组织WHO制定的标准是：  BMI低于18.5，偏轻；  18.5~24.9，正常；  25~29.9，偏重；  30~34.9，超重；  35及以上，重度肥胖。  BMI越大，患高血压、糖尿病的风险越高。  请编写程序，输入你的体重与身高，计算你的BMI，对自身体重进行评估（偏轻/正常/偏重/超重/重度肥胖）。  实验2-2：将作业2-1、2-2、2-3综合起来，从键盘读取一个考试分数列表，统计：总考试人数、最高分、最低分、平均分、每个级别的人数、每个级别的人数占总考试人数的百分比。  实验报告包含设计思路、算法、注意事项、完整代码、运行结果、问题及解决过程、综合分析、总结、体会。  五．实验方案及技术路线（设计思路、算法、注意事项）  2-1实验方案与技术路线：  定义一个方法，名为BMI（）{  这个方法直接在方法体内接收我的体重、身高值；  同时在方法内输出我的BMI值；  判断我的超重指标；  }  2-2实验方案与技术路线：   1. main（）{   执行方法inputscores（）；  执行方法outputscores（）；  }  实例化ArrayList类并提供相应的引用变量；   1. 定义inputscores（）{   从键盘获取一个输入的分数；  ArrayList.add（分数）；  从键盘获取一个正误字符（Y/N）；  当为Y时，递归inputscores（）；  当为N时，方法结束；  }   1. 定义outputscores（）{   ArrayList.length（）；  遍历ArrayList，累加总分；  遍历ArrayList，判断分数段；  输出各种题目要求的数据；  } | | | | | |
| 第二部分：实验过程记录（可加页）（代码、运行结果、实验中出现的问题及解决过程）  实验2-1：计算我的BMI    小程序运行结果    实验2-2：考试分数列表  **import** java.lang.reflect.Array;  **import** java.text.NumberFormat;  **import** java.util.\*;  **public** **class** score {  **static** ArrayList<Integer> *score* = **new** ArrayList();//存放分数的数组  **static** **int** *inputscore*;//此变量接管输入流分数数据  **static** Scanner *keyboard* = **new** Scanner(System.***in***);  **public** **static** **void** main(String[] args) {  *inputscores*();  *outputscores*();  }  **static** **void** inputscores() {  //本方法进行分数的录入  String yesorno = **new** String();  System.***out***.println("请输入学生的成绩：");  *inputscore* = *keyboard*.nextInt();  *score*.add(*inputscore*);  System.***out***.println("还要录入吗？（Y/N）");  yesorno = *keyboard*.next();  **if**(yesorno.equals("Y")||yesorno.equals("y")) {  *inputscores*();//当需要多次录入，本方法将递归使用  }  }  **static** **void** outputscores() {  //本方法依照要求（找最高最低分，分数划段，百分比）进行输出分数的操作  **double** average = 0;//平均分  **int** highest;//最高分  **int** thelast;//最低分  **int** total;//总人数  **int** excellent = 0;//优  **int** good = 0;//良  **int** middle = 0;//中  **int** passed = 0;//刚及格  **int** failed = 0;//不及格  //获取百分数格式化对象（nt为引用变量）  NumberFormat nt = NumberFormat.*getPercentInstance*();  //设置百分数精确度2即保留两位小数  nt.setMinimumFractionDigits(2);  total = *score*.size();  highest = *score*.get(0);  thelast = *score*.get(0);    **for**(**int** i = 0;i < total;i++) {  //进行一次for循环比对出最高分和最低分，同时累加分数  **if**(*score*.get(i) > highest) {  highest = *score*.get(i);  }  **if**(*score*.get(i) < thelast) {  thelast = *score*.get(i);  }  average += *score*.get(i);//计算总分  }  average = average / total;//计算得到分数平均数  **for**(**int** j = 0;j < total;j++) {  //第二次for循环将分数分段，统计优良中差分数段人数  **if**(*score*.get(j) <= 100 && *score*.get(j) >= 90) {  excellent += 1;  }  **else** **if**(*score*.get(j) >= 80 && *score*.get(j) <= 89) {  good += 1;  }  **else** **if**(*score*.get(j) >= 70 && *score*.get(j) <=79) {  middle += 1;  }  **else** **if**(*score*.get(j) >= 60 && *score*.get(j) <= 69) {  passed += 1;  }  **else** {  failed += 1;  }  }    System.***out***.println("考试总人数为：" + total);  System.***out***.println("最高分为：" + highest);  System.***out***.println("最低分为：" + thelast);  System.***out***.println("平均分为：" + String.*format*("%.2f", average));  System.***out***.println("优秀人数为：" + excellent + "占总人数百分比：" + nt.format((**double**)excellent/total));  System.***out***.println("良好人数为：" + good + "占总人数百分比：" + nt.format((**double**)good/total));  System.***out***.println("中级人数为：" + middle + "占总人数百分比：" + nt.format((**double**)middle/total));  System.***out***.println("刚好及格人数为：" + passed + "占总人数百分比：" + nt.format((**double**)passed/total));  System.***out***.println("不及格人数为：" + failed + "占总人数百分比：" + nt.format((**double**)failed/total));  }  }  2-2小程序运行结果： | | | | | |
| 第三部分 结果与讨论（可加页）   1. 实验结果分析   ①计算数据和结果有小数点时，应该使用double类型数据存储，如果使用float型数据，那么在输入时需要再数字后用f标识，从用户角度看显然不符合输入要求。  ②实验二中有一个百分比输出需要留意，仍然使用（import）封装好的数字类型数据标准化NumberFormat类，实例化后运用引用变量调整好输出格式（预留小数点后几位），使用相应的方法输出即可（此时输出结果是字符串类型）。  ③当我们不知道输入的数字集合有多少元素时，装载并使用ArrayList是很好的选择。在声明int类型ArrayList时，参数为Integer。   1. 小结、建议及体会   这次实验学习了arraylsit的使用方式，在使用前需要装载ArrayList类，也就是原本在java中已经封装好。需要使用array对象的add方法，这样的好处在于这个“数组”可以不定长，非常适合不确定数量的元素录入。  第四部分 实验评分标准（教师自行设计）及成绩   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | 观测点 | 考核目标 | 权重 | 得分 | | 实验预习 | 程序设计方案、技术路线（设计思路、算法、注意事项）科学、合理、可行 | 实验方案设计能力 | 20% |  | | 实验过程 | 1. 按时参加实验，不迟到，不早退 2. 源代码正确、完整、可读性强、编码规范，运行结果正确，实验中出现的问题得到解决 | 学生的实验态度、UML面向对象建模与设计能力、分析并解决问题的专业素养及编程技能、良好的编程风格与习惯 | 50% |  | | 结果分析 | 实验结果分析到位、总结完善、体会深刻 | 学生对实验数据处理与分析的能力；对专业知识的综合应用能力；事实求实的精神 | 30% |  | | 附加分 | 正确、合理地完成选做实验内容 | 克服困难、刻苦钻研的顽强精神与严谨求实的科学作风 | 最高可加20% |  | | 该项实验报告最终得分 | | |  | |   教师签名： | | | | | |