**“非数值计算” 教学设计**

|  |  |  |  |
| --- | --- | --- | --- |
| **课程标准与教学目标** | | | |
| 适应的课程标准：1.7 掌握一种程序设计语言的基本知识，使用程序设计语言实现简单算法。通过解决实际问题，体验程序设计的基本流程，感受算法的效率，掌握程序调试与运行的方法。 | | | |
| 所针对的教材内容：4.3 非数值计算 | | | |
| 教室环境：有教学控制软件的多媒体机房 | | | |
| 预计课时：1课时 | | | |
| 教学目标：  1、运用合适的算法形成解决问题的方案。  2、了解算法设计中的分治思想，并运用二分查找解决实际问题。  3、体验递归算法，并结合具体问题开展编程实践。 | | 所指向的核心素养：  信息意识：对日常生活中的问题进行分析、判断，学会运用合适的算法解决问题。  计算思维：运用基本算法设计解决问题的方案，能使用编程语言实现这一方案。 | |
| 教学重点：  理解二分思想、递归思想，运用二分算法解决实际问题。 | | 教学难点：  理解递归算法及工作流程。 | |
| **评价设计** | | | |
| 过程性评价：   * 观察学生是否认真参与游戏、积极思考。 * 实践探究：观察任务完成情况，及时指导 * 反馈调整：通过以完成的任务展示，了解学生的掌握情况。 | | 终结性评价：   * 完成任务中代码的理解与补充。重点评估对两种算法思想的理解与应用实践。 | |
| **学习活动设计（1课时）** | | | |
| 教学环节 | 教学过程 | | 设计意图 |
| 引入 | **小游戏：猜数字**  老师挑一个100以内的数字，请同学们猜数字，看谁猜的快。  问题分析：要猜0-99之间的一个数，那么最好的方法就是从0-99的中间数49开始猜。如果要猜的数小于49，就猜24（0-48的中间数）；如果要猜的数大于49，就猜74（50-99的中间数）。重复这个过程来缩小猜测的范围，直到猜出正确的数字。 | | 通过生活中的小游戏引入，激发学生学习兴趣的同时，对二分查找的思想有更好的理解。 |
| 知识讲解 | **分治策略：**  分治的设计思想，是将一个难以直接解决的大问题，分割成一些较小的同类问题，各个击破，最终达到解决问题的目的。  **二分查找：**  分治策略的一种典型应用。猜数字小游戏用到的方法称之为二分查找，是一种在有序数组中查找某一特定元素的查找算法。查找过程从数组的中间元素开始，如果中间元素正好是要查找的元素，则搜索过程结束；如果某一特定元素大于或者小于中间元素，则在数组大于或小于中间元素的那一半中查找，而且跟开始一样从中间元素开始比较。如果在某一步骤数组为空，则代表找不到。这种搜索算法每一次比较都使搜索范围缩小一半。 | | 通过完善程序代码更好的理解二分查找的算法思想，并能很好的运用的编程实践中。 |
| 学生实践：任务一 | **任务一：**尝试完善二分查找程序  x = int(input(**"请输入要查找的数据："**)) step = 0 *#记录查找次数* flag1 = 1 *#目标区域左边界* flag2 = 1000 *#目标区域右边界* **while \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_***#区间数据范围小于1则结束循环* mid = \_\_\_\_\_\_\_\_\_\_\_\_\_\_ *#中间值* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *#查找次数加1* **if** mid>x:   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *#右边界前移* **elif** mid<x:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *#左边界后移* **else**:  **break** *#恰好找到目标数据，推出循环* print(**"查找次数为："**,step) *#输出次数*  思考：如果输入的数据不在范围内，会出现什么结果呢？ | |
| 体验小游戏，体会算法思想 | **小游戏：汉诺塔**  体验小游戏：*汉诺塔*  思考：怎样正确而快速的移动木盘？  **递归：**  举个从小就听过的例子：从前有座山，山里有座庙，庙里有个和尚，和尚在讲故事，从前有座山，山里有座庙，庙里有个和尚，和尚在讲故事，从前有座山...  **直接或间接地调用自身的方法称为递归**  是计算科学领域中一种重要的计算思维模式。既是一种抽象表达的手段，也是一种问题求解的重要方法。  递推关系是递归的重要组成，而边界条件是递归的另一要素，对递归而言，递推与回归，二者缺一不可。 | | 让学生通过体验汉诺塔小游戏，更好的理解递归的算法思想 |
| 学生实践：任务二 | **任务二：完善汉诺塔小程序，体验递归思想**  算法分析：    代码实现：  **def** hanoi(n,s,m,t): *#定义一个函数，将n个木盘从s借助m移动到t*  **if** n==1: *#当只有一个木盘时，直接从起始杆移动到目标杆*  print(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)  **else**:   hanoi(\_\_\_\_\_\_\_ , s,\_\_\_\_,\_\_\_\_\_\_)  *#将前n-1个木盘借助t从s移到m*  print(\_\_\_\_\_\_\_\_\_) *#将最下面的木盘从s移到t*  hanoi(\_\_\_\_\_\_,\_\_\_\_\_\_,\_\_\_\_\_,t)  n=int(input(**"请输入木盘的个数："**)) hanoi(n,**'A'**,**'B'**,**'C'**) *#调用函数，将n个木盘从A借助B移动到C*  思考：修改代码，计算汉诺塔游戏移动的次数 | | 通过完善汉诺塔的程序代码更好的理解递归的算法思想，并能很好的运用的编程实践中。 |
| 知识拓展 | **迭代与递归的关系**  从概念上讲，递归就是指程序调用自身的编程思想，即一个函数调用本身；迭代是利用已知的变量值，根据递推公式不断演进得到变量新值得编程思想。简单地说，递归是重复调用函数自身实现循环。迭代是函数内某段代码实现循环，而迭代与普通循环的区别是：循环代码中参与运算的变量同时是保存结果的变量，当前保存的结果作为下一次循环计算的初始值。  以斐波那契数列为例，体验递归与迭代的不同： | | 对比两个概念，便于学生更好的理解算法思想 |
| 小结 | 本节课通过两个任务来体验分治和递归两种算法思想，同学们可以通过体验进一步了解这两种算法思想，并将其运用到实际编程实践中。 | | 总结归纳，升华知识。 |
| **对差异化教学的考虑** | | | |
| 本课堂以两个任务为主，了解了算法思想之后，完善程序是基本任务，基本任务完成之后会有一个思考，是对现有程序代码的完善，这些可以留给学有余力的学生来尝试完成。 | | | |
| **教学设计思路** | | | |
| 教学内容分析：  本节内容是《必修1数据与计算》第4单元“计算与问题解决”第三小节的内容。本单元对“计算”进行了更多的探究。进一步说明了计算与算法的关系，明确算法的基本特征；讲解了数值类及非数值类问题求解中的常用算法思想，并尝试将这些知识运用到实际问题的解决中。本小节内容在4.2数值计算之后，继续探讨非数值的计算。计算是一个广泛的领域，数据是普遍存在的，对数据的分析、处理都属于计算的范畴。本节主要学习如何运用合适的算法形成解决问题的方案，并介绍了分治思想，二分查找和递归的算法思想，需要学生能够体会算法思想，结合具体问题开展编程实践。  本堂课以两个小游戏为例，让学生了解算法设计中的分治思想，并运用二分查找解决实际问题并体验递归算法。课堂开始以“猜数字”小游戏引入，一方面以游戏的形式激发学生学习的兴趣，一方面将分治思想渗入其中，让学生更好的理解二分查找的算法。接下来通过具体的实践来进一步巩固学生对二分查找算法的实践应用。  递归是本堂课重点及难点，关键在于学生对递归思想的理解。教学中以经典的小游戏“汉诺塔“为例，通过对汉诺塔问题的分析及求解过程，让学生体验递归的思想。 | | | |
| **针对核心素养培养的设计考虑** | | | |
| 本课对学生信息素养的培养：  信息意识：针对学生的认知水平及认知习惯，为学生设计了可以激发学习兴趣小游戏，使学生在游戏的同时产生较强的学习动机，能尝试运用合适的算法形成解决问题的方案  计算思维：通过小游戏，让学生尝试完善代码，并用编程的方法解决问题，在不断完善程序的过程中习得知识、发展思维、提升能力。 | | | |