

2024-2025 学年初中信息技术（信息科技） 八年级上册粤教清华版教学设计合集

目录

一、第1单元 用计算机程序解决问题

- 1.1 第1节 处理批量数据
- 1.2 第2节 神奇的字典
- 1.3 第3节 简单实用的枚举算法
- 1.4 第4节 “无所不能”的模块
- 1.5 本单元复习与测试

二、第2单元 体验计算机辅助设计

- 2.1 第1节 初识二维设计
- 2.2 第2节 三维建模和3D打印
- 2.3 本单元复习与测试

第1单元 用计算机程序解决问题第1节 处理批量数据

一、设计思路

本节课旨在让学生掌握使用计算机程序处理批量数据的基本方法，结合粤教清华版初中信息技术（信息科技）八年级上册第1单元第1节内容，设计以下课程：

- 引入实际生活中的批量数据处理场景，激发学生学习兴趣。
- 通过讲解和演示，使学生了解计算机程序处理批量数据的基本原理。
- 结合课本案例，引导学生动手实践，运用编程语言处理实际数据。
- 针对不同学生的掌握程度，设计梯度性的练习题，巩固所学知识。
- 总结课堂所学，布置课后作业，鼓励学生探索更多批量数据处理方法。

二、核心素养目标

- 信息意识：培养学生主动获取、处理和应用信息的能力，提高对批量数据处理重要性的认识。
- 计算思维：通过编程实践，发展学生的逻辑思维和问题解决能力，使其能够运用计算思维解决实际问题。
-

信息伦理：教育学生遵循信息伦理规范，尊重数据隐私，合法合规地处理批量数据。

4. 合作交流：在小组讨论和合作完成任务的过程中，提高学生的团队协作能力和沟通交流技巧。

三、学习者分析

1. 学生已经掌握了计算机基础知识，能够使用常见的办公软件，对编程语言有一定的了解，如 Python 或 Scratch 的基本操作。

2. 学习兴趣：学生对计算机编程有较高的兴趣，愿意尝试解决实际问题。学习能力：学生具备基本的逻辑思维和问题解决能力，能够跟随教程完成简单编程任务。学习风格：学生偏好动手实践，喜欢在操作中学习和探索。

3. 学生可能遇到的困难和挑战包括：对编程概念的理解不够深入，可能导致在处理复杂批量数据时感到困惑；编程实践中可能会遇到语法错误或逻辑错误，需要耐心调试和解决；同时，对于信息伦理和隐私保护的认识可能不足，需要在教学中加以引导。

四、教学方法与手段

教学方法：

1. 讲授法：讲解批量数据处理的基本概念和编程语言的基础语法。

2. 案例教学法：通过具体案例引导学生理解编程在处理批量数据中的应用。

3. 任务驱动法：设计实际操作任务，让学生在动手实践中掌握编程技能。

教学手段：

1. 多媒体教学：使用 PPT 展示教学要点和案例，增强视觉效果。

2. 编程软件：利用编程软件（如 Python 环境）进行实时编码演示和练习。

3. 网络资源：引导学生访问在线编程学习平台，扩展学习资源和实践机会。

五、教学过程

一、导入新课

1.1 向学生问好，确认学生出勤情况。

1.2 通过一个生活中的实例（如班级成绩统计）引入批量数据的概念，询问学生如何处理这些数据。

1.3 学生分享自己的想法后，我总结并引出本节课的主题——用计算机程序处理批量数据。

二、知识讲解与演示

2.1 介绍批量数据处理的必要性，以及计算机程序在这一过程中的作用。

2.2 讲解编程语言处理批量数据的基本原理，包括变量的使用、循环结构等。

2.3 通过 PPT 展示和现场演示，让学生观察程序如何读取、处理和输出批量数据。

三、案例分析与讨论

3.1 提供一个简单的批量数据处理案例，如计算一组数的平均值。

3.2 分组让学生讨论如何使用编程语言实现该案例，并鼓励学生提出问题。

3.3 每组派代表分享讨论结果，我进行点评和补充。

四、动手实践

4.1

发给学生编程任务，如使用 Python 编写一个程序，读取一组数据并计算总和、平均值等。

4.2 学生在计算机上实际操作，我巡回指导，解答学生在编程过程中遇到的问题。

4.3 学生完成编程任务后，邀请几名学生展示他们的程序，并简要解释代码逻辑。

五、深入学习与拓展

5.1 引导学生深入学习如何处理更复杂的批量数据，如使用条件语句和循环结构处理不同类型的数据。

5.2 提供一些进阶编程任务，鼓励学生尝试使用不同的编程语言和工具处理批量数据。

5.3 学生在计算机上完成进阶任务，我继续巡回指导，提供必要的帮助和建议。

六、信息伦理教育

6.1 讨论在处理批量数据时可能遇到的信息伦理问题，如数据隐私和合法使用。

6.2 提供一些案例，让学生分析如何正确处理这些伦理问题。

6.3 学生分享自己的看法，我进行总结和点评。

七、课堂小结

7.1 回顾本节课的主要内容，强调批量数据处理的重要性和编程语言的作用。

7.2 学生分享他们在本节课中学到的知识和技能。

7.3 我对学生的学习成果表示肯定，并布置相关的课后作业。

八、课后作业

8.1 布置一个编程作业，要求学生使用所学知识处理一组实际数据。

8.2 提供一些在线资源，帮助学生复习课堂内容并扩展知识。

8.3 鼓励学生在下节课前提交作业，并准备好分享他们的学习成果。

六、教学资源拓展

拓展资源：

1. 批量数据处理案例：收集不同行业和领域中批量数据处理的实际案例，如金融数据分析、市场调查数据统计、教育资源管理等，让学生了解编程在实际应用中的重要性。

2. 编程语言进阶知识：介绍 Python、Java、C++ 等编程语言中处理批量数据的高级技巧，包括数据结构（如列表、字典、集合）、算法（如排序、搜索）、数据库操作等。

3. 信息伦理案例：搜集涉及数据隐私、知识产权等方面的真实案例，引导学生深入理解信息伦理和法律法规，培养正确的数据处理观念。

4. 编程社区和论坛：推荐学生关注一些编程社区和论坛，如 Stack Overflow、GitHub、CSDN 等，以便在学习过程中遇到问题时可以寻求帮助和交流经验。

拓展建议：

1. 案例学习：鼓励学生通过分析批量数据处理的案例，了解编程在不同领域中的应用，并尝试将所学知识应用于解决实际问题。

2.

实践操作：建议学生在课后尝试使用不同的编程语言编写程序，处理一些简单的批量数据，以巩固课堂所学知识。

3. 信息伦理讨论：组织学生进行小组讨论，分析信息伦理案例，探讨如何在编程实践中遵循伦理原则和法律法规。

4. 编程社区参与：鼓励学生积极参与编程社区和论坛的活动，学习他人的编程经验，解决自己在编程过程中遇到的问题。

5. 阅读相关书籍：推荐学生阅读一些关于编程和批量数据处理的书籍，如《Python 编程：从入门到实践》、《数据结构与算法分析》等，以拓展知识面。

6. 学习编程工具：引导学生学习使用一些编程工具和库，如 Excel、Pandas、NumPy 等，这些工具在处理批量数据时非常实用。

7. 持续实践：鼓励学生持续参与编程项目，通过不断实践来提高编程能力和解决问题的能力。

七、典型例题讲解

本节课我们将通过几个典型例题来深入理解如何使用计算机程序处理批量数据。以下是几个与课本内容紧密相关的例题，每个例题都配有详细的解题步骤和答案。

例题 1：数据排序

题目：给定一个包含学生成绩的列表，请编写一个程序，将成绩从高到低排序。

解答：

```
```python
scores = [85, 90, 78, 92, 88]
sorted_scores = sorted(scores, reverse=True)
print("Sorted scores:", sorted_scores)
```
```

答案：Sorted scores: [92, 90, 88, 85, 78]

例题 2：计算平均值

题目：给定一个包含学生成绩的列表，请编写一个程序，计算这些成绩的平均值。

解答：

```
```python
scores = [85, 90, 78, 92, 88]
average_score = sum(scores) / len(scores)
print("Average score:", average_score)
```
```

答案：Average score: 86.6

例题 3：筛选数据

题目：给定一个包含学生成绩的列表，请编写一个程序，筛选出所有高于平均成绩的学生分数。

解答：

```
```python
```

```
scores = [85, 90, 78, 92, 88]
average_score = sum(scores) /
```

```

len(scores)
above_average = [score for score in scores if score > average_score]
print("Scores above average:", above_average)
```

```

答案: Scores above average: [90, 92, 88]

例题 4: 数据统计

题目: 给定一个包含学生成绩的列表, 请编写一个程序, 统计每个成绩范围内的学生人数。成绩范围分为: 0-59, 60-69, 70-79, 80-89, 90-100。

解答:

```

```python
scores = [85, 90, 78, 92, 88, 65, 70, 55, 95, 60]
range_counts = {
 '0-59': 0,
 '60-69': 0,
 '70-79': 0,
 '80-89': 0,
 '90-100': 0
}
for score in scores:
 if score < 60:
 range_counts['0-59'] += 1
 elif score < 70:
 range_counts['60-69'] += 1
 elif score < 80:
 range_counts['70-79'] += 1
 elif score < 90:
 range_counts['80-89'] += 1
 else:
 range_counts['90-100'] += 1
print("Score ranges and counts:", range_counts)
```

```

答案: Score ranges and counts: {'0-59': 1, '60-69': 1, '70-79': 1, '80-89': 3, '90-100': 4}

例题 5: 数据查找

题目: 给定一个包含学生信息的列表, 每个学生信息包含姓名和成绩, 请编写一个程序, 查找特定姓名的学生成绩。

解答:

```

```python
students = [

```

('Alice',

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/398121054124006135>