

2024-2025 学年初中信息技术（信息科技）八年级 下粤教 B 版（第 4 版）教学设计合集

目录

一、第一章 感受程序设计的魅力

1.1 第一节 初识计算机程序

1.2 第二节 感受可视化的程序设计

1.3 本章综合

二、第二章 程序设计入门

2.1 第一节 电子相册

2.2 第二节 投票器

2.3 第三节 模拟抽奖箱

2.4 第四节 逐帧动画

2.5 第五节 移动的坦克游戏

2.6 第六节 节日彩灯

2.7 本章综合

三、第三章 程序设计综合应用

3.1 第一节 开发简易记事本

3.2 第二节 设计打地鼠游戏

3.3 第三节 综合活动：开发自己的程序

3.4 本章综合

第一章 感受程序设计的魅力第一节 初识计算机程序

一、设计意图

二、核心素养目标

1. 理解计算机程序的基本概念，培养信息意识。
2. 通过编写简单程序，发展逻辑思维与问题解决能力。
3. 激发对程序设计的兴趣，培养信息技术的实践与创新精神。

三、重点难点及解决办法

重点：理解计算机程序的定义、功能及其在生活中的应用。

难点：掌握程序设计的基本逻辑和简单的编程语言。

解决办法：

1. 使用生活中的实例帮助学生理解程序的作用，如手机应用、电脑游戏等。
2. 通过分步骤讲解和演示，使学生逐步理解编程的基本逻辑。
3. 引导学生通过编写简单的程序来实践，如使用 Scratch 或 Python 等入门级编程工具。
4. 鼓励学生小组讨论，共同解决问题，以增强理解和记忆。
5. 定期进行小测验和反馈，帮助学生巩固重点知识，及时发现并解决学习中的困惑。

四、教学资源

1. 硬件资源：计算机实验室、多媒体教室
2. 软件资源：Scratch 编程软件、Python 编程软件
3. 课程平台：学校教学管理系统
4. 信息化资源：教学 PPT、编程练习题库
5. 教学手段：案例分析、小组讨论、课堂互动

五、教学过程

1. 导入（约 5 分钟）

激发兴趣：通过展示一些常见的程序设计应用，如手机游戏、电脑软件等，引发学生对程序设计的好奇心。

回顾旧知：简要回顾上节课学习的内容，如算法的概念和特点，为学习程序设计打下基础。

2. 新课呈现（约 30 分钟）

讲解新知：介绍计算机程序的定义、功能、组成及编程语言的基本概念。

举例说明：通过展示简单的程序实例，如打印“Hello, World!”，让学生直观理解程序的作用和结构。

互动探究：组织学生讨论，思考程序设计在日常生活中的应用，以及如何利用程序解决问题。

3. 巩固练习（约 20 分钟）

学生活动：让学生利用 Scratch 或 Python 等编程工具，编写一个简单的程序，如画图、计算等，以加深对编程知识的理解和应用。

教师指导：在学生编写程序的过程中，教师巡回指导，解答学生的疑问，帮助学生解决遇到的问题。

4. 总结反馈（约 10 分钟）

总结：教师简要总结本节课的主要内容，强调程序设计在信息技术领域的重要性。

反馈：学生分享自己在编写程序过程中的体验和收获，教师对学生的表现给予肯定和鼓励。

5. 作业布置（约 5 分钟）

布置作业：要求学生在家中利用编程工具，尝试编写一个稍微复杂一些的程序，如计算器、小游戏等，以提高编程能力。

6. 课堂延伸（约 10 分钟）

延伸活动：鼓励学生参加学校或社区举办的编程比赛，激发学生的编程热情，提升学生的实践创新能力。

六、知识点梳理

1. 计算机程序的定义

- 程序是一系列指示计算机执行操作的命令。
- 程序用来处理数据、完成任务和解决问题。

2.

程序设计的基本概念

- 算法：解决问题的步骤序列。
 - 语法：编程语言中规定的语句结构和规则。
 - 逻辑结构：程序中的顺序结构、分支结构和循环结构。
- ### 3. 编程语言的分类
- 低级语言：接近计算机硬件的语言，如汇编语言。
 - 高级语言：更接近人类语言的编程语言，如 Python、Java、C++ 等。
- ### 4. 编程环境
- 编译器：将高级语言编写的程序转换成计算机能执行的机器语言。
 - 解释器：逐行解释和执行高级语言编写的程序。
- ### 5. 程序的执行过程
- 编写代码：使用编程语言编写程序。
 - 调试代码：发现并修正程序中的错误。
 - 运行程序：执行程序，完成指定的任务。
- ### 6. 程序设计的基本元素
- 变量：用于存储数据的标识符。
 - 函数：执行特定任务的代码块。
 - 控制结构：用于控制程序执行流程的语句，如 if、for、while 等。
- ### 7. 程序设计的步骤
- 分析问题：理解问题的需求。
 - 设计算法：规划解决问题的步骤。
 - 编写代码：根据算法编写程序。
 - 测试代码：检查程序的正确性和效率。
 - 优化代码：改进程序的性能和可读性。
- ### 8. 常见编程范式
- 命令式编程：直接命令计算机执行操作。
 - 声明式编程：描述程序的状态和结果，让计算机决定如何执行。
 - 函数式编程：使用函数和不可变数据来构建程序。
- ### 9. 程序调试技巧
- 打印语句：在程序中添加打印语句来查看变量的值。
 - 调试工具：使用集成开发环境（IDE）提供的调试工具。
 - 单步执行：逐行执行程序，观察程序的执行过程。
- ### 10. 程序设计的安全性和道德规范
- 遵守版权法：使用开源软件时遵守相应的许可证。
 - 避免恶意代码：不编写或传播有害程序。
 - 保护用户隐私：不收集或泄露用户个人信息。

七、板书设计

1. 计算机程序的基本概念

- ① 程序的定义
- ② 程序的功能

③ 程序的组成

2.

编程语言和编程环境

- ① 编程语言的分类
- ② 编程环境的作用
- ③ 常见的编程环境

3. 程序的执行过程

- ① 编写代码
- ② 调试代码
- ③ 运行程序

4. 程序设计的基本元素

- ① 变量的使用
- ② 函数的定义
- ③ 控制结构的类型

5. 程序设计的步骤

- ① 分析问题
- ② 设计算法
- ③ 编写代码
- ④ 测试代码
- ⑤ 优化代码

6. 编程范式和安全规范

- ① 命令式编程
- ② 声明式编程
- ③ 函数式编程
- ④ 安全性和道德规范

八、教学评价

1. 课堂评价

- ① 提问：通过课堂提问，检查学生对程序设计基本概念的理解程度，如程序的定义、功能等。
- ② 观察：观察学生在互动探究环节的表现，了解学生的合作能力和问题解决能力。
- ③ 测试：在课程结束时进行小测验，评估学生对课堂所学知识的掌握情况。
- ④ 及时反馈：对学生在课堂上的表现给予及时反馈，指出优点和需要改进的地方。

2. 作业评价

- ① 批改：认真批改学生的编程作业，检查代码的正确性、逻辑性和可读性。
- ② 点评：对学生的作业进行详细点评，指出代码的优点和存在的不足，如算法的选择、代码的结构等。
- ③ 反馈：及时将作业评价结果反馈给学生，鼓励学生根据反馈进行调整和改进。
- ④ 鼓励：对学生的进步和努力给予肯定和鼓励，激发学生的学习兴趣和自信心。

3. 过程性评价

- ① 跟踪：定期跟踪学生的学习进度，了解学生在学习过程中的困惑和问题。
- ② 讨论：组织学生进行小组讨论，观察学生在讨论中的表现，评估学生的交流合作能力。
- ③ 实践：鼓励学生参与编程实践活动，如编程竞赛、开源项目等，评估学生的实践能力和创新能力。

4. 综合性评价

- ①

综合评估：结合学生的课堂表现、作业完成情况、实践参与度等方面，进行综合性评价。

② 总结反馈：在学期末对学生的学习情况进行总结，给予综合评价，并提出改进建议。

③ 激励：对表现优异的学生给予表彰和奖励，激发学生的学习动力和进取心。

九、课后作业

1. 编写一个 Python 程序，实现以下功能：输入三个整数，输出这三个整数中的最大值。

```
答案：```python
a = int(input("请输入第一个整数："))
b = int(input("请输入第二个整数："))
c = int(input("请输入第三个整数："))
max_value = max(a, b, c)
print("最大的整数是：", max_value)
```
```

2. 编写一个程序，使用 for 循环打印 1 到 10 的平方。

```
答案：```python
for i in range(1, 11):
 print(i, "的平方是：", i**2)
```
```

3. 编写一个程序，使用 while 循环计算 1 到 10 的阶乘。

```
答案：```python
factorial = 1
i = 1
while i <= 10:
    factorial *= i
    i += 1
print("10 的阶乘是：", factorial)
```
```

4. 编写一个程序，实现以下功能：输入一个字符串，输出这个字符串中所有字母的大写形式。

```
答案：```python
input_str = input("请输入一个字符串：")
output_str = input_str.upper()
print("转换为大写的字符串是：", output_str)
```
```

5. 编写一个程序，实现以下功能：输入一个整数列表，输出这个列表中所有整数的和。

```
答案：```python
numbers = input("请输入一系列整数，用空格分隔：").split()
sum_numbers = sum(int(num) for num in numbers)
print("列表中所有整数的和是：", sum_numbers)
```
```

## 第一章 感受程序设计的魅力第二节 感受可视化的程序设计

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/017135166021006162>