

《面向对象程序设计》教学方法的探索与实践

张 力, 慕晓冬, 范青刚

西安高新技术研究所, 陕西 西安
Email: qingyi922@sina.com

收稿日期: 2021年4月7日; 录用日期: 2021年6月1日; 发布日期: 2021年6月8日

摘 要

《面向对象程序设计》课程中引入了面向对象思想和理论, 对于初学者来说, 理解和掌握类和对象的概念并能使用它们编写程序既是难点也是重点。笔者针对本门课程的特点, 根据多年的教学实践, 探索总结了几种行之有效的教学方法。

关键词

面向对象, 程序设计, 对象, 类

Exploration and Practice of Teaching Methods for *Object-Oriented Programming*

Li Zhang, Xiaodong Mu, Qinggang Fan

Xi'an High-Tech Research Institution, Xi'an Shaanxi
Email: qingyi922@sina.com

Received: Apr. 7th, 2021; accepted: Jun. 1st, 2021; published: Jun. 8th, 2021

Abstract

Because of the introduction of object-oriented thought in the *Object-Oriented Programming* course, it is difficult and important for beginners to understand and grasp the concept of class and object and use them to program. The author explores and summarizes several effective teaching

methods according to years of teaching practice and characters of this course.

Keywords

Object-Oriented, Programming, Object, Class

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

《面向对象程序设计》这门课程，是针对本科生开设的一门专业基础课，该课程对于学生理解面向对象的思想，掌握面向对象分析与程序设计方法，具备面向对象程序设计的基本技能具有重要的作用。

在本课程的教学过程中，需要始终围绕一条主线进行，那就是：让学生掌握面向对象程序设计理论的同时，能够运用面向对象的思想编写程序，并在思维能力、创新能力等方面得到提高。为此，在本课程的教学过程中，需要根据课程特点，采用合适的、灵活多样的教学方法以圆满完成教学任务。本文作者多年来一直从事这门课程的教学工作，对课程的内容有着较深的理解，并且积累了大量的教学经验，下面介绍一下本文作者从多年的教学工作中总结出的几种行之有效的教学方法。

2. 使用举例教学法讲解一些抽象难懂的概念

课程开始，学生首先碰到的就是面向对象、类等生涩的概念，教师首要的任务就是要将面向对象的概念植入学生的脑海中，使面向对象的思想在学生的脑海中扎下根。在这种情况下，举例教学法是非常必要的，采用举例教学法，深入浅出，使抽象的问题具体化，使学生很容易地理解面向对象、类等概念。例如，在讲授面向对象的思想时，举例：面向对象的思想就是把大千世界看成是由对象组成的，我们把客观世界中的所有的事物都看成是对象，石头、树，汽车、房子、动物、人，甚至表格都是对象，这样一讲，学生很容易就了解了面向对象的思想；又例如，在讲解对象的两个要素——属性和行为时，举了一个文具盒的例子，文具盒就是一个对象，它的属性就是它的长宽高，用来描述它的形状；它的行为是可以存放文具，取出文具，也就是对文具盒的操作；通过这个例子，学生就能轻易地理解掌握对象这个非常重要的概念。课堂实践证明，这是一种非常有效的方法，通过这种方法，可以使学生很快理解掌握对象、类等这些抽象难懂的概念。

3. 使用提问式教学法促进学生思考

课堂上为了启发学生不断思考，增强课堂注意力，增加互动，活跃课堂气氛，在讲授的过程中，需要不时地提问，让学生回答，通过这种方式，提高学生的学习的主动性、积极性和学习兴趣，激发学生的潜能，从而对所学知识印象深刻。例如，讲完一道例题后，可以提问：这道题是如何体现面向对象的编程思想的？这样可以启发学生的深层次思考，从而更好地掌握面向对象的理论。另外，也可以采用连环式的提问，层层剖析。同时，还可以列出一些学生易犯的错误，让他们分析思考，例如，给出下列类的定义，让学生找出其中的错误[1]：

```
class Sample
{
    int i=0;
```

```
public:
void Sample( );
~ Sample( );
~ Sample( int value ); }
```

先让学生们自己分析，然后再给出正确答案[1]：共有四个错误 1) int i=0；数据成员在类的定义中不能直接初始化；2) void Sample()；构造函数没有返回值类型；3) ~ Sample(int value)；析构函数无参数；4) 析构函数不能重载。

4. 随时注意培养树立学生的面向对象的思想，帮助学生掌握面向对象的程序设计方法

在讲解概念理论时，讲解例题时，都可以将面向对象的思想 and 面向对象的程序设计方法贯穿其中。比如，讲解一道例题时，可以引导学生从头开始思考，面对这样一个现实问题，如何用面向对象方法解决呢？然后，一步步讲解如何根据实际需要，抽象出类进行封装，如何建立对象，最后如何编写程序等等，这样循环往复，日积月累，增强了学生综合运用知识的能力，学生用面向对象的思想分析问题，解决问题的能力就逐步培养起来了。例如，在给学生讲授类、对象、属性、行为、封装和抽象等概念时，引入一个钟表的例子[2]，要求使用上述面向对象的概念编写一个钟表的应用程序，可以这样引导学生，要编写这样一个面向对象的程序，首先，要抽象出关于钟表的有用的属性，那么关于钟表的有用的属性是什么呢？很明显就是时、分、秒，那么就可以用时、分、秒设为私有数据成员；下一步，抽象出关于钟表的行为，根据现实中钟表的用途，钟表的行为有两个，一个是设置时间，一个是显示时间，这样就可以把这两个行为设为公有成员函数；属性和行为抽象出来以后，通过类的声明和封装，就建立起钟表类。代码如下[2]：

```
class Clock
{
public:
void SetTime( int NewH, int NewM, int NewS);
void ShowTime( );
private:
int Hour,Mintute,Second;
};
```

通过这个例子，帮助学生学会编写面向对象程序时如何下手，如何抽象出类。

5. 理论讲授和程序演示相结合，帮助学生更好地理解 and 掌握所学知识，提高学生的学习兴趣，活跃课堂气氛

在讲解完一些抽象的理论知识后，学生对于这些知识的理解还处于比较肤浅的水平，这个时候，必须将这些理论付诸实践，也就是运用这些理论编写程序，现场演示一下程序的运行，然后再通过运行讲解程序中所包含的理论和知识，这样会帮助学生深刻理解掌握所讲授的内容，也会极大地提高学生的学习兴趣 and 积极性，活跃课堂气氛。例如，在讲授了构造函数的概念 and 作用后，学生对构造函数的理解还不是很深刻，这时演示一个包含构造函数的程序[3]：

```
class A
{ private:
int a,b;
```

```
public:
    A()
    { a=b=0; cout<<a<<" "<<b<<endl; }
    A( int aa,int bb)
    { a=aa; b=bb; cout<<a<<" "<<b<<endl; }
};
void main( )
{ A x,y(2,3),z(4,5); }
```

这个程序中包含一个构造函数和一个默认的构造函数，首先让学生阅读程序，根据所学知识判断在什么情况下调用构造函数，在什么情况下调用默认构造函数，然后给出结果，学生给出的这个结果可能正确，也可能不正确，但从这个结果可以判断出学生在这部分知识的理解和掌握上存在什么问题；下一步就运行程序，显示出正确结果，再根据学生的问题，分析结果如何得出。这样一来，学生对于构造函数的作用就会有一个深刻的理解和掌握。另外，在演示程序的过程中，还可以人为地在程序中制造一些错误，然后再给学生展示如何调试程序，无形中也培养提高了学生的上机实践能力。

6. 抓好上机实践环节，逐步培养起学生的面向对象编程能力和调试程序的能力

在学生掌握了一定的面向对象的理论知识和程序方法后，就可以编写程序上机实践。通过上机实践加深对类、对象等知识的理解，提高编程能力和实际动手能力。通常，学生编写在书面上的程序，都会包含各种语法或者逻辑上的错误，需要上机运行检验，在反复修改这些错误，调试程序的过程中，学生可以不断积累经验，动手能力逐渐提高；通过上机实践，可以使得学生更好地掌握面向对象的理论和编程方法，提高学生的编程能力，调试程序的能力和解决问题的能力。开始时，实践的题目不易求多而要求精，每编一个程序，就要求学生通过这个程序更加深刻地理解课堂上所学的面向对象的理论概念以及如何运用这些理论概念编写程序。上机课上，要引导学生在程序出现错误时如何发现问题，解决问题，同时要求学生从程序运行失败中总结经验；这样循序渐进，逐步积累，动手能力切实得到提高。而且实践证明，通过上机实践，还可以极大提高学生对本门课程的学习兴趣。

7. 结语

《面向对象的程序设计》是学生既感兴趣又觉得难以掌握的一门课程，难点就在于利用类与对象编写程序。针对这个难点，在教学过程中，需要采取灵活多样、循序渐进、理论讲授和上机实践相结合的教学方法。本文作者根据多年教学实践总结出的上述几种方法在教学中取得了良好的教学效果，激发了学生学习这门课的积极性 and 兴趣，面向对象的编程能力得到了极大的提高。

参考文献

- [1] 王学颖, 黄淑伟, 李晖, 张燕丽, 赵娇洁. C++程序设计基础教程[M]. 北京: 清华大学出版社, 2019.
- [2] 郑莉, 董渊, 何江舟. C++语言程序设计[M]. 北京: 清华大学出版社, 2017.
- [3] 徐孝凯. C++语言程序设计[M]. 北京: 清华大学出版社, 2006.