# 基于计算思维的C语言教学案例设计

魏书堤,赵辉煌,邓红卫
(衡阳师范学院 计算机科学系,湖南 衡阳 421008)

**摘　要:**C语言是大学理工类必修的计算机语言类课程,也是一门实践性很强的课程,针对目前很多学生虽然掌握了C语言的语法规则,但由于缺少计算机思维的训练,仍无法利用C语言来解决一些实际问题的现状,提出一种基于计算思维的C语言教学方法。通过设计案例,详细阐述了基于计算思维的C语言教学具体过程。

**关键词:**案例设计;计算机思维;C语言
**中图分类号:**C934　　**文献标志码:**A　　**文章编号:**1674-5884(2014)03-0108-03

C语言介于高级语言与低级语言之间[1],是大学理工类必修的计算机语言类课程,也是数据结构等课程的前趋课程。由于C语言涉及的概念规则很多,且使用形式灵活,很容易出错。教学实践表明:初学者对教材前面的语句语法、变量表的学习还能跟得上,但一旦讲解比较复杂的章节时就困难重重,甚至有的学生学完了C语言,考试成绩也很好,但是让其用C语言去解决一个实际问题时,就无从下手,没有清晰的思路和合理的解决方案。为了使学生能更好地掌握C语言解决一些实际问题,教师要从实际生活中去挖掘一些较好的案例,对教学中的问题进行分析并将教学的案例设计与计算思维培养结合起来,以提高C语言的教学质量。

## 1　当前C语言教学中存在的问题

### 1.1　课时减少,教学内容未整合优化

随着各专业人才培养方案的修改,作为公共基础课程的C语言教学课时被压缩,很多任课教师未能对教学内容进行整合优化,上课时仍然按部就班讲解C语言,比较复杂的教学内容讲解不透,没有结合生活工作实际来设计合理的案例,把知识融入到案例中去,忽视了学生计算思维的培养,而计算思维恰恰又是对问题抽象的基础。

### 1.2　实践教学存在弊端,使得教学效果不理想

根据我们对学生的调查与了解,很多老师布置的课外作业都流于书本之上,很少根据自己的教学设计,有针对性布置一些思维性强的课外作业,这样学生一上机实验都在验证一些书上已经有的程序,而且象这些程序早已经过教材编写者调试,学生在调试时很少出现一些意想不到的错误,很难分析错误产生的原因,在解决实际问题时,很难进行战术方面的思维,也就是说有了抽象方法以后,也难有成功的程序。

## 2　案例设计与计算思维相结合

老师在进行教学案例设计时,不仅要把握知识的易理解性,而且要把握思维规律,渐进式地演绎分析实际问题,找出其中内在规律,抽象出基础模型算法,讲解时可以用形象类比等方法进行启发性教学。

### 2.1　精心组织教学内容,合理设计教学案例

老师应认真分析教材和学生的实际情况,精心组织教学内容,设计合理的教学案例来实施教学。在实施教学过程中,应从实际问题引入教学,通过对实际问题的抽象来启发学生的思维,通过问题-算法-程序这一系列的过渡对实际问题进行解决,从而达到对知识理论的掌握和运用。学生在学习C语言程序设计时,不外乎2个方面的学习,一方面是对C语言课程知识的理解,另一方面是专业知识的灵活运用,所以老师们的教学重点应放在对学生思维能力的培养和思维习惯的养成上。

C语言实际上是一门实践性非常强的课程,老师在进行案例设计时,要坚持以培养学生的学习理解能力、计算思维能力和创新能力为目的[2],案例内容要能有利于激发学生的学习兴趣,能引导学生积极进行多种抽象思维并最终解决实际问题。老师在课堂微观教学上采用案例递进驱动教学法[3],改变满堂灌的做法,充分调动学生的积极性,活跃学生的计算机思维。精心设计实例,给学生一个比较实际的切入点,通过老师的讲解和演示使学生有直观感觉和理性思维,然后再通过将此实例不断修改、扩充,引导学生参与到程序的编制过程中。在这个过程

**收稿日期:**2013-10-22
**基金项目:**2012年湖南省普通高等学校教学改革研究项目(338);衡阳师范学院教学改革研究项目(JY201105);2013年湖南省普通高等学校教学改革研究项目(333)
**作者简介:**魏书堤(1969-),男,湖南衡阳人,高级工程师,主要从事管理与决策研究。

中,学生展示所编制的程序,老师评判优劣并讲解理由和规律,吸收优点,修改错误,引导学生进行优化。在这样的案例教学中,因为有提出问题、解决问题、扩展问题、再解决问题、对解决问题的方法评价、优化设计等几个环节,实际上是一个螺旋式滚动向前的过程。在这个螺旋式不断向前的过程中,能够很好地调动学生的参与,而且通过问题的不断扩展和一个问题多种解决方法,能有效拓展学生的计算思维,使得学生在课堂上真正成为"主体",教师只扮演"主导"角色[4-5]。通过老师的讲解调试和演示,使学生有直观的感觉,从而引导学生的思维与老师教学达成一致产生共鸣,达到理想的教学效果。这样通过一个程序实例,引入课程内容,使得学生的每一步学习都有基础,是循序渐进,螺旋式上升的过程。

### 2.2　遵循循序渐进规律,精心设计实验项目

在设计实验项目的过程中,要采取循序渐进思路。首先要让学生做最基础的理解性实验,通过这种理解性的实验来理解课堂上的理论知识;然后要通过验证实验,把书上例题和老师讲解的例题进行系统对比验证,从而达到理解和掌握程序设计的关键步骤,达到自己可以灵活设计习题程序的目的;最后设计一个把过程设计和算法设计整合在一起的实验,逐渐地提高学生实践和应用能力,另外实验项目设计还要具有一定的趣味性。在实验设计案例时应将知识点融入进去,让学生在实践中锻炼自己的思维,在对问题思索中形成习惯和兴趣。

## 3　案例设计实例

设计程序打印输出如下螺旋方阵(见图 1)。

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 16 | 17 | 18 | 19 | 6 |
| 15 | 24 | 25 | 20 | 7 |
| 14 | 23 | 22 | 21 | 8 |
| 13 | 12 | 11 | 10 | 9 |

图 1　打印输出螺旋方阵

### 3.1　案例分析

由上例直观可知,以顺时针方向从外围开始递增的填充矩阵,每填充一个外围,问题即被分解为与此相同更小的问题,重复地从外围填充相应子矩阵,即可完成。

设原问题为 $N$ 阶的螺旋方阵,起始值为 $ns$,终止值为 $N*N$,则矩阵也为 $N$ 阶的矩阵,起始行为 $rows$,起始列为 $cols$.

递归开始:

第一步:从映射矩阵的第一行开始开始向右填充 $n$ 个螺旋方阵中的值(从传入的起始值开始,值递增变化)。

第二步:从映射矩阵的最后一列第二行开始向下填充 $n-1$ 个螺旋方阵中的值(值递增变化)。

第三步:从映射矩阵的最后一行最后一列开始向左填充 $n-1$ 个螺旋方阵中的值(值递增变化)。

第四步:从映射矩阵的倒数第二行第一列开始向左填充 $n-2$ 个螺旋方阵中的值(值递增变化)。

此时矩阵的最外围被填充完毕,原问题转化为 $n-2$ 阶的螺旋方阵,最小值为 $ns$(一直递增变化),最大值为 $N*N$,则映射矩阵也为 $n-2$ 阶的矩阵,起始行为 $rows+1$,起始列为 $cols+1$ 的子问题,递归地解决子问题,直到传入的螺旋矩阵起始值大于或者等于终止值时,递归调用结束,函数退出,问题解决。过程如下图 2 所示。

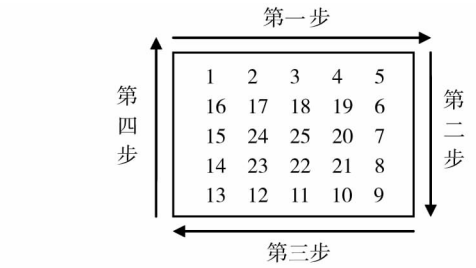


图 2　递归步骤图

### 3.2　实际数据运行

$n=4$ 的情况(初始值均赋位 0)

第一次遍历结果:

1 2 3 4

12 0 0 5

11 0 0 6

10 9 8 7

第二次遍历结果:

1 2 3 4

12 13 14 5

11 16 15 6

10 9 8 7

C 语言代码

```
#include < stdio. h >
#include < stdib. h >
// - - - - - - 参数说明 - - - - - -
// N:螺旋方阵的最大阶数
// n:子矩阵的阶数
// ns:递归遍历螺旋矩阵时,当前计数的值
// arr:矩阵
// rows:本次遍历,映射矩阵的开始行
// cols:本次遍历,映射矩阵的开始列
// 无返回值,递归遍历
// - - - - - - - - - - - - - - - - - - - - - -
Void SortArt( int N , int ns, int * * arr, int rows, int cols);
// - - - - - 参数说明 - - - - -
// n:n 阶矩阵的阶数
// arr:打印的 n 阶矩阵
// - - - - - - - - - - - - - - - - - - - - - -
Void PrintArr( int n,int * * arr);
Int main( int argc,char * argv[ ])
{
Int N; // 螺旋方阵的阶数
Int * * arr;// 矩阵
Printf("Input(N 必须大于 0) N = ");
Scanf("%d",&N);
arr = ( int * * ) malloc( N * sizeof( int * ));
```

```
for(int i=0;i<N,i++)
{arr[i]=(int *)malloc(N*sizeof(int *));}
//---为映射矩阵的存储分配空间---
for(int i=0;i<N;i++)
  for(int j=0;j<N;j++)
  arr[i][j]=0;
// 递归遍历螺旋方阵填充映射矩阵
SortArr(N,N,1,arr,0,0);
// 打印映射矩阵
PrintArr(N,arr);
System("pause");
Return 0;}
Void SortArr(int N,int n,int ns,int **arr,int rows,
int cols) // 围绕递归解法,从0,0开始
  {if(ns>N*N) return;
  int row=rows;
  int col=cols;
  int c=ns;
  col=col-1;
for(int i=0;i<n;i++) // 从子矩阵的第1行开始
向右遍历
  {row=row+1;
  arr[row][col]=c;
  if(c>=N*N)return; // 当前值大于螺旋矩阵的
最大值,则跳出
  c=c+1;}
for(int i=0;i<n-1;i++) //从子矩阵的最右列第
2行开始向下遍历
  {row=row+1;
  arr[row][col]=c;
  if(c>=N*N)return;
  c=c+1;}
for(int i=0;i<n-1;i++) //从子矩阵的最后1行
的右侧开始向左遍历
  {col=col+1;
  arr[row][col]=c;
  if(c>=N*N)return; //当前值大于螺旋矩阵的
最大值,则跳出
  c=c+1;}
for(int i=0;i<n-1;i++) // 从矩阵的最右列第2
行开始向下遍历
  {row=row+1;
  arr[row][col]=c;
  if(c>=N*N)return;
  c=c+1;}
for(int i=0;i<n-1;i++) // 从子矩阵的最后一
行的右侧开始向左遍历
  {col=col-1;
  arr[row][col]=c;
  if(c>=N*N)return;
  c=c+1;}
for(int i=0;i<n-2;i++) // 从子矩阵的最左列
从下向上遍历
  {Row=row-1;
  arr[row][col]=c;
  if(c>=N*N)return;
  c=c+1;}
PrintArr(N,arr); // 追踪映射数组的填充情况
SortArr(N,n-2,c,arr,row,col+1);
}
Void PrintArr(int n,int **arr) // 打印矩阵函数
  {Printf("**********\n");
  for(int i=0;i<n;i++)
{for(int j=0;j<n;j++)
  {Printf("%d",arr[i][j]);}
Printf("\n");}
printf("**********\n");}
```

### 3.3 案例总结

上面案例的图案非常有趣,教师可以引导学生进行抽象思维,很容易理解螺旋矩阵算法,代码实现将数组、函数、数组做函数参数、循环、条件语句、及递归有机地结合起来。在布置实验时,可以让学生进行反螺旋输出如下图案(见图3),这样可以让学生举一反三,提高对实际问题的解决能力。

| | | | |
|---|---|---|---|
| 10 | 11 | 12 | 13 |
| 9 | 2 | 3 | 14 |
| 8 | 1 | 4 | 15 |
| 7 | 6 | 5 | 16 |

图3 输出结果图

## 4 结 语

如何提高学生利用c语言解决实际问题的能力,使学生的计算思维得到有效的培养,是摆在我们大多数计算机C语言教学老师面前的一大难题。我们认为教师只有根据教材、课时及教学计划精心设计有趣的教学案例,将书本上的知识点,融于实例之间,并进行启发式教学,才能激发学生的学习兴趣,潜移默化地诱导学生进行思维训练,提高学生对实际问题的抽象能力。

## 参考文献:

[1] 谭浩强. C程序设计[M]. 北京:清华大学出版社,2001.
[2] 邱建林,王 波. 计算机程序设计语言教学的探索[J]. 牡丹江大学学报,2001(4):14-15.
[3] 胡 枫. C语言程序设计6的案例式教学的设计[J]. 青海师范大学学报(自然科学),2012(4):48-51.
[4] 高 红. 开设自主性实验 培养学生的创新意识和创新能力[J]. 实验技术与管理,2001(12):60-62.
[5] 耿国华. 程序设计能力培养模式的探索与实践[J]. 中国大学教学,2009(3):30-32.

(责任校对 晏小敏)