

2024-2025 学年初中信息技术（信息科技）第四册 青岛版（2019）教学设计合集

目录

一、第一单元 开源硬件——智能机器人

- 1.1 第1课 初识BBC micro:bit
- 1.2 第2课 高效的循环
- 1.3 第3课 灵活的变量
- 1.4 第4课 智慧的选择
- 1.5 第5课 周到的预案
- 1.6 第6课 强大的通信
- 1.7 第7课 丰富的扩展
- 1.8 第8课 制作无线遥控车
- 1.9 本单元复习与测试

二、第二单元 物联网

- 2.1 第9课 初识物联网
- 2.2 第10课 物联网让城市更智能
- 2.3 第11课 智能家居
- 2.4 本单元复习与测试

三、第三单元 创客训练营——Scratch与硬件

- 3.1 第12课 初识Scratch
- 3.2 第13课 智能避障车
- 3.3 第14课 智能干手器
- 3.4 第15课 智能红外遥控风扇
- 3.5 本单元复习与测试

第一单元 开源硬件——智能机器人第1课 初识BBC micro:bit

主备人	
-----	--

备课成员				
课程基本信息	<p>1. 课程名称：初中信息技术（信息科技）第四册青岛版（2019）第一单元 开源硬件—智能机器人 第1课 初识 BBC micro:bit</p> <p>2. 教学年级和班级：八年级1班</p> <p>3. 授课时间：2022年10月18日 上午第二节课</p> <p>4. 教学时数：1课时</p>			
核心素养目标	<p>1. 技术意识：培养学生对开源硬件的认识，激发学生对智能机器人技术的兴趣。</p> <p>2. 创新思维：通过实践操作，培养学生的创新意识和解决问题的能力。</p> <p>3. 信息处理：使学生掌握 BBC micro:bit 的基本操作，提高信息处理和编程技能。</p> <p>4. 数字化学习与创新：培养学生利用数字化工具进行学习和创新的能力。</p> <p>5. 信息伦理与安全：引导学生正确使用开源硬件，树立信息伦理和安全意识。</p>			
教学重难点	<p>1. 教学重点：</p> <ul style="list-style-type: none"> - BBC micro:bit 的基本硬件结构：重点讲解 BBC micro:bit 的模块组成，如 LED 矩阵、按钮、加速度计等，以及它们的功能和连接方式。 - 基本编程环境：介绍 BBC micro:bit 的编程环境，包括代码编辑器、编程语言（如 Scratch 或 Python）的基本使用方法。 <p>2. 教学难点：</p> <ul style="list-style-type: none"> - 编程逻辑的理解：对于初学者来说，理解编程逻辑和流程图是难点。例如，如何通过编程让 BBC micro:bit 的 LED 矩阵显示特定的图案或响应按钮事件。 - 软件调试：学生可能难以理解软件调试的概念，包括如何识别和修正代码中的错误。例如，当代码运行不正常时，如何使用调试工具找到问题所在。 - 硬件故障排除：学生需要学会如何识别和处理 BBC micro:bit 的硬件故障，如 LED 灯不亮、按钮失灵等。 			
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
教学资源准备	<p>1. 教材：确保每位学生都有一本《初中信息技术（信息科技）第四册青岛版（2019）》教材，以便于跟随课堂内容进行学习。</p> <p>2. 辅助材料：准备 BBC micro:bit 的相关图片、图表和操作视频，帮助学生直观理解硬件结构和编程操作。</p> <p>3. 实验器材：准备足够的 BBC micro:bit 开发板、USB 线、电源适配器等实验器材，确保每位学生都能进行实际操作。</p> <p>4. 教室布置：布置教室，设置分组讨论区，每个小组配备实验操作台，方便学生进行实践操作和交流。</p>			
教学过程	<p>一、导入新课</p> <p>同学们，今天我们要一起探索一个充满活力的领域—开源硬件。在我们生活中，有很多智能设备都是通过编程和硬件结合来实现的。今天，我们就来认识一种非常有趣的开源硬件—BBC</p>			

micro:bit。

(教师引导学生回顾上节课内容，提出问题：“大家还记得我们之前学过的什么设备吗？它们是如何工作的？”)

二、新课导入

1. BBC micro:bit 介绍

(教师讲解 BBC micro:bit 的起源、特点以及应用领域，并展示 BBC micro:bit 的实物)

同学们，BBC micro:bit 是一种专为青少年设计的开源硬件，它具有丰富的功能，如 LED 矩阵、按钮、加速度计等。这些功能让 BBC micro:bit 可以在很多场合发挥作用，比如游戏、艺术创作、科学实验等。

2. BBC micro:bit 硬件结构

(教师讲解 BBC micro:bit 的硬件结构，包括主要模块和接口)

同学们，BBC micro:bit 主要由主控芯片、LED 矩阵、按钮、加速度计等模块组成。这些模块通过引脚连接，可以实现各种功能。例如，LED 矩阵可以显示图案或文字，按钮可以检测按键事件，加速度计可以检测运动状态。

3. BBC micro:bit 编程环境

(教师讲解 BBC micro:bit 的编程环境，包括 Scratch 和 Python 两种编程语言)

同学们，BBC micro:bit 支持 Scratch 和 Python 两种编程语言。Scratch 是一种图形化编程语言，适合初学者；Python 是一种文本编程语言，功能强大。我们可以根据自己的需求选择合适的编程语言进行编程。

三、课堂实践

1. LED 矩阵编程

(教师示范如何使用 Scratch 编程让 LED 矩阵显示“Hello World!”)

同学们，接下来我们来实践一下。首先，我们需要将 BBC micro:bit 连接到电脑，然后打开 Scratch 编程环境。在这里，我们可以通过拖拽代码块来编写程序。比如，我们可以编写一个程序，让 LED 矩阵显示“Hello World!”。

2. 按钮编程

(教师讲解如何使用 Scratch 编程让 BBC micro:bit 的按钮控制 LED 矩阵)

同学们，接下来我们再学习如何使用 Scratch 编程让 BBC micro:bit 的按钮控制 LED 矩阵。比如，当按下按钮时，LED 矩阵显示一个笑脸图案，松开按钮时显示一个哭脸图案。

3. 加速度计编程

(教师讲解如何使用 Python 编程让 BBC micro:bit 的加速度计检测运动状态)

同学们，今天我们还将学习如何使用 Python 编程让 BBC micro:bit 的加速度计检测运动状态。比如，当检测到运动时，LED 矩阵显示一个箭头图案，表示运动方向。

四、课堂小结

(教师总结本节课内容，回顾重点)

五、课后作业

1. 尝试用 Scratch 或 Python 编程，让 BBC micro:bit 的 LED 矩阵显示一个有趣的图案。

2. 尝试使用按钮控制 BBC micro:bit 的 LED 矩阵，实现不同的功能。

	3. 尝试使用加速度计检测 BBC	
--	-------------------	--

	<p>micro:bit 的运动状态，并编写相应的程序。 (教师布置课后作业，并提醒学生注意安全)</p>	
<p>知识 点梳 理</p>	<ol style="list-style-type: none"> 1. BBC micro:bit 概述 <ul style="list-style-type: none"> - BBC micro:bit 的起源和发展 - BBC micro:bit 的设计理念和应用领域 - BBC micro:bit 的特点和优势 2. BBC micro:bit 硬件结构 <ul style="list-style-type: none"> - 主控芯片：介绍 BBC micro:bit 所使用的微控制器，如 ARM Cortex-M0+ - LED 矩阵：介绍 LED 矩阵的组成、功能以及如何通过编程控制显示 - 按钮和触摸传感器：讲解按钮和触摸传感器的功能、连接方式以及如何读取输入 - 加速度计：介绍加速度计的原理、功能以及如何获取运动数据 - 磁力传感器：讲解磁力传感器的原理、功能以及如何检测方向变化 - 无线模块：介绍 BBC micro:bit 的无线通信功能，如蓝牙和 Wi-Fi - 音频输出：讲解 BBC micro:bit 的音频输出功能，如蜂鸣器和音乐播放 - 电源管理：介绍 BBC micro:bit 的电源管理机制，如电池续航和充电方式 3. BBC micro:bit 编程环境 <ul style="list-style-type: none"> - Scratch 编程：介绍 Scratch 编程语言的特点、基本语法和编程环境 - Python 编程：介绍 Python 编程语言的特点、基本语法和编程环境 - 编程工具：介绍 BBC micro:bit 的编程工具，如 Scratch 3.0 和 BBC micro:bit Python Editor 4. BBC micro:bit 编程实践 <ul style="list-style-type: none"> - LED 矩阵编程：讲解如何通过编程控制 LED 矩阵显示图案、文字和动画 - 按钮编程：讲解如何通过编程控制按钮的响应，如读取按键状态和执行动作 - 加速度计编程：讲解如何通过编程获取加速度计数据，如检测运动方向和速度 - 磁力传感器编程：讲解如何通过编程获取磁力传感器数据，如检测方向变化 - 无线通信编程：讲解如何通过编程实现 BBC micro:bit 之间的无线通信 - 音频编程：讲解如何通过编程控制 BBC micro:bit 的音频输出，如播放音乐和声音效果 5. BBC micro:bit 项目案例 <ul style="list-style-type: none"> - 基础项目：介绍一些简单的 BBC micro:bit 项目，如制作一个 LED 矩阵时钟、温度计等 - 进阶项目：介绍一些较复杂的项目，如制作一个智能机器人、智能家居系统等 - 创意项目：鼓励学生发挥创意，设计自己的 BBC micro:bit 项目 6. BBC micro:bit 安全与伦理 <ul style="list-style-type: none"> - 数据保护：讲解如何保护 BBC micro:bit 上的数据，如加密和备份 - 隐私保护：讲解如何保护用户隐私，避免泄露个人信息 - 软件安全：讲解如何避免软件漏洞和恶意攻击，确保 BBC micro:bit 的安全性 - 伦理道德：讲解在设计和使用 BBC 	

	<p>micro:bit 过程中应遵循的伦理道德规范</p> <p>7. BBC micro:bit 资源与支持</p> <ul style="list-style-type: none"> - 官方网站：介绍 BBC micro:bit 官方网站，提供编程教程、项目案例和社区支持 - 开源社区：介绍 BBC micro:bit 的开源社区，如 GitHub，提供丰富的编程资源和项目案例 - 教育资源：介绍 BBC micro:bit 的教育资源，如教学视频、教材和教学活动 - 技术支持：介绍 BBC micro:bit 的技术支持，如官方客服和社区论坛 	
<p>板书设计</p>	<ul style="list-style-type: none"> ① BBC micro:bit 概述 <ul style="list-style-type: none"> - BBC micro:bit 简介 - BBC micro:bit 特点 - BBC micro:bit 应用领域 ② BBC micro:bit 硬件结构 <ul style="list-style-type: none"> - 主控芯片 - LED 矩阵 - 按钮和触摸传感器 - 加速度计 - 磁力传感器 - 无线模块 - 音频输出 - 电源管理 ③ BBC micro:bit 编程环境 <ul style="list-style-type: none"> - Scratch 编程 - Python 编程 - 编程工具介绍 ④ BBC micro:bit 编程实践 <ul style="list-style-type: none"> - LED 矩阵编程 - 按钮编程 - 加速度计编程 - 磁力传感器编程 - 无线通信编程 - 音频编程 ⑤ BBC micro:bit 项目案例 <ul style="list-style-type: none"> - 基础项目 - 进阶项目 - 创意项目 ⑥ BBC micro:bit 安全与伦理 <ul style="list-style-type: none"> - 数据保护 - 隐私保护 - 软件安全 - 伦理道德 ⑦ BBC micro:bit 资源与支持 <ul style="list-style-type: none"> - 官方网站 - 开源社区 	

	-	
--	---	--

	<p>教育资源 - 技术支持</p>	
<p>教学反思与总结</p>	<p>今天的课，我觉得整体来说还是比较顺利的。首先，我注意到同学们对 BBC micro:bit 这个主题表现出了很高的兴趣，这在课堂上的互动和参与度上体现得非常明显。大家在编程实践环节都很积极，这让我感到非常欣慰。</p> <p>在教学过程中，我尽量让同学们通过动手操作来学习，比如让他们自己尝试用 Scratch 编程控制 LED 矩阵。我觉得这种方法挺有效的，因为它不仅让同学们学会了如何编程，还提高了他们的实践能力。</p> <p>不过，我也发现了一些需要改进的地方。比如，在讲解编程逻辑时，我发现有些同学理解起来有些吃力。这可能是因为他们对编程基础不够熟悉。因此，我考虑在今后的教学中，可以适当增加一些编程基础知识的讲解，帮助他们更好地理解编程逻辑。</p> <p>另外，课堂上的时间管理也是我需要改进的地方。由于同学们对 BBC micro:bit 的兴趣很高，所以在实践环节投入了较多的时间，导致课堂内容有些赶。今后，我会更加注意课堂节奏的把握，确保所有内容都能在规定时间内完成。</p> <p>至于教学效果，我觉得学生们在知识、技能和情感态度方面都有所收获。他们不仅学会了 BBC micro:bit 的基本操作，还体验到了编程的乐趣。在情感态度方面，我看到了同学们对科学技术的热情和对学习的积极性。</p> <p>针对今天的教学，我提出以下几点改进措施和建议：</p> <ol style="list-style-type: none"> 1. 加强编程基础知识的教学，为后续的编程实践打下坚实的基础。 2. 优化课堂节奏，确保教学内容和实践活动都能在规定时间内完成。 3. 鼓励同学们发挥创意，设计自己的 BBC micro:bit 项目，提高他们的创新能力和实践能力。 4. 定期进行教学反思，不断调整和改进教学方法，以提高教学效果。 	
<p>教学评价与反馈</p>	<p>1. 课堂表现： 同学们在今天的课堂上表现出了很高的积极性和参与度。大家对于 BBC micro:bit 的硬件结构和编程环境都表现出了浓厚的兴趣，课堂上的互动问答环节中，很多同学都能积极回答问题，这让我感到非常高兴。</p> <p>2. 小组讨论成果展示： 在小组讨论环节，同学们能够互相协作，共同探讨如何利用 BBC micro:bit 完成特定的编程任务。每个小组都展示了自己的项目成果，比如有的小组制作了一个简单的 LED 矩阵时钟，有的小组则设计了一个简单的游戏。这些成果展示了同学们的团队合作能力和创新思维。</p> <p>3. 随堂测试： 为了检验同学们对今天所学内容的掌握情况，我进行了一次随堂测试。测试结果显示，大部分同学能够正确回答关于 BBC micro:bit 硬件结构和编程环境的问题，但也有少数同学在编程逻辑的理解上存在困难。这提示我在今后的教学中加强对编程逻辑的讲解和练习。</p> <p>4. 学生自评与互评： 在课程结束后，我让同学们进行了自评和互评。同学们对自己的学习效果进行了反思，同时也对同伴的表现给予了评价。这种自我评价和同伴评价的方式有助于同学们认识到自己的优势和不足，促进他们的自我提升。</p> <p>5.</p>	

	<p>教师评价与反馈：</p> <p>针对今天的课程，我对同学们的表现给予以下评价与反馈：</p> <ul style="list-style-type: none"> - 积极性：同学们在课堂上的表现非常积极，能够主动参与讨论和实践活动。 - 学习态度：同学们对待学习的态度认真，对于新知识充满好奇心。 - 团队合作：在小组讨论中，同学们能够互相帮助，共同完成任务。 - 编程能力：部分同学在编程逻辑的理解上存在困难，需要加强练习。 - 改进建议：建议同学们在课后多进行编程练习，尤其是编程逻辑和算法方面的练习，以提高编程能力。 <p>总体来说，今天的课程达到了预期的教学目标，同学们在知识、技能和情感态度方面都有所收获。我会根据学生的反馈和评价，继续优化教学方法和内容，以更好地满足学生的学习需求。</p>	
<p>典型例题讲解</p>	<p>1. 例题：使用 Scratch 编程，编写一个程序，当按下按钮 A 时，LED 矩阵显示“Hello World！”；当按下按钮 B 时，LED 矩阵显示“Goodbye！”。</p> <p>解答：</p> <ul style="list-style-type: none"> - 初始化：创建一个新的 Scratch 项目，添加一个“when flag clicked”积木，用于初始化程序。 - 显示“Hello World！”：添加一个“set [LED Matrix v] to [0]”积木，将 LED 矩阵的值设置为 0。然后添加一个“say [Hello World! v] for (1) seconds”积木，显示“Hello World！”。 - 显示“Goodbye！”：添加另一个“when flag clicked”积木，用于处理按钮 B 的点击事件。同样，添加“set [LED Matrix v] to [0]”积木，然后添加“say [Goodbye! v] for (1) seconds”积木，显示“Goodbye！”。 <p>2. 例题：编写一个 Python 程序，使用 BBC micro:bit 的加速度计检测运动，当检测到向上倾斜时，LED 矩阵显示一个向上的箭头；当检测到向下倾斜时，显示一个向下的箭头。</p> <p>解答：</p> <pre> python from microbit import * while True: if accelerometer.x() > 500: display.show(Image.ARROW_N) elif accelerometer.x() < -500: display.show(Image.ARROW_S) else: display.clear() </pre> <p>3. 例题：使用 Scratch 编程，创建一个游戏，玩家通过控制 BBC micro:bit 的按钮来移动一个角色，角色在 LED 矩阵上移动，当角色碰到边缘时，游戏结束。</p> <p>解答：</p> <ul style="list-style-type: none"> - 初始化：创建一个新的 Scratch 项目，添加一个“when flag clicked”积木，用于初始化游戏。 - 移动角色：添加一个“forever”循环，在循环中添加一个“if”条件， 	

	检查按钮 A 是否被按下，如果是，则移动角色。	
--	-------------------------	--

-

	<p>检查边缘：在“forever”循环中，添加一个“if”条件，检查角色是否到达 LED 矩阵的边缘，如果是，则显示游戏结束的图片。</p> <p>4. 例题：编写一个 Python 程序，使用 BBC micro:bit 的磁力传感器检测方向，当检测到南方时，LED 矩阵显示一个指向南方的箭头；当检测到北方时，显示一个指向北方的箭头。</p> <p>解答：</p> <pre> python from microbit import * while True: if compass.heading() < 90: display.show(Image.ARROW_S) elif compass.heading() > 270: display.show(Image.ARROW_N) else: display.clear() </pre> <p>5. 例题：使用 Scratch 编程，创建一个简单的音乐播放器，当按下按钮时，BBC micro:bit 播放一段音乐。</p> <p>解答：</p> <ul style="list-style-type: none"> - 初始化：创建一个新的 Scratch 项目，添加一个“when flag clicked”积木，用于初始化音乐播放器。 - 播放音乐：添加一个“when flag clicked”积木，用于处理按钮的点击事件。然后添加一个“play sound [C5 v]”积木，播放 C5 音符。重复添加其他音符的积木，以创建一个简单的旋律。 	
--	--	--

第一单元 开源硬件—智能机器人第 2 课 高效的循环

主备人	
备课成员	
设计意图	<p>本课《高效的循环》旨在通过开源硬件—智能机器人的制作过程，引导学生深入理解循环在编程中的重要性。通过结合青岛版信息技术第四册教材，以实际操作为载体，激发学生的学习兴趣，培养学生的动手能力和逻辑思维能力。本节课将重点教授循环在智能机器人编程中的应用，帮助学生掌握基本的编程技巧，为后续学习打下坚实基础。</p>
核心素养目标	<ol style="list-style-type: none"> 1. 认知目标：理解循环在编程中的作用，掌握循环语句的基本结构和应用。 2. 技能目标：能够运用循环语句编写简单的智能机器人控制程序。 3. 情感目标：培养学生对开源硬件和编程的兴趣，增强解决问题的自信心。 4. 态度价值观目标：认识到创新思维和团队合作在科技发展中的重要性，树立科学精神和实践意识。
	1.

学习者分析	<p>学生已经掌握了哪些相关知识： 学生在进入本课程之前，已具备一定的信息技术基础，了解计算机的基本操作和基本概念。他们可能已经接触过简单的编程知识，如 Scratch 等图形化编程工具的基本操作。</p> <p>2. 学生的学习兴趣、能力和学习风格： 初中学生对新鲜事物充满好奇，对智能机器人等科技产品有较高的兴趣。他们的学习能力较强，能够快速适应新知识。学习风格上，部分学生可能更倾向于动手实践，通过操作来学习；而另一部分学生可能更偏好理论学习，通过阅读和理解来掌握知识。</p> <p>3. 学生可能遇到的困难和挑战： 在学习循环结构时，学生可能难以理解循环的概念和逻辑，尤其是在编写具体的控制程序时。此外，编程思维的培养也是一个挑战，学生可能难以将实际问题转化为程序逻辑。同时，对于一些学生来说，由于编程经验不足，可能难以调试程序中的错误。因此，教学中需要注重引导学生逐步理解编程逻辑，并提供足够的实践机会。</p>				
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
教学资源	<ul style="list-style-type: none"> - 软硬件资源：开源硬件套件（如 Arduino 或 Raspberry Pi）、智能机器人套件、编程板、连接线、传感器模块、执行器模块。 - 课程平台：学校网络教学平台、在线编程实验室。 - 信息化资源：智能机器人编程教材、教学视频、编程示例代码、在线编程教程。 - 教学手段：实物演示、分组讨论、互动编程环境、投影仪、计算机教室。 				
教学流程	<p>1. 导入新课</p> <ul style="list-style-type: none"> - 详细内容：首先，通过展示智能机器人的视频或图片，激发学生的兴趣。然后，简要介绍开源硬件和智能机器人的概念，提出本节课的学习目标：“今天我们将学习如何利用循环结构来编写智能机器人的控制程序，使机器人能够执行重复的任务。” - 用时：5 分钟 <p>2. 新课讲授</p> <ul style="list-style-type: none"> - 详细内容 1：讲解循环的概念和作用，通过示例代码展示循环在编程中的应用，如控制智能机器人的移动距离或重复执行某个动作。 - 详细内容 2：介绍循环语句的基本结构，包括 for 循环、while 循环和 do-while 循环，并解释循环变量、循环条件、循环体等概念。 - 用时：10 分钟 <p>3. 实践活动</p> <ul style="list-style-type: none"> - 详细内容 1：让学生分组，每组使用开源硬件套件和编程板，尝试编写一个简单的智能机器人程序，使机器人能够按照预设的循环结构执行任务。 - 详细内容 2：教师巡回指导，帮助学生解决编程过程中遇到的问题，如语法错误、逻辑错误等。 - 详细内容 3：组织学生展示自己的程序，分享编程过程中的经验和教训，鼓励学生互相学习和交流。 - 用时：15 分钟 				

	4. 学生小组讨论	
--	-----------	--

-

	<p>方面 1：讨论循环在智能机器人编程中的应用场景，如路径规划、动作重复等。</p> <ul style="list-style-type: none"> - 方面 2：分析不同循环结构的优缺点，以及在不同场景下的适用性。 - 方面 3：探讨如何优化循环程序，提高程序的执行效率和可读性。 - 举例回答： - 场景 1：在路径规划中，使用 while 循环可以实时检测机器人的位置，并根据预设的路径进行转向和前进。 - 场景 2：在动作重复中，使用 for 循环可以设定重复次数，方便控制机器人执行特定动作的次数。 - 优化方法：通过减少不必要的变量声明和简化循环体内的操作，提高程序的可读性和执行效率。 - 用时：10 分钟 <p>5. 总结回顾</p> <ul style="list-style-type: none"> - 内容：首先，回顾本节课所学内容，强调循环在智能机器人编程中的重要性。然后，总结学生在实践活动中的表现，肯定他们的努力和成果。最后，布置课后作业，要求学生完成一个包含循环结构的智能机器人程序，并提交给教师进行评价。 - 用时：5 分钟 <p>总用时：45 分钟</p>	
<p>学生 学习 效果</p>	<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. 知识掌握： 学生通过本节课的学习，能够理解并掌握循环在编程中的基本概念和作用。他们能够识别并应用 for 循环、while 循环和 do-while 循环等循环结构，理解循环变量、循环条件和循环体的概念。在课后作业中，学生能够独立编写包含循环结构的智能机器人程序，展示了他们对循环知识的实际应用能力。 2. 技能提升： 学生在实践活动中的动手能力得到了显著提升。他们通过实际操作开源硬件套件和编程板，学会了如何将理论知识转化为实际代码，实现了对智能机器人的基本控制。这种技能的提升不仅增强了学生的编程能力，也提高了他们的问题解决能力。 3. 思维发展： 通过本节课的学习，学生的逻辑思维和分析能力得到了锻炼。他们在讨论环节中能够分析不同循环结构的优缺点，并探讨如何优化程序。这种思维能力的培养有助于学生在未来的学习中更好地理解和应用编程知识。 4. 团队合作： 在小组讨论和实践活动过程中，学生学会了与他人合作，共同解决问题。他们学会了倾听他人的意见，尊重不同的观点，并在团队中发挥自己的作用。这种团队合作能力的培养对于学生未来的学习和工作具有重要意义。 5. 创新意识： 学生在完成智能机器人程序的过程中，不断尝试新的编程技巧和算法，展现了创新意识。他们能够根据实际需求调整程序，提出改进方案，这种创新能力的培养有助于学生在未来的科技发展中发挥积极作用。 6. 	

	<p>学习兴趣： 通过本节课的学习，学生对开源硬件和编程产生了浓厚的兴趣。他们愿意主动探索新的编程知识，并在课后继续学习相关内容。这种学习兴趣和激发的有助于学生形成终身学习的习惯。</p> <p>7. 实践应用： 学生通过本节课的学习，能够将所学的编程知识应用到实际项目中。他们能够根据实际需求设计智能机器人程序，解决实际问题。这种实践能力提升对于学生未来的职业发展具有积极影响。</p>	
<p>典型 例题 讲解</p>	<p>例题 1： 编写一个智能机器人程序，使其向前移动 10 步，每步移动的距离为 5 厘米。</p> <p>解答： ```python # 定义移动距离 distance = 5 # 循环移动 10 步 for step in range(10): print(f"机器人向前移动了第{step+1}步，总共移动了{step*distance}厘米。") # 假设移动函数 move() 实现机器人的移动 move(distance) ```</p> <p>例题 2： 编写一个智能机器人程序，使其绕圆形路径移动，圆的半径为 10 厘米，移动速度为每秒 1 厘米。</p> <p>解答： ```python import time # 定义圆的半径和移动速度 radius = 10 speed = 1 # 计算圆的周长 circumference = 2 * 3.14 * radius # 计算移动所需时间 total_time = circumference / speed # 开始移动 start_time = time.time() while time.time() - start_time < total_time: print("机器人正在绕圆形路径移动...") # 假设 move_circle() 实现机器人的圆形路径移动 move_circle(radius, speed) time.sleep(speed) ```</p> <p>例题 3： 编写一个智能机器人程序，使其在遇到障碍物时停止移动，并在障碍物移除</p>	

后继续前进。

解答：

```
```python
```

```
#
```



	<pre> 移动结束后，显示整个路径 print("机器人移动的路径为：") for point in path: print(f"({point[0]}, {point[1]})") ` ` ` </pre>	
<p>教学反思</p>	<p>哎呀，今天这节课过得还挺有意思的。咱们这节课主要是让学生们通过实践来理解循环在编程中的应用，我觉得效果还是不错的。但是，也有一些地方让我觉得可以改进。</p> <p>首先，我觉得在导入新课的时候，我可能可以更生動一些。我放了一段智能机器人的视频，学生们都挺感兴趣的，但是我觉得如果能够结合一些生活中的例子，可能会更好地激发他们的兴趣。比如说，我们可以讨论一下为什么智能机器人需要循环，它们在现实生活中是如何重复执行任务的。</p> <p>然后，在讲授新课的时候，我发现有些学生对于循环的概念理解得不是特别清楚。我可能需要花更多的时间来解释循环的原理，尤其是循环变量、循环条件和循环体这些概念。我打算在下一节课的时候，用一些简单的例子来帮助学生更好地理解这些概念。</p> <p>实践活动部分，学生们动手能力还是挺强的，但是有些小组在编写程序的时候遇到了一些问题。我发现，他们在调试程序的时候，往往不太会分析错误信息。我需要在未来的教学中，更加注重错误处理的教学，教他们如何阅读错误信息，如何定位问题所在。</p> <p>在小组讨论环节，我发现学生们能够积极地参与到讨论中来，但是他们的讨论内容有时候会比较发散。我需要更好地引导他们，让他们围绕重点问题进行讨论，比如循环的优化、不同循环结构的适用场景等。</p> <p>总的来说，这节课让我看到了学生们在编程方面的潜力和兴趣，也让我意识到了自己在教学上的不足。我会继续努力，改进教学方法，让我的学生们在信息技术这门课程中学到更多的知识，培养他们的创新能力。咱们一起加油吧！</p>	
<p>课堂</p>	<p>1. 课堂评价：</p> <ul style="list-style-type: none"> <li>- 提问环节：在课堂教学中，通过提问的方式检验学生对循环概念的理解。例如，询问学生：“循环在编程中有什么作用？”或者“你们能说出几种循环结构吗？”通过学生的回答，可以了解他们对循环知识的掌握程度。</li> <li>- 观察环节：在实践活动和小组讨论环节，观察学生的参与度和表现。例如，观察学生在编写程序时的操作是否熟练，是否能够独立解决问题，是否能够积极参与小组讨论。</li> <li>- 测试环节：在课程结束后，进行一次小测验，检验学生对循环知识的掌握情况。测试题目可以包括选择题、填空题和编程题，以全面评估学生的理解程度。</li> <li>- 具体实例： <ul style="list-style-type: none"> <li>- 提问：在讲解循环结构时，提问学生：“for 循环和 while 循环有什么区别？”通过学生的回答，发现他们对两种循环结构的理解存在差异，及时进行针对性讲解。</li> <li>- 观察：在实践活动环节，观察学生是否能够按照要求编写程序，是否能够与组员有效沟通，是否能够独立解决问题。</li> </ul> </li> </ul>	

	<p>测试：在课程结束后，进行一次小测验，包括选择题“循环结构中，循环体至少执行一次的是？”和编程题“编写一个程序，使机器人向前移动5步。”通过测试结果，了解学生对循环知识的掌握情况。</p> <p>2. 作业评价：</p> <ul style="list-style-type: none"> <li>- 批改作业：对学生的作业进行认真批改，包括程序的正确性、代码的规范性、注释的清晰度等方面。</li> <li>- 点评反馈：在批改作业的过程中，给予学生具体的点评和反馈，指出他们的优点和不足，并提出改进建议。</li> <li>- 及时反馈：将批改结果及时反馈给学生，鼓励他们在下一节课前完成改进。</li> <li>- 具体实例： <ul style="list-style-type: none"> <li>- 批改作业：在学生完成智能机器人程序作业后，逐个批改，确保每个学生都能得到反馈。</li> <li>- 点评反馈：例如，对于程序中的错误，可以指出具体错误原因，并给出修正建议；对于代码规范，可以指出不必要的空格、缩进等问题，并指导学生如何改进。</li> <li>- 及时反馈：在下一节课开始时，回顾上节课的作业情况，鼓励学生根据反馈进行改进，并分享他们的改进心得。</li> </ul> </li> </ul>	
<p>板书设计</p>	<ul style="list-style-type: none"> <li>① 循环结构概述 <ul style="list-style-type: none"> <li>- 循环的定义</li> <li>- 循环的作用</li> <li>- 常见循环结构</li> </ul> </li> <li>② 循环语句基本结构 <ul style="list-style-type: none"> <li>- 循环变量</li> <li>- 循环条件</li> <li>- 循环体</li> </ul> </li> <li>③ 循环结构示例 <ul style="list-style-type: none"> <li>- for 循环</li> <li>- while 循环</li> <li>- do-while 循环</li> </ul> </li> <li>④ 循环应用场景 <ul style="list-style-type: none"> <li>- 移动控制</li> <li>- 时间控制</li> <li>- 条件控制</li> </ul> </li> <li>⑤ 循环优化 <ul style="list-style-type: none"> <li>- 减少不必要的变量声明</li> <li>- 简化循环体内的操作</li> <li>- 提高程序可读性和执行效率</li> </ul> </li> </ul>	

## 第一单元 开源硬件—智能机器人第3课 灵活的变量

主备	
----	--

人	

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/495232313212012123>