

2024-2025 学年初中信息技术（信息科技）七年级 下册大连理工版（2015）教学设计合集

目录

一、第一单元 走进 Scratch 世界

- 1.1 第一课 可爱的小猫——认识 Scratch
- 1.2 第二课 巧解数学题——变量的应用
- 1.3 第三课 小猫报站名——链表的应用
- 1.4 第四课 我是艺术家——【外观】与【声音】模块组
- 1.5 第五课 我是小画家——【画笔】模块组
- 1.6 第六课 神奇的画笔——多层嵌套循环结构
- 1.7 第七课 巧接弹力球——复杂循环结构与跳出循环
- 1.8 第八课 双人赛车——循环结构与选择结构
- 1.9 第九课 小猫顶足球——侦测与机会指令
- 1.10 第十课 疯狂打地鼠——初始化游戏与“广播”命令
- 1.11 本单元复习与测试

二、第二单元 神奇的机器人

- 2.1 第十一课 认识机器人——机器人部件及控制器
- 2.2 第十二课 转动的风车——齿轮传动
- 2.3 第十三课 交通信号灯——“机器人快车”软件
- 2.4 第十四课 搬运机器人——马达的操控方法
- 2.5 本单元复习与测试

第一单元 走进 Scratch 世界第一课 可爱的小猫——认识 Scratch

主备人	
备课成员	
	1.

教学 内容 分析	<p>本节课的主要教学内容：本节课将带领学生们通过 Scratch 软件制作一个可爱的小猫动画，通过这个案例让学生们初步了解 Scratch 软件的基本操作和编程思想。</p> <p>2. 教学内容与学生已有知识的联系：本节课的教学内容与课本《初中信息技术（信息科技）七年级下册大连理工版（2015）》第一单元“走进 Scratch”中的第一课“可爱的小猫—认识 Scratch”紧密相关。学生们将运用已学到的信息技术知识，如计算机操作、图形界面等，来学习 Scratch 软件的基本操作。同时，通过本节课的学习，学生们能够将编程思想与实际操作相结合，为后续 Scratch 编程学习打下基础。</p>				
核心 素养 目标 分析	<p>本节课旨在培养学生的信息技术核心素养，具体目标如下：</p> <ol style="list-style-type: none"> 1. 培养学生的计算思维：通过 Scratch 编程，让学生体验算法设计、逻辑推理和问题解决的过程，提高计算思维能力。 2. 增强学生的创新意识：鼓励学生在 Scratch 软件中创作个性化作品，激发创新意识，培养学生的创意思维。 3. 提高学生的信息素养：使学生了解 Scratch 作为一种图形化编程语言的特点和优势，增强对信息技术的理解和应用能力。 4. 培养学生的团队协作能力：在课堂活动中，引导学生相互交流、分享学习心得，培养良好的团队协作精神。 5. 培养学生的信息道德意识：通过编程实践，使学生认识到信息技术在生活中的重要性，树立正确的信息道德观念。 				
学习 者分 析	<ol style="list-style-type: none"> 1. 学生已经掌握的相关知识： 学生在进入本节课之前，已经接触过计算机基础操作，对鼠标和键盘的使用有一定的了解。此外，部分学生可能已经接触过简单的图形界面操作，如 Windows 操作系统或一些基础的网络应用。然而，关于编程方面的知识，尤其是图形化编程语言如 Scratch，大多数学生可能没有接触过，这为本节课的教学提供了起点。 2. 学生的学习兴趣、能力和学习风格： 初中生对新鲜事物充满好奇心，Scratch 作为一种直观、有趣的编程工具，能够激发学生的学习兴趣。学生的学习能力方面，由于 Scratch 的图形化编程特性，学生不需要具备复杂的编程语言知识，因此大部分学生都能够快速上手。学习风格上，学生们可能表现出不同的偏好，有的学生可能更倾向于动手操作，而有的学生则可能更善于观察和模仿。 3. 学生可能遇到的困难和挑战： 对于初次接触编程的学生来说，理解编程逻辑和设计算法可能是一个挑战。此外，Scratch 软件的操作界面和编程块的使用可能需要一定的时间来熟悉。一些学生可能会在编程过程中遇到错误，需要通过调试来解决问题，这可能会让他们感到挫败。此外，由于 Scratch 编程涉及创意设计，学生可能会在创作个性化作品时遇到创意瓶颈。教师需要关注这些潜在困难，通过适当的指导和鼓励帮助学生克服。 				
学具 准备	Xxx				
课型	新授课	教法学法	讲授法	课时	第一课时
步骤	师生互动设计				二次备课
教学 方法 与策	1.				

略		
---	--	--

	<p>教学方法选择：本节课将采用讲授法、案例研究和项目导向学习相结合的教学方法。首先通过讲解 Scratch 的基本操作和编程概念，帮助学生建立初步认识。接着，通过案例研究，展示可爱小猫动画的制作过程，让学生直观理解编程逻辑。最后，采用项目导向学习，引导学生独立完成小猫动画的制作，培养实践能力。</p> <p>2. 教学活动设计：为了提高学生的参与度和互动性，将设计以下教学活动： (1) 角色扮演：让学生扮演 Scratch 编程导师，展示如何使用编程块； (2) 小组讨论：分组让学生讨论如何改进小猫动画的效果； (3) 实验：学生分组尝试不同的编程块组合，探索动画效果。</p> <p>3. 教学媒体使用：利用多媒体设备展示 Scratch 软件操作界面和编程块的示例，通过投影仪将屏幕内容展示给学生，方便学生跟随操作。同时，准备教学视频，展示 Scratch 动画制作的详细步骤，供学生课后复习和自主学习。</p>	
教学过程	<p>1. 导入（约 5 分钟）</p> <ul style="list-style-type: none"> - 激发兴趣：教师通过展示一些有趣的 Scratch 动画作品，如小猫跳跃、追逐等，引导学生思考这些动画是如何制作的，激发学生对编程的兴趣。 - 回顾旧知：教师简要回顾计算机基础操作和图形界面使用，帮助学生回忆与 Scratch 相关的已有知识。 <p>2. 新课呈现（约 20 分钟）</p> <ul style="list-style-type: none"> - 讲解新知：教师详细讲解 Scratch 软件的基本操作，包括界面布局、编程块类型、事件和动作等。 - 举例说明：教师通过制作一个简单的 Scratch 动画，如小猫眨眼，展示如何使用编程块实现动画效果。 - 互动探究：教师引导学生讨论如何改进小猫眨眼的动画，如增加眨眼频率、改变眨眼效果等。 <p>3. 新课呈现（续）（约 20 分钟）</p> <ul style="list-style-type: none"> - 讲解新知：教师继续讲解 Scratch 中的变量、控制结构等高级编程概念。 - 举例说明：教师通过制作一个复杂的小猫动画，如小猫追逐鼠标，展示如何使用变量和控制结构实现动画效果。 - 互动探究：教师引导学生思考如何让小猫在追逐鼠标时改变方向，提高动画的趣味性。 <p>4. 巩固练习（约 30 分钟）</p> <ul style="list-style-type: none"> - 学生活动：学生分组，每组选择一个主题，如小猫跳跃、追逐等，利用 Scratch 软件制作动画。 - 教师指导：教师在学生制作过程中巡回指导，解答学生在编程过程中遇到的问题，帮助学生克服困难。 <p>5. 巩固练习（续）（约 30 分钟）</p> <ul style="list-style-type: none"> - 学生展示：每组学生展示自己的动画作品，其他组学生进行评价和反馈。 - 教师点评：教师对学生的作品进行点评，指出优点和不足，并提出改进建议。 <p>6. 总结与反思（约 5 分钟）</p> <ul style="list-style-type: none"> - 教师总结：教师对本节课的主要内容进行总结，强调 Scratch 编程的基本概念和技巧。 - 学生反思：学生回顾本节课的学习内容，思考自己在编程过程中的收获和不足，提出改进措施。 	

7. 课后作业 (约 10 分钟)

- 教师布置课后作业：让学生独立完成一个小猫动画，如小猫跳跃、追逐等，并尝试添加新的功能，如增加背景音乐、改变角色造型等。

-

	<p>教师提醒：教师提醒学生在课后及时复习本节课的内容，巩固所学知识。</p> <p>教学过程中，教师应根据学生的实际情况调整教学内容和进度，确保每位学生都能够跟上教学节奏。同时，教师应关注学生的个体差异，给予不同层次的学生适当的指导和帮助。</p>	
学生 学习 效果	<p>学生学习效果主要体现在以下几个方面：</p> <ol style="list-style-type: none"> 1. 编程技能的提升：通过本节课的学习，学生能够掌握 Scratch 软件的基本操作，包括界面布局、编程块类型、事件和动作等。学生能够运用这些技能制作简单的动画，如小猫眨眼、跳跃等，为后续的编程学习打下坚实的基础。 2. 计算思维能力的培养：学生在制作动画的过程中，需要运用逻辑思维和算法设计，这有助于提高学生的计算思维能力。学生通过解决编程中的问题，如控制角色动作、处理变量等，能够锻炼自己的问题解决能力。 3. 创意和设计能力的增强：本节课鼓励学生发挥创意，设计个性化的动画作品。学生在创作过程中，不仅能够提高自己的审美能力，还能够学会如何将创意转化为实际的作品。 4. 团队合作能力的提升：在小组合作完成动画制作的过程中，学生需要相互沟通、协作，这有助于培养学生的团队协作能力。学生通过分工合作，共同解决问题，能够学会如何在团队中发挥自己的优势。 5. 信息素养的提高：通过学习 Scratch 编程，学生能够更好地理解信息技术在生活中的应用，提高自己的信息素养。学生能够认识到编程的重要性，以及如何利用编程技术解决实际问题。 6. 自主学习和探究能力的培养：学生在课后作业中，需要独立完成动画制作，这有助于培养学生的自主学习能力。学生在遇到问题时，能够通过查阅资料、请教同学等方式进行探究，提高自己的学习能力。 7. 情感态度和价值观的塑造：在制作动画的过程中，学生能够体验到编程的乐趣，培养对信息技术的兴趣。同时，学生在面对挑战和困难时，能够展现出坚持不懈的精神，这有助于塑造积极向上的情感态度和价值观。 	
作业 布置 与反 馈	<p>作业布置：</p> <ol style="list-style-type: none"> 1. 完成课后练习：学生需根据教材中的 Scratch 编程案例，自行完成至少一个动画作品，如“小猫追逐鼠标”或“小猫跳跃”。要求作品包含基本的编程逻辑，如事件处理、控制角色动作等。 2. 创意设计：鼓励学生发挥创意，设计一个独特的动画角色，并编写一个至少包含两个场景的动画故事。学生可以使用 Scratch 软件中的各种编程块，实现角色的运动、对话和场景切换。 3. 技能提升：学生可以选择学习 Scratch 软件中的高级功能，如变量、循环、函数等，并将其应用到自己的动画作品中，以提高作品的技术含量。 <p>作业反馈：</p> <ol style="list-style-type: none"> 1. 及时批改：教师应在学生提交作业后的第二天内完成批改，确保学生能够及时收到反馈。 2. 详细评价：教师在批改作业时，应详细记录学生的作品完成情况，包括编程的正确性、创意的独到之处、作品的完整性和技术难度等。 3. 存在问题指出：对于学生在作业中存在的问题，教师应具体指出，如编程逻辑错误、代码冗余、设计不合理等。 4. 改进建议：针对学生作业中的不足，教师应给出具体的改进建议，如如 	

	何优化代码、如何改进设计、如何提高动画效果等。	
--	-------------------------	--

5.

	<p>集体反馈：教师可以将一些具有代表性的作业进行展示和讨论，让学生共同学习，提高整体水平。</p> <p>6. 个别辅导：对于作业中存在严重问题的学生，教师应进行个别辅导，帮助学生理解知识点，提高编程能力。</p>	
课后作业	<p>1. 编程任务：设计一个“小猫捉老鼠”的动画，其中小猫和老鼠都需要通过编程控制其动作。小猫能够跳跃追捕老鼠，老鼠可以左右移动躲避。要求使用 Scratch 编程语言完成。</p> <p>答案示例：</p> <ul style="list-style-type: none"> - 小猫角色： - 当接收到“跳跃”事件时，将小猫的 Y 坐标减少一个固定值。 - 当小猫靠近老鼠时，播放一个“追捕”的声音。 - 老鼠角色： - 当接收到“移动”事件时，将老鼠的 X 坐标增加或减少一个固定值。 - 当老鼠离开屏幕边缘时，将其 X 或 Y 坐标重置到屏幕中心。 <p>2. 逻辑挑战：编写一个 Scratch 脚本，使得一个角色（例如一个小鸟）在屏幕上随机移动，并在接触到地面时发出“啾啾”的声音。</p> <p>答案示例：</p> <ul style="list-style-type: none"> - 角色移动： - 使用“随机位置”编程块来使角色在屏幕上随机移动。 - 触地检测： - 使用“如果…那么”条件编程块，检查角色的 Y 坐标是否低于地面坐标，如果是，则执行“播放声音”事件。 <p>3. 创意扩展：制作一个“生日派对”动画，包括一个装饰蛋糕和蜡烛的角色。当角色靠近蛋糕时，蜡烛会依次点亮。</p> <p>答案示例：</p> <ul style="list-style-type: none"> - 蛋糕和蜡烛角色： - 蛋糕角色在初始状态时没有蜡烛。 - 当角色靠近蛋糕时，依次点亮蜡烛，每个蜡烛使用“等待”编程块来延迟点亮时间。 <p>4. 变量应用：编写一个 Scratch 脚本，使一个角色在屏幕上移动时，其速度随着时间逐渐增加。</p> <p>答案示例：</p> <ul style="list-style-type: none"> - 调整速度： - 使用变量来存储角色的速度。 - 在“移动”事件中，每移动一步，就增加变量的值来提高速度。 - 使用“设置变量”编程块来初始化变量。 <p>5. 循环结构：设计一个“打地鼠”游戏，玩家通过点击屏幕上的地鼠来得分。每个地鼠出现后都会有一定的持续时间，然后消失。</p> <p>答案示例：</p> <ul style="list-style-type: none"> - 地鼠生成： - 使用“等待”编程块来控制地鼠出现的间隔时间。 - 当地鼠出现时，显示在屏幕上的某个位置。 - 地鼠消失： - 在地鼠上使用“等待”编程块来控制地鼠的持续时间。 - 当时间到时，使用“隐藏”编程块来使地鼠从屏幕上消失。 	

	-	
--	---	--

	<p>玩家互动：</p> <ul style="list-style-type: none"> - 使用“鼠标点击”事件来检测玩家是否点击了地鼠。 - 当点击发生时，使用“增加变量”编程块来增加得分。 	
<p>教学反思与总结</p>	<p>今天这节课，我带学生们走进了 Scratch 的世界，我们一起认识了 Scratch，也尝试制作了一个可爱的小猫动画。回顾一下，我觉得有几个方面做得不错，也有一些地方可以改进。</p> <p>首先，我觉得今天的教学方法还是挺有效的。我采用了讲授、案例研究和项目导向学习相结合的方式，这样既能让学生们系统地了解 Scratch 的基本操作，又能通过实际操作来巩固所学知识。特别是项目导向学习，让学生们分组合作，不仅提高了他们的团队协作能力，也激发了他们的创造力。</p> <p>不过，我也发现了一些问题。比如，在讲解 Scratch 的基本操作时，我发现有些学生反应比较慢，可能是因为他们对计算机操作不太熟悉。这让我意识到，在今后的教学中，我需要更加关注学生的个体差异，对于基础较薄弱的学生，可能需要更多的个别辅导。</p> <p>在教学过程中，我还发现了一些亮点。比如，当学生们看到自己制作的动画作品时，那种兴奋和满足的表情让我感到非常欣慰。这说明，Scratch 这种图形化编程工具确实能够激发学生的学习兴趣，让他们在轻松愉快的氛围中学习编程。</p> <p>当然，也有不足之处。比如，在学生分组合作时，我发现有些小组的讨论不够积极，可能是因为他们对编程不够熟悉，不知道如何开始。这让我想到，在今后的教学中，我可以在课前提供一些讨论指南，帮助学生更好地进行小组合作。</p> <p>针对教学中存在的问题，我提出以下改进措施和建议：</p> <ol style="list-style-type: none"> 1. 对于基础较薄弱的学生，提供更多的个别辅导，帮助他们掌握计算机操作和编程基础知识。 2. 在课前提供讨论指南，帮助学生更好地进行小组合作，提高讨论效率。 3. 在教学过程中，多鼓励学生提出问题，培养他们的探究精神和解决问题的能力。 4. 定期举办 Scratch 编程比赛或展示活动，激发学生的学习兴趣，提高他们的编程技能。 <p>我相信，通过不断的反思和改进，我能够在今后的教学中做得更好，帮助学生们在信息技术的道路上越走越远。</p>	
<p>板书设计</p>	<p>① Scratch 软件简介</p> <ul style="list-style-type: none"> - 图形化编程语言 - 用户界面 - 编程块类型 <p>② Scratch 基本操作</p> <ul style="list-style-type: none"> - 界面布局 - 编程块使用 - 事件和动作 <p>③ 编程案例讲解</p> <ul style="list-style-type: none"> - 小猫眨眼动画 - 编程逻辑 - 事件处理 	

	变量和控制结构 - 变量的定义和作用 - 循环结构 - 控制结构 ⑤ 项目导向学习 - 分组合作 - 创意设计 - 作品展示 ⑥ 课后作业提示 - 完成课后练习 - 创意扩展 - 技能提升	
--	---	--

第一单元 走进 Scratch 世界第二课 巧解数学题—变量的应用

主备人	
备课成员	
设计意图	本节课旨在通过 Scratch 编程软件，让学生了解和掌握变量的概念和应用，培养学生的逻辑思维能力和编程技能。通过设计“巧解数学题”的编程任务，让学生在解决实际问题的过程中，学会运用变量进行数据存储和计算，提高学生的信息素养和创新能力。同时，本节课与课本内容紧密相连，帮助学生巩固所学知识，提高编程兴趣，为后续课程学习打下坚实基础。
核心素养目标分析	本节课旨在培养学生的信息意识、计算思维、数字化学习与创新等核心素养。首先，通过引入 Scratch 编程，激发学生对信息技术的兴趣，培养他们的信息意识。其次，通过变量的应用，引导学生理解抽象概念，提升计算思维能力。再次，通过设计编程任务，鼓励学生进行创新实践，培养他们的数字化学习与创新精神。最后，通过小组合作和问题解决，促进学生合作学习能力的提升，培养他们的社会责任感和团队合作精神。
教学难点与重点	1. 教学重点 - 理解变量的概念：本节课的核心内容是让学生理解变量在编程中的作用，即作为存储数据的容器。重点在于通过实例让学生明白变量可以存储数值，并在程序中随时更新和引用。 - 变量的声明与赋值：强调学生能够正确声明变量并为其赋值，例如在 Scratch 中创建变量并使用“设置变量到”操作符来初始化变量值。 2. 教学难点 - 变量的作用域：难点在于理解变量在不同作用域（如全局变量和局部变量）中的使用规则，以及它们对程序执行的影响。 - 变量的类型和转换：学生可能难以掌握不同类型（如数字、字符串）的变量使用和转换，特别是在进行复杂计算时。 - 变量的更新与引用：难点在于让学生理解如何在程序中更新变量值，并正确引用变量进行计算，例如在 Scratch 中通过“改变变量”积木来实现。

	Xxx
--	-----

学具准备					
课型	新授课	教法学法	讲授法	课时	第一课时
步骤	师生互动设计			二次备课	
教学资源	<ul style="list-style-type: none"> - 软硬件资源：Scratch 编程软件、计算机设备、投影仪 - 课程平台：学校内部教学平台或 Scratch 在线社区 - 信息化资源：Scratch 官方教程、编程案例库、数学题库 - 教学手段：PPT 演示、实物教具（如计算器）、交互式编程软件 				
教学过程	<p>1. 课前自主探索教师活动：</p> <ul style="list-style-type: none"> - 学生利用学校提供的 Scratch 软件进行自主探索，尝试创建简单的程序，如移动角色或改变角色外观。 - 教师提前准备一系列引导性问题，引导学生思考变量的作用，如“如何让角色在舞台上移动更远的距离？” - 学生通过搜索和实验，初步了解变量的概念和基础用法。 <p>2. 课中强化技能教师活动：</p> <ul style="list-style-type: none"> - 教师通过 PPT 展示变量在 Scratch 中的具体操作，如创建变量、设置变量值、改变变量等。 - 学生跟随教师的演示，动手实践，在 Scratch 中创建一个简单的数学问题求解程序，如计算两数之和。 - 教师指导学生如何声明一个变量来存储计算结果，并展示如何通过“如果-否则”语句使用变量进行比较和判断。 <p>3. 课后拓展应用教师活动：</p> <ul style="list-style-type: none"> - 教师布置课后作业，要求学生设计一个更复杂的数学题求解程序，如计算多项式。 - 学生利用所学知识，尝试在 Scratch 中实现更高级的变量应用，如使用循环结构处理重复计算。 - 教师收集学生的程序作品，进行在线展示和互评，鼓励学生相互学习和改进。 				
学生学习效果	<p>学生学习效果主要体现在以下几个方面：</p> <ol style="list-style-type: none"> 1. 理解变量的概念和应用：通过本节课的学习，学生能够理解变量作为数据存储和计算工具的重要性，能够区分变量的声明、赋值、更新和引用等基本操作。学生在课后能够运用变量解决简单的数学问题，如计算两数之和、比较大小等，这表明他们对变量概念有了深刻的理解。 2. 编程技能的提升：学生在 Scratch 编程软件中实践变量的应用，通过编写程序来处理数学问题，这不仅提高了他们的编程技能，还增强了他们的逻辑思维和问题解决能力。学生在设计程序的过程中，学会了如何规划程序结构，如何使用循环和条件语句来处理更复杂的计算。 3. 创新能力的培养：本节课通过让学生自主设计和实现数学题求解程序，激发了学生的创新思维。学生在解决实际问题的过程中，能够提出不同的解决方案，并尝试优化程序，这有助于培养他们的创新精神和实践能力。 4. 				

	<p>团队合作能力的加强：在课后拓展应用环节，学生被鼓励以小组形式合作完成更复杂的编程任务。通过讨论、分工和协作，学生学会了如何在团队中有效沟通，共同解决问题，这有助于提高他们的团队合作能力。</p> <p>5. 学习兴趣和激发：通过使用 Scratch 这样的互动式编程工具，学生能够直观地看到自己的程序运行效果，这极大地激发了他们的学习兴趣。学生在学习过程中体验到了编程的乐趣，愿意主动探索和学习更多的编程知识。</p> <p>6. 数学知识的应用：本节课的教学内容与数学课程紧密相关，学生在学习变量的同时，复习和巩固了数学中的基础概念，如加法、减法、乘法、除法等。这种跨学科的学习方式有助于学生将数学知识应用于实际情境中。</p> <p>7. 自主学习能力的培养：学生在课前自主探索阶段，通过自学和实验，初步接触了变量的概念。这种自主学习的方式不仅提高了学生的自主学习能力，还培养了他们的探究精神和终身学习的习惯。</p>	
<p>典型例题讲解</p>	<p>1. 例题：编写一个 Scratch 程序，让角色从位置 (0, 0) 移动到位置 (100, 100)，每次移动 10 个单位。 答案：首先，创建一个变量名为“移动距离”，初始化为 0。然后，使用“当绿旗被点击”事件作为循环的开始，使用“改变变量移动距离”积木增加移动距离的值，每次增加 10。接着，使用“移动 (移动距离) 步”积木让角色移动相应的步数。最后，在循环中添加条件判断，当移动距离达到 100 时，停止循环。</p> <p>2. 例题：编写一个 Scratch 程序，计算并显示两个数的平均值。 答案：首先，创建两个变量，分别命名为“数 1”和“数 2”，并初始化为任意数值。然后，创建一个新变量，命名为“平均值”，用于存储计算结果。使用“设置变量平均值到” (数 1 + 数 2) / 2 的操作来计算平均值。最后，使用“显示变量平均值”积木来显示计算结果。</p> <p>3. 例题：编写一个 Scratch 程序，让角色在舞台上随机移动到不同的位置。 答案：首先，创建一个变量名为“随机数”，用于存储随机位置。然后，使用“当绿旗被点击”事件作为循环的开始，使用“设置变量随机数到”随机数生成器生成的数值的操作来获取随机数。接着，使用“移动到 X: (随机数) Y: (随机数)”积木让角色移动到随机位置。最后，在循环中添加条件判断，以防止角色重复移动到相同的位置。</p> <p>4. 例题：编写一个 Scratch 程序，计算并显示一个数的平方。 答案：首先，创建一个变量名为“数”，并初始化为任意数值。然后，创建一个新变量，命名为“平方”，用于存储计算结果。使用“设置变量平方到”数 * 数的操作来计算平方。最后，使用“显示变量平方”积木来显示计算结果。</p> <p>5. 例题：编写一个 Scratch 程序，让角色根据用户输入的数值进行相应的动作。 答案：首先，创建一个变量名为“用户输入”，用于存储用户输入的数值。然后，使用“当绿旗被点击”事件作为循环的开始，使用“输入”积木提示用户输入数值，并将输入值存储在“用户输入”变量中。接着，根据用户输入的数值，使用条件判断语句 (如“如果... 那么... 否则...”) 来决定角色应该执行的动作，如移动、改变外观等。最后，使用“显示变量用户输入”积木来显示用户输入的数值。</p>	

板书 设计	① 变量的概念 - 变量是用于存储数据的容器 -	
----------	--------------------------------	--

	<p>变量可以存储数值、文字等数据</p> <ul style="list-style-type: none"> - 变量名：用于标识变量的名称，如“计数器” <p>② 变量的操作</p> <ul style="list-style-type: none"> - 声明变量：创建一个新的变量，如“设置变量计数器到0” - 赋值：给变量赋值，如“改变变量计数器” - 引用：在程序中使用变量，如“显示变量计数器” <p>③ 变量的应用</p> <ul style="list-style-type: none"> - 数据存储：使用变量存储程序运行中的数据 - 数据计算：使用变量进行数学计算 - 数据比较：使用变量进行条件判断 <p>④ Scratch 中变量的使用</p> <ul style="list-style-type: none"> - 创建变量：在 Scratch 中，通过“变量”菜单创建新的变量 - 显示变量：使用“显示变量”积木来显示变量值 - 改变变量：使用“改变变量”积木来更新变量值 	
<p>反思改进措施</p>	<p>反思改进措施（一）教学特色创新</p> <ol style="list-style-type: none"> 1. 互动式教学：尝试在课堂上引入更多互动环节，比如小组讨论、编程竞赛，让学生在合作中学习，提高他们的团队协作能力。 2. 实践导向：加强实践环节，让学生通过解决实际问题来学习变量应用，例如设计一个简单的游戏或动画，让学生在实践中加深理解。 <p>反思改进措施（二）存在主要问题</p> <ol style="list-style-type: none"> 1. 学生参与度：部分学生在编程过程中显得比较被动，可能是因为对编程不够感兴趣或者缺乏足够的自信心。 2. 知识掌握不牢固：有些学生在变量概念的理解上存在困难，这可能是由于对抽象概念的理解不够深入。 3. 教学评价单一：目前的评价方式主要依赖于学生的编程作品，缺乏对学生学习过程的持续跟踪和评价。 <p>反思改进措施（三）</p> <ol style="list-style-type: none"> 1. 提高学生参与度：可以通过设置有趣的项目和挑战，激发学生的学习兴趣。同时，定期进行小测验或编程小比赛，让学生在竞争中学习。 2. 加强基础知识教学：针对变量概念的理解，可以采用更多实例和类比，帮助学生建立对抽象概念的理解。此外，可以设计一些基础练习，让学生反复练习变量的声明、赋值和引用。 3. 丰富教学评价方式：除了编程作品，可以引入过程性评价，如课堂表现、小组合作情况等，全面评估学生的学习情况。同时，可以鼓励学生进行自我评价和同伴评价，提高他们的反思能力。 	
<p>教学评价与反馈</p>	<ol style="list-style-type: none"> 1. 课堂表现：通过观察学生的课堂参与度、提问回答情况以及编程过程中的表现，评价学生对变量概念的理解和应用能力。例如，记录学生是否能够独立完成变量的声明、赋值和引用，以及在编程过程中是否能够正确运用变量解决问题。 2. 小组讨论成果展示：组织学生进行小组讨论，鼓励他们在讨论中提出问题、分享想法和解决问题。评价小组讨论的成果，包括讨论的深度、广度和创新性，以及学生之间的合作和沟通能力。 3. 	

	<p>随堂测试：设计简单的编程题目，让学生在课堂上进行实际操作，检验他们对变量应用的理解程度。例如，给出一个简单的数学问题，要求学生编写程序来求解，并观察他们是否能够正确使用变量来存储数据和计算结果。</p> <p>4. 课后作业反馈：收集学生的课后作业，评估他们在家庭作业中的表现。重点关注学生是否能够独立完成编程任务，是否能够理解并应用变量解决实际问题，以及是否能够对程序进行优化。</p> <p>5. 教师评价与反馈：针对学生的编程作品，给出具体的评价和建议。例如，评价学生在程序设计中的逻辑清晰度、代码的整洁性和程序的运行效果。同时，提供个性化的反馈，帮助学生识别自己的优点和需要改进的地方。</p> <ul style="list-style-type: none"> - 评价学生的编程逻辑：检查学生的程序是否能够按照预期的方式运行，是否能够正确处理输入数据并给出正确的结果。 - 评价学生的编程风格：关注学生的代码是否简洁、易读，是否遵循良好的编程习惯。 - 评价学生的创新性：鼓励学生在程序中尝试不同的解决方案，评价他们是否能够提出有创意的编程思路。 	
--	--	--

第一单元 走进 Scratch 世界第三课 小猫报站名——链表的应用

主备人	
备课成员	
设计思路	<p>本节课以“小猫报站名——链表的应用”为主题，结合大连理工版七年级下册信息技术教材，旨在让学生通过实际操作，深入理解链表的概念和应用。课程设计围绕以下思路展开：首先，通过引入小猫报站名的场景，激发学生的学习兴趣；其次，引导学生逐步掌握链表的基本概念和操作方法；最后，通过具体实例，让学生学会运用链表解决实际问题。课程内容与课本紧密相连，注重培养学生的动手实践能力和创新思维。</p>
核心素养目标	<ol style="list-style-type: none"> 1. 技术应用：培养学生运用 Scratch 编程语言解决实际问题的能力，理解链表在编程中的应用价值。 2. 问题解决：通过设计小猫报站名程序，提升学生分析问题和解决问题的能力，学会运用算法思维。 3. 创新意识：鼓励学生在编程过程中进行创新尝试，培养学生的创新意识和实践能力。 4. 合作学习：引导学生进行小组合作，共同完成编程任务，提高沟通协作和团队协作能力。 5. 情感态度：激发学生对编程的兴趣，培养严谨的科学态度和勇于探索的精神。
教学难点与重点	<ol style="list-style-type: none"> 1. 教学重点 <ul style="list-style-type: none"> - 核心内容：链表的基本概念与操作，包括创建链表、插入节点、删除节点和遍历链表。 - 举例解释：学生需要理解链表是如何通过节点之间的链接实现的，并能通过 Scratch 编程语言中的列表功能来模拟链表的行为。 2. 教学难点 <ul style="list-style-type: none"> - 难点内容：链表节点的插入与删除操作。

<p>- 举例解释：学生在操作过程中可能会遇到如何正确插入和删除节点的问题，尤其是在链表中间插入节点时，需要保持前后节点的正确链接。此外，删除节点时还需注意释放内存以避免内存泄漏。</p>
--

学具准备	多媒体				
课型	新授课	教法学法	讲授法	课时	第一课时
步骤	师生互动设计				二次备课
教学资源准备	<p>1. 教材：确保每位学生都有大连理工版七年级下册信息技术教材，以便于跟随教材内容学习。</p> <p>2. 辅助材料：准备与链表概念相关的图片、图表，以及 Scratch 编程环境的截图，帮助学生直观理解链表结构。</p> <p>3. 实验器材：准备一台或多台联网的计算机，确保 Scratch 软件安装正常，以便学生进行编程实践。</p> <p>4. 教室布置：布置教室环境，设置分组讨论区，每个小组配备一台计算机，方便学生协作学习和操作。</p>				
教学流程	<p>1. 导入新课</p> <ul style="list-style-type: none"> - 详细内容：首先，通过展示一段 Scratch 编程制作的动画，引发学生对编程的兴趣。然后，提出问题：“如何让动画中的角色能够记住并报出多个站名？”引导学生思考记忆多个数据的方法，自然过渡到本节课的主题——链表的应用。 <p>2. 新课讲授</p> <ul style="list-style-type: none"> - 详细内容 1：讲解链表的基本概念，通过图片和图表展示链表的结构，让学生理解节点和链接的关系。 - 详细内容 2：演示如何使用 Scratch 编程语言创建链表，包括创建节点、插入节点和删除节点等操作。 <p>3. 实践活动</p> <ul style="list-style-type: none"> - 详细内容 1：学生分组，每组一台计算机，根据教材内容，尝试在 Scratch 中创建一个简单的链表。 - 详细内容 2：引导学生尝试在链表中插入新的节点，并展示如何正确地链接前后节点。 - 详细内容 3：让学生尝试删除链表中的节点，并讨论如何处理删除操作中的边界情况。 <p>4. 学生小组讨论</p> <ul style="list-style-type: none"> - 举例回答 1：讨论如何处理链表为空的情况，例如在插入或删除节点时如何判断链表是否为空。 - 举例回答 2：讨论在删除节点时如何防止内存泄漏，例如在删除节点后释放该节点所占用的内存。 - 举例回答 3：讨论如何优化链表的查找操作，例如通过链表排序或使用哈希表等方法提高查找效率。 <p>5. 总结回顾</p> <ul style="list-style-type: none"> - 内容：对本节课的学习内容进行总结，强调链表的基本概念和操作方法。通过提问和解答的方式，帮助学生巩固重难点知识，例如： - 提问：“链表与数组有什么区别？” - 				

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/928017044057007051>