软件学院实验报告

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 学 号 | 542313460109 | | 姓 名 | 胡华吉 | 专业班级 | 软件工程23-01 |
| 课 程 | Python程序设计 | | 实验日期 | 2024.11.4 | 实验时间 | 第四大节 |
| 实验情况 | | | | | | |
| **实验六：面向对象程序设计**  **一、实验目的**  （1）掌握定义带数据域和方法的类；  （2）掌握使用构造方法初始化数据域以构建一个对象；  （3）掌握使用圆点运算符(.)访问对象成员；  （4）通过选做任务，让学生针对一元三次方程，利用面向对象程序设计思想，利用二分法或牛顿逼近法，求方程的根，并能够分析算法的复杂度。  （5）培养学生科学计算能力和算法设计方面的创新思维能力，以及抽象逻辑思维能力。  **二、课程目标**  （1）能够在软件开发过程中，针对特定需求，综合数据结构、算法复杂性分析、Python语言等知识设计实现相关算法，解决实际问题。能够独立设计相关算法，使用Python开发软件系统。  （2）培养学生跟踪行业需求和技术发展的意识，培养计算思维方式、团队协作能力和自主学习能力，具有软件工程师职业道德。  **三、实验任务**  1、已有若干个学生数据，这些数据包括学号，姓名，Python程序设计成绩，高等数学成绩和大数据技术成绩，要求定义学生类，并用其成员函数修改各门课程的分数（要求将方法装饰成属性）、求各门课程的平均成绩。  ‬2、设计一个名为SolveEquation类来计算方程式ax2+bx+c=0的平方根。该类包括：  私有数据域a、b和c表示3个系数；  以a、b和c为参数的构造方法；  名为getDiscriminant()的方法返回判别式，即b2-4ac  名为getRoot1()和getRoot2()的方法使用下面这些公式返回方程式的两个根：  3、**高阶拓展**(选做)。基于面向对象程序设计思想，利用二分法或牛顿逼近法求x3+3x2+x+1=0的近似根，并分析算法的复杂度。  **四、实验要求**  1. 对每个任务求解步骤是：算法思想、程序代码、运行结果的截图。  2. 提交实验报告。  **五、实验设计及结果**  **请在程序代码中通过注释加入自己的学号和姓名，将代码和运行结果截图放在此处。注意排版，保证图中字体能够看清。**   1. **：**   **算法思想：面向对象编程**  **实现代码：**  结果：    2）  算法思想：暴力法  实现代码：    运行结果：    3）  算法思想：数学法（二分法）  **二分法**：每次迭代将区间减半，复杂度为 O（log（1/precision))，即取决于所需精度的对数。  实现代码：  运行结果：     1. **思考总结（本次实验的经验、教训，遇到的问题及解决方法，待解决的问题等）**  经验总结 **面向对象编程的灵活性**：在任务一和任务二中，使用类来组织学生和方程对象，不仅提高了代码的可读性和扩展性，还便于复用和管理。在学生成绩管理中，使用 @property 装饰器可以将属性封装，便于操作和控制输入数据的范围，增强了代码的安全性。  **数值方法的应用场景**：任务三中，通过二分法求解非线性方程让我理解了数值方法在没有解析解的情况下的重要性。二分法是一种简单且可靠的求根方法，适用于特定区间内确定有根的情况，使用该方法可以更有效地逼近解。  **计算的稳定性和效率**：在使用二分法时，控制精度对算法的效率有很大影响。精度越高，收敛速度越慢，需要更多迭代，因此需要权衡精度与效率。这种平衡在数值计算中非常重要。 遇到的问题及解决方法 **数据封装的理解**：最初在设计 Student 类时，对如何用 @property 进行数据封装不太清晰，后来通过查阅文档和实践，理解了属性装饰器的作用，能够更好地实现数据保护和简化代码。  **非线性方程求解的边界问题**：在任务三的二分法中，最初对边界条件设置不够严谨，导致算法在没有根的区间中陷入死循环。通过在代码中增加判断条件，确保区间端点的函数值异号，这样可以保证区间内存在根，从而提高了算法的鲁棒性。  **二次方程判别式判断**：在任务二中，遇到判别式为负数的情况，最初没有处理该情形，导致返回的值无效。为解决这一问题，我在 getRoot1 和 getRoot2 中增加了判断条件，使得当判别式小于零时返回 None 表示没有实数根。 待解决的问题 **数值方法的更优解**：虽然二分法有效，但在某些情况下效率较低。未来可以尝试使用牛顿法等更高效的数值方法来解决类似问题，并对比它们的收敛速度和计算效率。  **代码复用性和模块化改进**：可以进一步优化代码结构，将公用的计算逻辑提取为单独的函数或类，以提高代码的复用性。  **错误处理和异常捕获**：在处理一些不合理的输入（如非法字符或非数值类型）时，程序可能会崩溃。可以考虑加入更多的异常处理逻辑，增强程序的健壮性。  总体而言，本次实验让我加深了对面向对象编程和数值算法的理解，也培养了发现问题和解决问题的能力。在未来的学习和开发中，我会继续改进代码结构，提高效率。 | | | | | | |
| 实验报告成绩 | |  | | 指导老师 |  | |

注：1）专业班级按打印课表中名称填写；2）课程名称按课表中名称填写，不能简写；

3）实验日期格式示例：2024.03.09;4）实验时间格式示例：“第三大节”5）实验报告成绩按百分制评分。