

## 【学科教学】

# 面向计算思维培养的编程教学研究<sup>\*</sup>

## ——以高中生 Python 编程教学为例

傅 骞 王钰茹

**【摘 要】**随着人工智能时代的到来,在编程教学中进行计算思维培养正在成为信息技术教育的关注重点。基于计算思维的问题解决过程一般包括问题界定、问题分解、模式匹配、数据抽象与表达、算法设计和评估反思等环节。面向计算思维培养的编程教学以项目学习和问题解决为中心,由创设情境、分析问题、设计方案、实施反馈、项目迁移和分享总结等环节组成;该流程强调通过叠加项目功能或加大问题复杂度来启发学生运用计算思维方法解决问题。基于此流程设计的《田忌赛马》教学案例旨在为教学提供参考,并推动计算思维和编程教育的相关研究。

**【关 键 词】**编程教育; 计算思维; 教学流程

**【作者简介】**傅骞,北京师范大学教育技术学院副教授(北京 100875);王钰茹,北京师范大学教育技术学院本科生(北京 100875)。

随着人工智能时代的到来,在编程教育中进行计算思维培养正在成为信息技术教育的关注重点。2017年7月,国务院印发了《新一代人工智能发展规划》,要求在中小学阶段设置人工智能相关课程,并逐步推广编程教育<sup>[1]</sup>。《普通高中信息技术课程标准(2017年版)》中将“计算思维”作为信息技术学科的核心素养之一,强调信息技术教育中应贯穿计算思维培养。

计算机程序设计语言众多,其中Python因具有简单易学、语法优美、扩展丰富等优点被广泛用于数据分析和人工智能等领域,非常适合作为中学生编程学习的入门语言,既能使学生快速入门,也有助于学生了解人工智能等前沿技术。

目前,国内已有不少高中开设了编程课程,但普遍存在以下问题。一是编程教学以语法算

法的教条学习为主<sup>[2]</sup>,学生容易感到枯燥乏味,逐渐失去编程兴趣和信心。二是教师往往预先设定解题路径,让学生根据步骤完成任务<sup>[3]</sup>,这使得学生缺乏独立探索、创新的机会,利用计算机解决实际问题的能力难以得到提升,对学生日后的成长意义不大。显然,这与当前所提倡的培养学生利用计算思维解决实际问题的能力有一定差距。因此,本文旨在研究如何在编程教学中培养学生的计算思维,帮助学生在实际生活和学习中更好地解决问题。

### 一、基于计算思维的问题解决过程

关于计算思维的定义和过程,国内外很多学者和机构都提出了不同的观点。2006年,周以真

<sup>\*</sup> 本文系中国科学院院士咨询项目“我国科学教育发展战略研究”(项目编号:2018-Z10-A-025)的研究成果。

把计算思维定义为:运用计算机科学的基础概念进行问题求解、系统设计以及人类行为理解等涵盖计算机科学之广度的一系列思维活动,包括抽象、分解、问题表达、建模、权衡评估等环节。<sup>[4]</sup>2011年,美国国际教育技术协会(International Society for Technology in Education, ISTE)和计算机科学教师协会(Computer Science Teachers Association, CSTA)给出了计算思维的操作性定义:计算思维是一种问题解决过程,包括将问题阐述为计算机可处理的形式、组织和分析数据、通过模型或模拟等方法抽象表达数据、设计算法、综合利用各种资源实施方案并评估方案、概括并将问题解决过程迁移到其他问题解决中。<sup>[5]</sup>2013年,南安普顿大学的Selby博士和Woollard博士提出计算思维包括算法思维、评估、分解、抽象、概括五个方面的要素<sup>[6]</sup>。宾夕法尼亚大学的Susan Davidson教授和Chris Murphy副教授在其在线课程“面向问题解决的计算思

维”(Computational Thinking for Problem Solving)中提出,计算思维是运用计算机科学领域的概念及方法,运用计算机解决实际问题的一系列思维过程,主要包括分解、模式匹配、数据表达和抽象以及算法设计。<sup>[7]</sup>

《普通高中信息技术课程标准(2017年版)》中将计算思维纳入信息技术课程核心素养中,并将计算思维定义为“运用计算机科学领域的思想方法,在形成问题解决方案的过程中产生的一系列思维活动”<sup>[8]6</sup>。其强调:“具备计算思维的学生,在信息活动中能够采用计算机可以处理的方式界定问题、抽象特征、建立结构模型、合理组织数据;通过判断、分析与综合各种信息资源,运用合理的算法形成解决问题的方案;总结利用计算机解决问题的过程与方法,并迁移到与之相关的其他问题解决中。”<sup>[8]6</sup>

表1展示了不同研究总结的计算思维核心过程。

表1 相关研究中的计算思维过程

计算思维过程	周以真	Selby & Woollard	ISTE & CSTA	Susan & Chris	《普通高中信息技术课程标准(2017年版)》
问题界定					√
分解	√	√		√	
模式匹配 / 概括		√	√	√	√
抽象	√	√	√	√	√
数据表达 / 组织	√		√	√	√
算法设计		√	√	√	√
建立模型	√				√
评估	√	√	√		
迁移			√		√

综上所述,本文最终采用的计算思维过程是问题界定、问题分解、模式匹配、数据抽象与表达、算法设计以及评估反思,该过程本质上就是利用计算思维解决问题的过程,包括问题分析、问题表达和问题解决三个阶段。在问题分析阶段,需要通过问题界定判断该问题能否利用计算机解决,接着通过问题分解、模式匹配等计算思维将问题化简,以找到问题解决的办法;在问题表达过程,首先需要通过抽象提取出问题中的关键要素、合理组织数据,再将问题重新表达为计算机可处理的形式;在问题解决过程,则需设计算法、编程实现,并通过评估反思不断迭代优化

问题解决方案。各个阶段具体描述如下。

一是问题界定:明确问题的求解内容和待实现功能,包括问题的输入、输出和终止条件等。计算机只能处理有明确结果或终止条件的问题,通过问题界定可判断该问题能否利用计算机解决。

二是问题分解:将复杂任务层次化分解为可处理的子任务,可从过程、功能等维度分解问题,有助于简化和解决问题,各个击破。例如,复杂系统的设计可先将其按功能分解为若干子系统,再将子系统按过程分解为若干子过程。

三是模式匹配:将已有的解决其他问题的方法迁移到新问题解决中;或通过发现子问题

之间的异同点,将子问题的解决方法进行封装以简化整个问题。模式匹配也是知识学习和方法学习的重要思维过程,如枚举法、递归法等经典算法的学习需要通过模式匹配归纳总结出算法的核心思想和适用场景。化归、比较、类比等方法有助于模式匹配。

四是数据抽象与表达:提取出与问题解决相关的对象及对象之间的关系,并通过编码、变量、表达式、集合、列表、类、程序指令等方式对问题进行重新表达。将现实世界中各种数量关系、空间关系、逻辑关系和处理过程表示成计算机世界中的数据结构和控制结构,建立计算模型。

五是算法设计:设计问题解决的步骤,即指令执行的顺序,算法是编程的前提,通常用伪码或流程图等表述。当将伪码或流程图通过编程实现时,就是将人类指令翻译成计算机指令。

六是评估反思:将算法通过编程实施,并衡

量方案和执行结果,包括程序的时间复杂度、空间复杂度、健壮性和可扩展性等。通过不断评估和反思,我们可以发现并提出新问题,逐步优化问题解决方案。

二、面向计算思维培养的编程教学流程设计

传统编程教育强调的是语法和算法的教学,教学的起点往往是已经明确的并适合利用编程来解决的问题,但如何从现实问题过渡到计算机能够解决的问题却很少涉及,而这恰恰是问题解决的起点。因此,本研究提出了面向计算思维培养的编程教学流程,把基于计算思维的问题解决流程融入创设情境、分析问题、设计方案、实施反馈、项目迁移和分享总结等六个教学过程中,各教学过程由一系列教学活动组成,教学活动主要由教师和学生行为组成,具体流程如图 1。

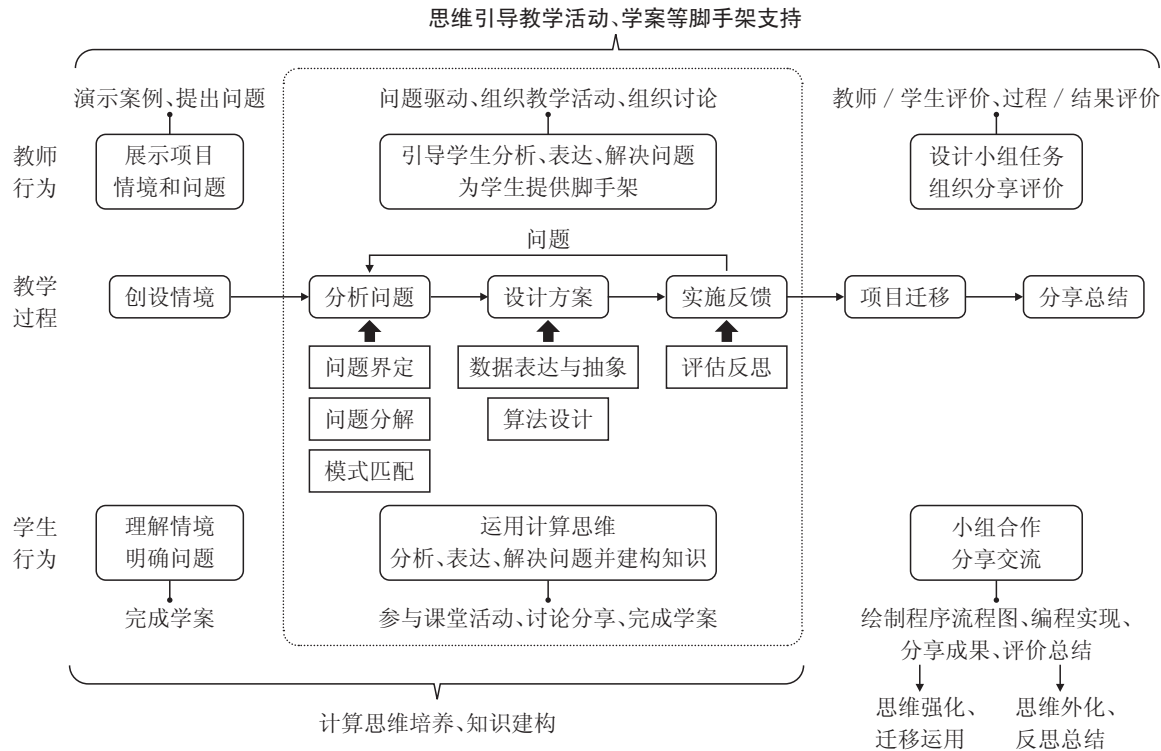


图 1 面向计算思维培养的编程教学流程

(一) 创设情境

项目式学习中,项目是具有情境的。在正式教学活动开始之前,教师应向学生展示项目的情境和待解决的问题,激发学生的学习兴趣和学习动机。同时,学生也需要理解情境,明确要实现的功能和待解决的问题,为接下来的问题解决

做好准备。

(二) 问题解决: 分析问题、设计方案、实施反馈

问题解决过程包括分析问题、设计方案和实施反馈三个阶段。与传统的教师直接指定项目解题路径的教学方式不同,面向计算思维培养

的编程教学自下向上逐步叠加项目功能或问题复杂度,采用问题驱动教学方式,引导学生对已有解决方案及实施结果进行评估和反思,发现并提出方案中存在的问题,进而进行新一轮的分析问题、设计方案和实施反馈,直到获得一个相对完善的问题解决方案。在教学中,教师要针对问题解决各过程中所需的计算思维设计相应的思维引导教学活动,引导学生运用计算思维方法主动分析问题、表达问题和解决问题,组织讨论并鼓励学生积极思考、表达分享,设计学案等脚手架为学生的问题解决过程提供支持等。

#### 1. 分析问题

分析问题一般需要经过问题界定、问题分解和模式匹配三个过程。在问题界定过程中,教师可以通过学案等脚手架引导学生明确问题的输入、输出和终止条件,判断问题是否能用计算机解决。在问题分解过程中,教师可以组织学生进行讨论,引导学生将问题分解为若干子问题,并确定子问题中哪些是运用已有知识经验可被解决的、哪些是暂时不能被解决的。完成问题分解后,若子问题与已有知识经验存在关联,教师可通过相关生活经验或教学活动引导学生运用模式匹配思维从中寻找子问题的解决办法。若子问题与子问题之间存在关联,教师可引导学生将子问题和子问题进行比较,发现其间的异同点,从而简化并解决问题。这一过程中,教师可以设计表格等工具让学生列举出新问题和已有问题之间或子问题和子问题之间的异同点,有助于学生厘清思路、发现模式。

#### 2. 设计方案

设计方案需要经过数据抽象与表达、算法设计两个过程。在数据抽象与表达过程阶段,教师可以借助对象表征表、对象关系表征表、对象处理过程表征表等脚手架引导学生提取出与问题解决相关的对象和对象之间的关系,并用合理的数据结构完成数据表达。对于较复杂的关系,教师可以鼓励学生用图或表等可视化方式来表述对象之间的关系。数据抽象与表达需要以编程知识作为基础,因此教师要及时进行知识讲解和补充。算法设计过程中,教师可以通过启发式教学活动、小组讨论等,引导学生设计问题解决方案并与其他同学分享交流。

#### 3. 实施反馈

实施反馈阶段,学生需要对方案、程序和执行结果进行评估反思,并针对评估反思之后提出的问题进行新一轮的问题解决。这一环节的关键是发现问题并提出问题。教师可以引导学生对比现有方案的实施效果与已有知识经验之间的差别,从而提出现有方案中存在的问题;也可以增加问题复杂度或项目功能,启发学生在简单问题解决的基础上思考如何解决更复杂的问题或实现更复杂的功能。

#### (三) 项目迁移

在项目迁移环节,教师需设计相似或更高级的任务作为小组合作任务。项目迁移需要学生在对已有知识和方法掌握的基础上找到迁移问题与项目问题之间的联系,灵活运用所学的知识和方法设计解决方案、绘制程序流程图并编程实现。这是对学生计算思维尤其是模式匹配思维的强化,同时也是对所学知识的巩固,有助于培养学生的迁移运用能力、问题解决能力和小组合作能力。

#### (四) 分享总结

在分享总结环节,各小组展示自己的成果,包括小组的问题解决思路、代码设计等,促进学生的思维外化。教师将对各小组进行过程性评价和总结性评价,组织学生进行自评和互评,最后对教学内容进行总结和升华。通过评价和总结,学生对运用计算思维解决问题的过程进行反思和回顾,强化思维。

### 三、面向计算思维培养的编程教学案例

#### ——《田忌赛马》

本课以田忌赛马为情境,提出初级问题“能否利用计算机找到使田忌的马获胜的出场方式”,之后增加问题复杂度,提出“若田忌的马对齐王的马有一定胜率,能否利用计算机找到使田忌的马胜率最大的出场方式”这一问题。对于初级问题,学生需要用枚举算法找到所有出场顺序及对应的获胜方。对于高级问题,学生需要用二维数组存储田忌的马对齐王的马的胜率,利用枚举算法在循环中获取胜率,从而计算出各组合的总胜率。在这个过程中,学生要利用列表的动态生成保存各出场方式,



最后计算出最大胜率及对应的出场方式。

### (一) 创设情境

该环节中,教师讲述田忌赛马故事并提出问题:“田忌的马有哪些出场顺序?田忌如何能赢?”学生思考马的出场顺序及每种出场顺序的获胜方,画出如下对战表(见表1)。

表1 田忌的马与齐王的马对战表

	第一场	第二场	第三场	获胜方
齐王	上	中	下	
田忌1	上	中	下	齐王
田忌2	上	下	中	齐王
田忌3	中	上	下	齐王
田忌4	中	下	上	齐王
田忌5	下	上	中	田忌
田忌6	下	中	上	齐王

### (二) 问题解决——田忌赛马初级问题

#### 1. 界定问题

该环节涉及的计算思维是问题界定。在这个环节,教师提出问题:“该问题可以用计算机解决吗?若可以,输入、输出和终止条件各是什么?”学生讨论后提出该问题可以用计算机解决,输入为田忌的马、齐王的马的等级,输出为所有出场顺序及获胜方。

```
horse=('上','中','下')# 存储田忌的马
n=0# 记录出场方式的数目
#i,j,k 分别代表第1个、第2个、第3个出场的马的序号
for i in range(3):
    for j in range(3):
        for k in range(3):
            if i!=j and i!=k and j!=k:
                n=n+1
                print(" 出场顺序 {}:{} {}".format(n, horse[i], horse[j], horse[k]))
```

#### 4. 解决子问题2: 判断并输出各出场方式的获胜方

该环节涉及的计算思维包括数据抽象与表达、算法设计和模式匹配,由两个连续的教学活动组成。

第一,教师提出问题:“比赛规则是三局两胜,程序如何判断田忌是否获胜?”然后,教师

```
win=0
if horselevel[i]-3 > 0:
    win=win+1
if horselevel[j]-2 > 0:
    win=win+1
if horselevel[k]-1 > 0:
    win=win+1
if win >= 2:
    print(" 出场顺序 {}:{} {}: 获胜)".format(n, horse[i], horse[j], horse[k]))
else:
    print(" 出场顺序 {}:{} {}: 失败 !)".format(n, horse[i], horse[j], horse[k]))
```

### 2. 分解问题

该环节涉及的计算思维是问题分解。在此环节,教师提出问题:“利用计算机找到使田忌的马获胜的出场方式,需经过哪些步骤?”学生经讨论后将问题分解为两个子问题:一是找到所有的出场方式;二是判断每种出场方式中的获胜方。

#### 3. 解决子问题1: 找到所有出场方式

该环节涉及的计算思维包括数据抽象与表达、算法设计,由两个连续的教学活动组成。

第一,教师提出问题:“若不限制马的出场次数,田忌的马有哪些出场方式?如何设计程序输出所有出场方式?”学生讨论后提出,田忌的马和齐王的马排列组合后共有  $3 \times 3 \times 3 = 27$  种方式,可用三重循环输出。

第二,教师提出问题:“若各匹马只能出场一次,田忌的马有哪些出场方式?如何修改程序输出该规则下的所有出场方式?”学生讨论后提出,不限制出场次数时,循环中  $i, j, k$  可相同;限制出场次数时,  $i, j, k$  必互异。因此,需在循环中增加判断,满足“互异”条件的才输出。

解决该问题时输出的程式如下。

对学生进行思维引导:“判断胜负中,确定马的等级大小关系是关键,程序如何反映马匹的等级大小?”学生讨论后提出可用数值表示马的等级大小:horseLevel=(3, 2, 1)。由此,将现实问题中的胜负比较转化为计算机可处理的数值运算,同时用变量 win 记录获胜次数,条件判断  $\text{win} \geq 2$  时获胜。解决此问题时输出的程式如下。

第二,教师以田忌赛马问题为背景讲解枚举算法,并以“从一筐绿豆中挑出混入的红豆”等生活经验加深学生对枚举算法的理解,组织学生讨论枚举算法的适用场景和基本结构。学生讨论后提出,当研究对象可数且有范围,最优解可从该范围中逐一列举检验得到时,可用枚举算法实现,其基本结构为确定范围、列举和检验。之后,教师引导学生思考生活中有哪些问题可以用枚举算法解决。

#### 5. 实施反馈

该环节涉及的计算思维是评估反思。在此环节,教师提出问题:“调整出场顺序使得田忌的马能以弱敌强。实际生活中,实力悬殊的两支队伍比赛,是否也一定能找到以弱敌强的方法呢?”学生讨论后提出,在生活中,胜利与否具有一定概率,中等马不一定总赢下等马。当田忌的马对齐王各匹马的胜率过低时,不管什么方法都不能取得胜利。

### (三) 问题解决——田忌赛马高级问题

#### 1. 界定问题

该环节涉及的计算思维是问题界定。在此环节,教师提出问题:“若知道田忌的马对齐王的马的胜率,如何找到总胜率最大的出场方式?”学生讨论后提出,该问题的输入为田忌的马对齐王的马的胜率,输出为最大总胜率及对应的出场方式。

#### 2. 分解问题

该环节涉及的计算思维是问题分解。在此环节,教师提出问题:“若利用计算机寻找田忌的马最高胜率的出场方式,需要经过哪些步骤?”学生讨论后将问题分解为三个子问题:一是找到所有的出场顺序(已解决);二是计算各出场顺序中田忌的马的总胜率;三是找到并输出最大胜率及其出场顺序。

3. 解决子问题2: 计算各出场顺序中田忌的马的总胜率

该环节涉及的计算思维包括数据抽象与表达、算法设计和模式匹配,由三个连续的教学活动组成。

第一,教师提出问题:“总胜率如何计算?若田忌的马出场顺序为A-B-C,齐王的马为X-Y-Z,田忌的马总胜率是多少?”学生讨论提出,该方式的总胜率为 $(0.4+0.3+0.4)/3=0.37$ ,胜率的

计算方法为(第一回合胜率+第二回合胜率+第三回合胜率)/3。

第二,教师提出问题:“胜率是齐王的马和田忌的马两两对象之间的关系,如何用更简洁的方式表达?”然后,教师引导学生用二维表格表示田忌的马对齐王的马的胜率(见表2)。

表2 田忌的马对齐王的马的胜率

胜率	X	Y	Z
A	0.4	0.8	0.9
B	0.2	0.3	0.8
C	0.1	0.2	0.4

第三,教师提出问题:“如何在循环中获取胜率并计算出总胜率?”学生类比之前在循环中取出horse元组中的值,提出可用二维元组winP=((0.4,0.8,0.9),(0.2,0.3,0.8),(0.1,0.2,0.4))存储胜率,以便循环取出胜率并计算总胜率,编程式如下。

```
totalwinP=(winP[i][0]+winP[j][1]+winP[k][2])/3
totalwinP=round(totalwinP,2)
```

4. 解决子问题3: 找到并输出最大胜率及其出场顺序

该环节涉及的计算思维包括数据抽象与表达、算法设计,由两个连续的教学活动组成。

第一,教师提出问题:“如何找到最大胜率?”然后,教师对学生进行思维引导:“拍卖活动中如何售出最高价?”学生类比拍卖方式,讨论提出每产生一个对战方式就将该方式的总胜率totalWinP与现有最高胜率totalWinP\_max比较,若更大,则更新最高胜率。此时,输出的编程式如下。

```
if totalwinP > totalwinP_max:
    totalwinP_max=totalwinP
    maxIndex=n-1
```

第二,教师提出问题:“如何找到最大胜率的出场顺序?”然后,教师对学生进行思维引导:“不考虑胜率和考虑胜率的两种情况下,问题解决有何区别?”学生讨论后提出,前者可在过程中直接判断是否获胜,后者必须在所有出场方式都产生出来之后才能计算出最大胜率。因此,学生要在过程中保存各出场方式,并在所有出场方式生成之后根据最大胜率的索引找到最大胜率下的出场方式。教师讲解用列表的append()方法保存各生成的出场方式。此时,输出的编程式如下。

```

for i in range(3):
    for j in range(3):
        for k in range(3):
            if i!=j and i!=k and j!=k:
                n=n+1
                combineways.append((horse[i], horse[j], horse[k]))
                totalwinP=(winP[i][0]+winP[j][1]+winP[k][2])/3
                totalwinP= round(totalwinP, 2)
                if totalwinP > totalwinP_max:
                    totalwinP_max=totalwinP
                    maxIndex=n-1
combineway_max=combineways[maxIndex]
print("总胜率最大为:{}".format(totalwinP_max))
print("出场顺序为:", end=' ')
for i in range(3):
    prin(combineWay_max[i], end=' ')

```

## 5. 实施反馈

该环节涉及的计算思维为评估反思,涉及的知识为枚举算法的局限和时间复杂度。在此环节,教师提出问题:“田忌的马和齐王的马各3匹时,程序需要计算多少次?当双方马匹数量很多时,用枚举算法有什么局限?”同时,教师讲解时间复杂度的概念及计算方法。学生讨论得出,当双方各有 $n$ 匹马、比赛 $n$ 局,且 $n$ 匹马不能重复上场,时间复杂度为 $O(n!)$ , $n$ 较大时,程序将计算很久。

### (四)项目迁移

该环节中,教师发布小组任务:“在一条笔直的公路上,每隔10千米有A、B、C、D、E五个粮站,分别存有10、20、30、0、40吨粮食,现要把粮食运到一个站点,每吨运1km需0.5元,运到哪个站费用最少?”教师引导学生找到运粮问题和田忌赛马问题的联系:一是研究对象都是可数的且有一定范围,最优解可以从这个范围中逐一列举检验得到,因此运粮问题也可以用枚举算法解决;二是站与站之间的路程关系和马与马之间的胜率关系一样,可以用二维元组存储。

### (五)分享总结

该环节中,各小组展示分享作品,包括实现的功能、解决思路及方案、遇到的问题等。教师对小组作品进行评价总结,组织小组互评,包括过程性评价和总结性评价。

### (六)学案设计示例

#### 示例1:问题分析过程学案

若每匹马可以重复出场,有哪些出场顺序?若每匹马不可重复出场,又有哪些出场顺序?这两种规则导致的结果有何不同?在不可

重复出场的情况下,如何设计程序输出田忌马的所有出场方式?

	出场方式的数量	出场方式的结构特点
可重复出场		
不可重复出场		

#### 示例2:问题表达过程学案

利用计算机解决问题的关键在于找到与问题解决相关的对象及对象之间的关系,并将问题重新表达为计算机可处理的形式。

对象	变量
田忌的马	horse = ['上', '中', '下']

对象之间的关系	表达式
田忌获胜	win >= 2

对象处理过程	程序指令
田忌获胜次数加1	win = win + 1

本研究提出了面向计算思维培养的编程教学流程,打破了以往以知识为主线的教学思路,把基于计算思维的问题解决流程融入创设情境、分析问题、设计方案、实施反馈、项目迁移和分享总结六个教学过程中,让学生学会从现实问题过渡到计算机能够解决的问题,将计算思维方法迁移到实际生活的问题解决中。该教学流程通过叠加项目功能或增加问题复杂度激发学生不断发现并提出新的问题;通过思维引导教学活动和学案等脚手架引导并支持学生经历计算思维各个过程;通过讨论分享活动促进学生表达观点、外化思维。

在研究的具体实践中发现,相比于传统的教学方式,学生在按此教学流程进行的教学中表现出更强的逻辑性和积极性,在独立解决新问题时表现得更好。在之后的研究工作中,笔者将进一步展开实践,逐步完善和优化面向计算思维培养的编程教学流程,使该流程对中学编程教学实践更具有参考价值。

#### 参考文献:

- [1] 国务院印发《新一代人工智能发展规划》[J]. 中小学信息技术教育, 2017 (9): 4.
- [2] 张宁. 编程教学优质高效课堂教学的分析与思考[J]. 中国信息技术教育, 2018(17): 21-23.
- [3] 郁晓华, 肖敏, 王美玲, 陈妍. 基于可视化编程的计算思维培养模式研究——兼论信息技术课堂中计算思维的培养[J]. 远程教育杂志, 2017(6): 12-20.

[4] JEANNETTE WING. Computational thinking[J]. Communications of the ACM, 2006: 33-35.

[5] ISTE, CSTA. Operational Definition of Computational Thinking for K-12 Education[EB/OL]. [2019-06-12]. <https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn=2>.

[6] CYNTHIA SELBY, JOHN WOOLLARD. Computational Thinking: The Developing Definition[EB/OL]. [2019-06-12]. <http://eprints.soton.ac.uk/356481>.

[7] SUSAN DAVIDSON, CHRIS MURPHY. Computational Thinking for Problem Solving[EB/OL]. [2019-06-15]. <https://www.coursera.org/learn/computational-thinking-problem-solving/home/welcome>.

[8] 中华人民共和国教育部. 普通高中信息技术课程标准(2017年版)[S]. 北京: 人民教育出版社, 2017.

## Research on Programming Teaching for Computational Thinking: Taking Python Programming Teaching in High School as an Example

FU Qian WANG Yuru

(Beijing Normal University, Beijing 100875, China)

**Abstract:** With the advent of artificial intelligence era, the cultivation of computational thinking in programming education is becoming the focus of information technology education. Problem-solving based on computational thinking generally includes defining problem, decomposing problem, pattern matching, data abstraction and expression, algorithm design, and evaluating & reflecting. Programming teaching for computational thinking training is centered on project learning and problem solving. It consists of creating situations, analyzing problems, designing solutions, implementing feedback, transferring project and sharing reflection. The process emphasizes overlapping projects or increasing problem complexity. To inspire students to use computational thinking to solve problems, the teaching case of "Tianji Horse Racing" based on this process is designed to provide reference for such teaching and to promote research related to computational thinking and programming education.

**Key words:** programming education; computational thinking; teaching process