**以“画程”为桥，到达计算思维的彼岸**

【摘要】画程软件使得程序设计教学可以将更多精力放在算法的验证和实现过程上，缓冲了算法难理解和语法繁杂多等诸多问题的障碍，为高中生在程序和算法学习的起步阶段铺就了一条平缓的道路，有利于培养学生的计算思维能力。但算法的实现又不能仅仅停留在流程图上，而应该深入到代码中去落地。本文探讨了如何将画程软件作为工具和桥梁，借助程序流程图来逐步渗透程序算法设计的思想，引导学生从计算思维的角度来思考问题、解决问题的同时，也兼顾到算法到代码的最终实现。

【关键词】高中信息技术 画程软件 计算思维

如何从计算思维的角度来设计和组织教学，让学生领会学科核心素养，提升学生信息素养，是每一堂信息技术教学设计都应该思考的重中之重。核心素养中，计算思维占据着相当重要的位置，对学生的培养目标是能结合日常生活，初步运用基本算法，设计出解决问题的方案，乃至能使用编程语言或其他数字化工具实现这一方案。笔者所执教的是一所普通农村高中，学生普遍的学习起点是：面对情境时，不容易形成问题，更不知如何思考问题，从何处入手，也就谈不上将思维过程转换成程序代码。如果仅一味地“一板一眼”从代码处着手程序设计的教学，学生不单是学着累，就算是硬啃下教师示范过的例子，遇到稍微不一样的问题又会无从下手。久而久之，学生势必会产生学而无趣、学而无用的情绪。另一方面，如果完全地弃代码于不顾，只高谈算法本身，也会使得算法悬至在思想的云团里，得不到“眼见为实”的执行，而让信息技术的课堂变成一锅夹生饭。在对新教材的深入解读之下，笔者认为，教学重点不应该让程序设计的学习重点从程序设计的思想和方法，偏移到语言的语法规则上，忽略计算机求解问题的核心：程序的算法设计；也不能偏执在有算法却不能落地实现的空中楼阁上。在帮助学生建立计算思维，理解程序算法设计不仅是计算机解决问题的核心，更是程序设计的灵魂的同时，也要让学生体会到从算法变成代码并顺利实现的喜悦。为此，我们试着将画程软件作为工具和桥梁，借助程序流程图来逐步渗透程序算法设计的思想，引导学生从计算思维的角度来思考问题、解决问题，帮助学生掌握并实现程序设计的思想和方法以及它的代码实现。

**一、以 “画程”软件为桥，架起计算思维的通途**

画程软件是一种基于流程图的可视化编程开发环境， “ ‘画程’，寓意‘用画图（流程图）的方式画出程序来’。画程软件能把流程图自动翻译为程序代码（支持Python，Java，C，C++，VB，QB等六种计算机高级语言），能编译（或解释）运行实现算法，这样就使原来的算法教学的三步变成了两步：第一步，设计算法（用自然语言表示）；第二步，用画程软件画出流程图，而这个流程图是可以执行的。”①

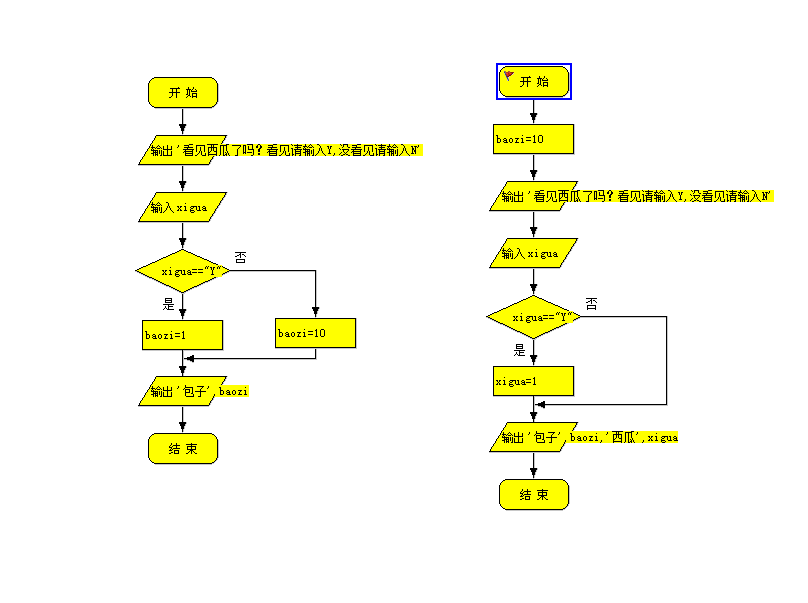
画程软件具有设计程序简单、直观的特性，让学生能够在学习程序设计时不仅学会程序设计的思想和方法，还可以体会计算机解决问题的过程与方法，从而影响了学生的思维方式和思维习惯，在这一过程中让程序或算法思想变得更加清晰。同时，利用画程软件制作的流程图可以编译运行实现算法这一优点，可以让学生在对代码并不深入了解的情况下，就能看到自己算法的执行结果。例如在学习分支结构时，我们设计了这样一个小问题:

【小笑话】翠花让当程序员的老公去买早餐：“去买一斤(10个)包子回来，如果看到卖西瓜的，买一个。”

不久，程序员老公手捧一个包子进了家门……

程序员有错吗？请你画一画他的算法流程。还有几种画法？哪种流程图求生欲更强烈？

在尚未学习到if语句的代码如何写的情况下，学生凭借画程软件很方便地画出了几种流程图，还原了程序员买早餐时的各种可能性，并能直观地看到输出的结果。下图为学生的作业中有代表性的两种画法。

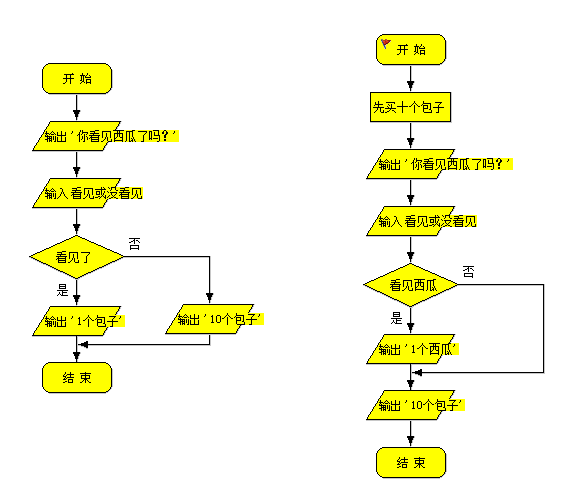


可以看出，学生已经建立了初步的思维能力，懂得了分支语句的关键点在于条件的判断。在此基础上，教师引导学生对他们自己画出来的不同流程图进行对比，让学生分别对不同算法设计进行分析，不仅让学生体会程序员可能存在的思维方式，而且也让学生在比较中思考：怎样的算法才能称得上更优的算法。课堂教学效果表明，将流程图、算法设计、程序设计融为一体不仅是一种综合的体验，也是培养学生计算思维的一个过程。

**二、以 “画程”软件为桥，沟通算法到代码的两岸**

画程软件虽然容易上手，但要用好画程，也不是一蹴而就的事情。

既然画程软件可以看成是自然语言和代码之间的桥梁，起点仍然要设在自然语言的描述上，对于初识算法的学生来说这是不可逾越的环节，它是思维过程的一种体现。能用自然语言把实际问题的算法描述清楚，才有可能进一步用画程软件的流程图来表达。仍以上述中的程序员买包子事件为例，在一开始，学生并不能熟练地引入变量，也不能实现结果的输出，他们的流程图最初是如下这样的：



这些流程图应该被称为“画程版”的自然语言表达，但好在，思想被无误地表达出来了，只剩“以更接近计算机程序化的语言来加工，以便让计算机输出结果”这件事。这便成了教学上最好的起点，利用学生急切想输出结果的心情，恰好是教师引入“变量”，“常量”，“赋值”，“关系式”等一系列晦涩难懂的概念的最好时机，在一个“早餐的购进和摆放”之中让自然语言巧妙地接近计算机的算法表达，直至学生们交出图1所示的流程图，实现程序的编译运行。学生在编译成功的刹那，体会到了成功的喜悦，对程序学习的接受度明显有所提高，也使得学习的趣味性有所增加。而到这一步，并不是学习的终点。

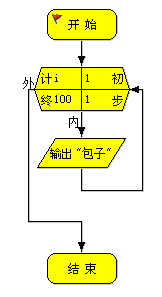
“探究算法教学，其实也就是三个基本步骤：第一步，从分析问题得到用纸和笔分步骤可实现的通用算法，可称为纸笔算法；第二步，把用自然语言描述的纸笔算法转为用流程图表达；第三步，把流程图翻译为程序代码实现算法。”②也就是说，用代码对算法的实现是从分析问题起始最终要到达的问题解决的彼岸，如果完全依赖于画程软件的流程图，仅仅靠读识画程软件上的代码，并不能让学生完全掌握代码，还会让算法的学习瘸了一条腿。实践证明，画程软件对代码转译存在一定的局限性，例如，它能够根据流程图和学生键入的信息自动生成程序代码，却不能改变代码区的现成代码，这使得代码的表达存在一定的局限性，这就需要学生对这些半成品式的代码进一步完善，而这也正是学生在代码学习上的最好入口。从教学实践来看，读懂一行代码和写出一行代码之间还有最后的一里路。而要走好这一里路，还需要恰如其分地利用画程软件。经过对画程软件自动生成的程序代码的反复识读，再经过对代码的加工修改，在相当程度上消解了学生在写代码时的陌生感，但要独立写出一行正确的代码，对学生来说是有挑战的：英文符号、代码语法等方面还是会错误频出。

为此我们挑选了一个简单却内涵丰富的循环语句为契机，让学生亲自编写一个程序员买100个包子（输出100个“包子”）的代码。

for i in range(100):

print(“包子”)

虽然只有短短的两行代码，却包含了关键字，函数名，标点符号，循环体的缩进等一系列的易错点。直到代码的调试成功，学生对代码的书写规范和再识读新的代码能力有了质的提高。在这个基础上，我们让学生反过来再用画程软件表达出这个语句。



特别简单的一个流程图，却让学生在计数变量，初值，终值，步长上参照着巩固了一遍，再进一步和自己编写代码中的range(100)做了一个比较，学生更确切地明白了range函数左闭右开的特性。在教学过程中，我们经常引导学生进行类似的代码与流程图之间的相互转换，相互印证，起到了一种螺旋上升的迭代学习效果。

**三、以 “画程”软件为桥，到达更远更精彩的彼岸**

随着学习的深入，一定会出现更复杂问题的解决。对于比较复杂的流程图，教师要给予一定的支架。根据不同的学习阶段和学习重点，在教学的过程中，教师可以提供画程文件的半成品，引导学生进行想象与思考，补齐缺少的内容。经常为大家所使用不外乎以下三种方式：

**1. 添加式**

所谓添加式，具体指教师已经搭建好程序的主框架，甚至是已经可以编译成功的流程图，然后由学生自主发挥创意，对教师的成品再行添补。例如猜数游戏，在教师提供的画程文件基础上，学生添加一些框图，改变游戏的对话或猜数的范围，从而改变游戏的有趣程序和难易度，这样既能让学生站在使用者的角度体验程序设计，也可以站在开发者的角度参与到小游戏的设计之中，虽然只有点滴的参与，却有着很好的引导作用。需要注意的是，留给学生添加的部分应该是与本节的重点与难点相关的部分，比如说猜数游戏中，学生添加的对话是针对输入输出语句的反复习得而设计的，而改变猜数的范围则是针对分支结构的多分支部分而设计的。总之，看似给了学生自由添加的基础，但老师给出的支架作品是对学生有一定引导性的，包含着有意识的设计，而不是随机的生成。

**2. 纠错式**

在实践中最常用的一种方式是纠错式。所谓纠错式，是指通过让学生在教师提供的“有误”的半成品中寻找与改正错误，以达到教学目的的一种教学方式。同添加式的作品一样，教师提供的半成品程序上的“陷阱”也应该具有代表性，除了与学习的重点与难点相关的知识点，更多的是学生中常常出现的错误点。例如画程软件的输入框是有三种数据类型（字符串类型、整型或浮点型）可选的，学生忘记选择正确的数据类型会引出编译的错误，这是一种典型的错误。更常见的错误还有输出字符串时引号的丢失等等。另外，用来纠错的作品不一定是教师提供的，更可以从学生提交上来的作品中选择典型案例，隐去学生的姓名后再发给全体学生去纠错，这比直接转播学生的错误要更有利于维护学生的自尊心，不让任何学生因为担心被同学嘲笑而不敢尝试。

**3. 改造式**

利用改造已有的成品或半成品也能取得非常不错的教学效果。所谓改造式，具体是指在现有的流程图基础上进行修改，使其难度有所增加，或者经过修改之后实现流程图的变异或优化，使其进入更高一阶的算法思维，从而达到对学生实施拓展训练的目的。例如在讲到循环结构的两种循环（计数循环和条件循环）时，我们就以“国王与大米”为案例，提出问题：

国际象棋一共64个格子，第一个格子放1粒米，第二个格子放2粒米，第三个格子放4粒米……棋盘上的总米粒数是多少？

第一阶段让学生画出计数循环的流程图。而第二阶段就在这个基础上改造流程图，让程序实现以2019年中国的粮食总产量，能放到棋盘的第几格的计算。这样的改变也正好应对了画程软件上两种不同循环结构的框图使用，让学生懂得这样的变式是基于循环体不变的异变，从而让学生掌握循环体，也便于学生通过计数循环和条件循环的对比，选择恰当的算法解决实际问题。

以上全部仅为我们一段不长时间的教学实践体会，对画程软件在计算机教学中的应用还存在许多盲区和不尽之处。但可以肯定的是，作为一种简单易懂的可视化编程软件，画程软件在教学过程中的使用，缓冲了语法繁杂多等诸多问题的障碍，为高中生在程序和算法学习的起步阶段铺就了一条平缓的道路，不仅增强了学生学习程序设计的兴趣，而且通过对问题的思考、讨论、交流，出示解决问题的流程图并运行结果，将程序设计中原本抽象的教学过程变得更加清晰，也使我们的教学有精力加强算法的设计和实现，培养学生的计算思维能力。 与此同时，教学目标又不能仅仅停留在流程图的完成上，而应该深入到代码中去落地，只有恰当地以画程软件作为工具和桥梁，才能更好的引导学生从计算思维的角度来思考问题、解决问题的同时，也兼顾到算法到代码的最终实现。

1. ②伍先军 《人人都能学会编程——用“画程”软件激活编程教学》《中国信息技术教育》 2018年06期