

# 初中信息技术教学中利用 C++ 培养学生计算思维的策略研究

李冬

上饶市第二中学

**摘要：**随着信息技术教育的深化，计算思维作为核心素养已成为初中信息技术课程的重要目标。C++ 语言以其结构化编程特性与算法实现优势，成为培养初中生计算思维的有效载体。本文结合初中阶段学生认知特点，分析 C++ 教学与计算思维培养的内在关联，提出情境化任务设计、模块化项目实践、分层思维训练等策略，并通过具体教学案例阐述实施路径，旨在为初中编程教学提供可借鉴的思路。

**关键词：**初中信息技术；C++ 编程；计算思维；教学策略

**【DOI】** 10.12252/j.issn.2096-627X.2025.08.168

## 引言

计算思维是数字化时代的核心素养，《义务教育信息技术课程标准（2022 年版）》明确要求初中阶段通过编程教学发展学生计算思维与问题解决能力。C++ 语言的结构化特性与算法逻辑，为初中编程教学提供了重要载体。

当前教学中，初中生抽象思维发展尚不成熟，易陷入“重代码、轻思维”的误区，且传统“讲解+模仿”模式难以落实计算思维的核心训练。如何结合 C++ 语言特点与初中生认知规律，设计有效教学策略，成为初中信息技术课程的重要课题。

本研究立足教学实践，分析计算思维培养与初中阶段的适配性，探索情境任务、项目实践等策略，为编程教学中融入思维训练提供可参考路径。

## 一、计算思维的内涵及初中阶段培养价值

### （一）计算思维的本质特征

计算思维是运用计算机科学的思维方法界定问题、抽象模型、设计算法的思维过程，核心包括问题分解（将复杂问题拆解为子任务）、模式识别（提炼问题共性与规律）、抽象建模（用符号化语言描述问题本质）、算法设计（构建有序解决方案）四个维度<sup>[1]</sup>。在初中阶段，培养计算思维可帮助学生建立“用计算机解决问题”的思维习惯，为高中阶段算法与程序设计学习奠定基础。

### （二）C++ 语言的教学适配性

C++ 作为兼具低级语言效率与高级语言抽象性的编程语言，其特性与初中计算思维培养需求高度契合<sup>[2]</sup>：

1. 结构化编程：通过函数模块化设计，引导学生将问题分解为可复用的功能模块，如将“学生成绩统计”拆解为数据输入、排序、计算平均分等函数，强化“分而治之”思维。

2. 基础算法直观性：通过循环（for/while）、条件判断（if-else）等语句实现经典算法（如冒泡排序、斐波那契数列），适合初中生从具体案例中理解算法逻辑。

3. 数据类型抽象训练：通过定义整型（int）、浮点型（float）、字符型（char）等变量，培养学生将现实问题数字化抽象的能力，如用变量存储“学生年龄”“商品价格”等信息。

### （三）初中阶段培养意义

1. 衔接小学与高中的思维进阶：小学阶段图形化编程（如 Scratch）侧重直观操作，高中阶段侧重复杂算法与面向对象编程，初中阶段借助 C++ 可实现从“可视化思维”到“代码逻辑思维”的过渡<sup>[3]</sup>。

2. 提升问题解决能力：通过 C++ 编程实践，学生会用计算思维分析学科问题（如数学函数图像绘制、物理数据模拟），增强跨学科迁移能力。

## 二、初中 C++ 教学中计算思维培养的现存问题

### （一）学生认知与语言难度的矛盾

C++ 语法（如指针、类）对初中生抽象思维要求较高，部分教师直接讲授语法规则，导致学生陷入“死记代码”误区，忽视思维过程培养。例如，学生能背诵“for 循环格式”，却无法用循环解决“求 1-100 累加和”的问题。

### （二）教学方式单一化

传统“讲解-模仿”模式缺乏思维激活，学生被动接受代码模板，难以体验计算思维的建构过程。如在“水仙花数”教学中，教师直接给出代码，学生未经历“问题分解→算法设计→代码调试”的完整思维链条。

### （三）评价侧重结果而非思维过程

作业与测试多以“程序能否运行”为标准，忽视对问题分解能力、算法优化意识的评价。例如，学生用暴

力枚举法解决“百钱买百鸡”问题，教师未引导其思考更高效的数学建模方法<sup>[4]</sup>。

### 三、基于 C++ 的计算思维培养策略与实践

#### (一) 优化认知基础：情境化任务驱动思维建构

##### 1. 生活情境引入，激活问题意识

校运会 4×100 米接力赛的计时需求，为编程教学提供了天然的问题载体。当提出“如何用程序记录四棒成绩”时，学生需理解变量作为数据存储单元的功能——如同为每位选手准备独立的“数据盒子”，需通过唯一变量名（如 runner1Time）区分存储内容，若统一使用“time”会导致数据混淆。选择 float 类型存储成绩，因其能精确表达 10.23 秒这类小数数值，这一过程让学生直观理解数据类型与现实需求的映射关系。

在“计算总时长”任务中，将加法逻辑封装为函数的设计，可类比计算器的“+”按钮功能。学生通过讨论“工具如何获取输入数据”，理解参数传递的机制，进而建立“黑盒思维”——只需关注输入（四棒时间）与输出（总时长）的逻辑关联，无需深究内部计算过程。这种抽象与封装思想，是计算思维的核心要素。

处理多组队伍成绩时，循环结构成为关键解决方案。以“流水线作业”比喻 for 循环的批量处理机制，引导学生设计循环条件（如根据队伍数量确定循环次数）与最优值比较算法（通过 if 语句更新最佳成绩）。借助流程图可视化“输入-计算-比较”的重复逻辑，学生能理解循环如何通过“临时冠军”变量（存储当前最优值）实现高效数据处理，完成从具体需求到算法模型的抽象转化。

##### 2. 跨学科融合，拓展思维边界

编程与数学的融合能揭示不同领域问题的“计算本质”。在绘制  $y=2x+1$  ( $x \in [0, 5]$ ) 的字符图像时，学生首先面临连续数学思维与离散计算思维的冲突：数学中光滑的直线在屏幕上需转化为离散的字符点阵。通过引导“采样”思想，学生理解选取  $x=0, 1, 2, 3, 4, 5$  作为

离散点的必要性，并用 for 循环遍历计算对应 y 值，完成从连续函数到离散数据的转化。

坐标映射是任务的核心难点。学生需辨析数学坐标系（原点左下，y 轴向上）与屏幕坐标系（原点左上，y 轴向下）的差异，设计映射公式（如 screenRow= 基准行 - y× 缩放比例）。这一过程类似物理中的“坐标系变换”，通过缩放因子调整图像大小、基准行定位垂直位置，最终在循环中通过行列坐标精准输出“\*”字符（辅以空格填充定位）。调试中图像歪斜等问题，促使学生反思映射公式的符号逻辑与采样步长的合理性，深化对“离散化建模”的理解。

当屏幕呈现由“\*”构成的斜线时，学生不仅掌握了循环、计算等编程技能，更体会到“建模”的普适性——将数学函数抽象为计算机可处理的离散步骤，本质上与物理实验的数据采样、工程信号的数字化处理共享同一思维逻辑。这种跨学科的思维迁移，让学生认识到编程作为通用工具的问题解决价值，其成就感源自“用计算思维连接知识孤岛”的认知突破。

##### 3. 教学实践的思维发展价值

从运动会计时系统到函数图像绘制，教学过程贯穿“需求分析-抽象建模-算法设计-调试优化”的完整问题解决链。变量对应现实事物的符号化表征，函数体现操作逻辑的模块化封装，循环实现批量任务的自动化处理，而跨学科任务则凸显计算思维作为“元能力”的迁移价值。这种将编程知识与生活经验、学科概念深度绑定的教学策略，既降低了认知门槛，又通过真实问题的复杂性培养了学生的系统性思维——当他们能自如地用代码描述现实规则时，计算思维已从知识技能内化为解决问题的思维习惯。

#### (二) 创新教学模式：模块化项目进阶训练

##### 1. 阶梯式项目设计，分层培养思维能力

根据学生水平设计“基础→进阶→挑战”三级项目，逐步提升思维复杂度。

项目层级	案例名称	核心思维训练点
基础层	简易计算器	变量定义、运算符优先级、条件判断
进阶层	班级通讯录管理	结构体定义、数组存储、函数封装
挑战层	迷宫寻路程序	二维数组建模、深度优先搜索算法实现

##### 2. 小组合作探究，强化思维碰撞

采用“问题共探-方案互评-优化迭代”模式，让学生在合作中暴露思维差异。

案例：图书管理系统开发

分组任务：每组设计一个包含“借书、还书、查询”功能的控制台程序。

思维互动：小组讨论，头脑风暴，寻找最优算法；

组内讨论：如何设计数据结构（数组 / 链表）存储图书信息（抽象建模）；

跨组互评：对比不同小组的查询算法效率（线性查找 vs 二分查找），引导优化思维。

（三）深化思维训练：可视化调试与算法优化

1. 调试工具辅助，具象化思维过程

利用 Dev-C++ 等 IDE 的调试功能（断点设置、变量监控），让学生观察代码执行流程，直观理解抽象逻辑。

案例：冒泡排序调试

步骤：设置断点逐行执行排序代码，通过监控数组元素变化，理解“相邻元素比较 - 交换”的核心逻辑，避免机械记忆算法步骤。

2. 算法优化竞赛，培养思维敏捷性

定期开展“一题多解”比赛，鼓励学生用不同算法解决同一问题，比较时间复杂度与空间复杂度。

案例：求素数问题

任务：编写程序判断一个数是不是素数，素数就是除了 1 和它自身外，不能被其他自然数整除的数。

思维训练

基础解法：直观判断法

算法逻辑：要判断一个数 n 是不是素数，就看它不能被 2 到 n-1 之间的数整除。

时间复杂度：大概是 O(n)，因为要检查 n-2 次。

代码示例：

```

1  bool isPrime(int n) {
2      if (n <= 1) return false;
3      for (int i = 2; i < n; i++) {
4          if (n % i == 0) return false;
5      }
6      return true;
7  }
8

```

（四）完善评价体系：过程性思维评估

设计“计算思维能力评价量表”，从问题分解、抽象建模、算法设计、调试优化四个维度进行过程性评价。

评价指标示例：

维度	评价要点	等级标准（示例）
问题分解	能否将复杂问题拆解为可执行的子任务	初级：需教师引导分解；高级：独立完成拆解
抽象建模	能否用代码符号描述问题本质	初级：直接复制模板；高级：自主设计数据结构
算法设计	能否设计高效、无歧义的解决方案	初级：实现功能即可；高级：追求算法优化

四、教学实践效果与反思

（一）实践成效

在某初中初二年级开展一学期教学实验，对比实验班（采用 C++ 计算思维培养策略）与对照班（传统编程教学），结果显示：

实验班学生在“问题分解”、“算法设计”测试中的平均分分别比对照班高 21.3%、18.7%；有 92% 的实验班学生表示“编程不再是背代码，而是像解谜题一样有趣”。

（二）反思与改进

避免过早引入指针、类等高级概念，初中阶段聚焦结构化编程（函数、循环、数组），通过类比（如将数组比作“抽屉”，索引为“抽屉编号”）降低抽象难度。

增加“无代码思维”活动，如用流程图设计算法、用自然语言描述问题解决步骤，强化计算思维与日常问题的联结。

结语

在初中信息技术教学中，利用 C++ 培养计算思维需遵循“从具象到抽象、从单一到综合”的认知规律。教

师应跳出“语法教学”的窠臼，以真实问题为载体，通过情境化任务、项目式学习、可视化调试等策略，让学生在“做中学”中体验计算思维的建构过程。<sup>[5]</sup>未来可进一步融合 AI、物联网等前沿技术，拓展 C++ 的应用场景，让计算思维成为学生应对数字化挑战的核心能力。

参考文献

[1] 周以真. 计算思维 [J]. 中国计算机学会通讯, 2007, 3 (3): 33-35.

[2] 中华人民共和国教育部. 义务教育信息技术课程标准 (2022 年版) [S]. 北京: 人民教育出版社, 2022.

[3] 王荣良. 计算思维养成: 概念、工具及教学策略 [M]. 北京: 高等教育出版社, 2016.

[4] 李艺, 钟柏昌. 计算思维: 信息技术课程的价值追求 [J]. 中国电化教育, 2012 (11): 5-9.

[5] 任友群, 隋丰蔚, 李锋. 数字土著何以可能? ——也谈计算思维进入中小学信息技术教育的必要性和可能性 [J]. 中国电化教育, 2016 (1): 1-8.