

面向对象编程工作坊 - 学生手册

第一部分：OOP概念及应用领域简介

什么是面向对象编程？

面向对象编程(OOP)是一种编程范式，它使用"对象"来组织代码，这些对象包含数据和操作数据的方法。OOP的核心理念是将现实世界的事物和概念抽象为代码中的类和对象。

核心概念

1. **类(Class)** - 对象的蓝图或模板，定义了对象的属性和行为。
2. **对象(Object)** - 类的实例，具有特定状态和行为。
3. **封装(Encapsulation)** - 将数据和方法绑定在一起，限制对数据的直接访问。
4. **继承(Inheritance)** - 允许一个类继承另一个类的特性，促进代码重用。
5. **多态(Polymorphism)** - 允许不同类的对象对相同消息做出不同响应。

OOP在现实世界中的应用

图形用户界面(GUI)应用程序

GUI应用程序使用对象来表示按钮、窗口和各种交互元素。例如，一个按钮对象可能具有属性(颜色、大小、位置)和行为(点击、悬停)。

游戏开发

游戏开发广泛使用OOP，游戏中的角色、敌人、物品和环境元素都可以表示为对象，每个对象都有自己的特性和行为。

设备控制

在物联网和嵌入式系统中，OOP用于创建表示不同设备的类，这些设备有各自的状态和控制方法。

模拟系统

OOP特别适合模拟现实世界的系统，从简单的物理模拟到复杂的生态系统或经济模型，对象可以模拟系统中的实体及其交互。

数据库交互

数据库应用程序使用对象映射技术来表示数据库表和记录，简化数据操作。

第二部分：p5.js球体模拟项目 - 下载与运行指南

在本次工作坊中，我们将使用p5.js创建一个互动的球体物理模拟系统。首先，让我们从获取并运行基础代码开始。

运行球体模拟程序的步骤

1. 打开浏览器，访问 <https://github.com/quenton1225/Balls>
2. 点击页面上的 **balls.txt** 文件，进入文件页面
3. 点击 "**Download raw file**" 按钮，下载后打开文件，复制全部代码
4. 打开新标签页，访问 p5.js 在线编辑器：<https://editor.p5js.org/>
5. 将复制的代码整体粘贴到编辑器的代码区域，替换原有内容
6. 点击左上角的▶（运行）按钮启动程序

基础代码功能说明

成功运行后，你将看到一个包含多个彩色小球的模拟场景。这个程序展示了面向对象编程的核心概念：

- **基础球体类(Ball)**：所有球体的基础类，定义了基本属性和行为
- **弹跳球类(BouncingBall)**：继承自基础球，能够在屏幕边界反弹
- **重力球类(GravityBall)**：继承自基础球，受重力影响并具有弹性

互动方式

- 点击屏幕：在点击位置创建一个新球
- 按键 1：切换为创建弹跳球模式
- 按键 2：切换为创建重力球模式
- 按空格键：给所有重力球一个向上的力（使它们跳起来）
- 按 R 键：重置场景，清除所有球体并创建新的初始球体

花一些时间玩一玩这个模拟，观察不同类型球体的行为差异。注意这些差异是如何通过面向对象编程实现的。

程序OOP概念图解

Ball (基类/父类)

- 属性:
 - x - X坐标位置
 - y - Y坐标位置
 - r - 半径
 - color - RGB颜色数组 [255,0,0] (红色)
 - xSpeed - 随机水平速度
 - ySpeed - 随机垂直速度
- 方法:
 - display() - 渲染球体
 - move() - 更新位置并检查边界
 - checkBoundaries() - 处理与画布边缘的碰撞

BouncingBall (子类)

- 属性:
 - (继承Ball的所有属性)
 - xSpeed - 覆盖父类 (从参数设置)
 - ySpeed - 覆盖父类 (从参数设置)
 - color - 覆盖父类 [0,0,255] (蓝色)
- 方法:
 - (继承Ball的所有方法, 没有重写)

GravityBall (子类)

- 属性:
 - (继承Ball的所有属性)
 - velocity - 向量, 替代xSpeed/ySpeed
 - acceleration - 重力加速度向量
 - damping - 碰撞时的能量损失系数
 - color - 覆盖父类 [255,150,0] (橙色)
- 方法:
 - move() - 重写父类方法 (应用物理学)
 - checkBoundaryCollision() - 替换父类的checkBoundaries()
 - display() - 重写父类方法 (根据速度添加亮度变化)
 - applyForce() - 额外方法, 用于施加力

第三部分：与大模型一起完成实践任务：p5.js球体模拟

现在你已经运行了基础代码并了解了它的功能，让我们开始通过实践任务来加深对OOP概念的理解。

基础代码概述

- **Ball类**：基本球体，具有位置、颜色和简单移动功能
- **BouncingBall类**：继承自Ball，增加了反弹行为
- **GravityBall类**：继承自Ball，添加了重力和弹性效果

任务选择（可选择完成）

从创建新的球类型开始是理解继承和扩展类功能的好方法。

脉动球(PulsingBall)

- 创建一个大小周期性变化的球
- 需要添加什么属性？(提示：考虑大小变化范围、变化速度)
- 如何修改display()方法来实现脉动效果？

彩虹球(RainbowBall)

- 创建一个颜色持续变化的球
- 如何使用RGB颜色模式实现平滑的颜色过渡？
- 考虑添加颜色变化速度控制

轨迹球(TrailBall)

- 创建一个移动时留下轨迹的球
- 如何存储球的历史位置？
- 轨迹应该如何衰减？

传送门球(PortalBall)

- 创建一个到达屏幕边缘时会传送到对面边缘的球
- 如何修改边界检查逻辑？
- 可以添加传送效果吗？

挑战：碰撞系统

- 实现球与球之间的碰撞检测和反应
- 如何检测两个球是否碰撞？
- 如何计算碰撞后的速度变化？(提示：考虑质量和动量守恒)

与大模型协作创建新球类型的指南

岭南大学为学生提供了免费的岭南GPT服务，访问地址是<https://chatgpt.ln.edu.hk/>（需要使用学生账户登录）。以下是如何有效地与大模型协作创建新球类的三阶段方法，供你参考：

第一阶段：项目描述与需求分析

首先，向大模型解释项目并描述你想创建的新球类型：

我正在使用p5.js创建一个基于面向对象编程的球体物理模拟系统。我想创建一个新的球类型"**[球类型名称]**"，它应该**[描述新球体的特殊行为]**。
当前代码已经作为附件附在最后。
我想要这个新球类型继承自基础的Ball类，但具有**[描述特殊特性]**功能。
请帮我分析一下，要实现这个功能，我可能需要添加哪些属性和方法？请给出一个简单的列表，不需要详细代码，只需要属性和方法名称以及它们的作用描述。

重点：记得将balls.txt下载，作为附件提供给大模型

第二阶段：属性和方法详细定义

拿到大模型提供的属性和方法列表后，请求更详细的设计：

谢谢你的分析。基于你提供的属性和方法列表，我想进一步了解每个部分的具体实现。
请帮我详细设计以下内容：

1. 新球类的构造函数应该接收哪些参数，以及如何调用父类构造函数？
2. 对于**[特定属性名]**，什么样的初始值和范围是合理的？
3. 对于**[特定方法名]**方法，它应该具体如何实现其功能？

请分别解释每个属性和方法的具体实现思路，但暂时不需要给出完整代码。

第三阶段：完整代码实现

最后，请求完整的代码实现：

感谢你的详细解释，我现在对如何实现这个新球类有了清晰的理解。

请帮我编写完整的代码实现，包括：

1. 新球类的完整定义
2. 需要在全局变量中添加的内容（如有）
3. 需要在createBall()函数中添加的代码
4. 需要在keyPressed()函数中添加的代码（如需添加新的按键控制）

如果可能，请解释一下关键部分的代码如何体现面向对象编程的概念，比如继承、多态等。

最后，请提供完整的更新后代码，确保所有新添加的内容与现有代码能够无缝集成。

处理问题的提示词

如果代码没有按预期工作，可以使用以下提示词：

新的代码运行后出现了[具体问题]，这偏离了我们的[原有目标]。请你意识到问题，并修复它。首先分析问题，然后输出完整的修复后代码。

Tips: 将报错内容展开，整段粘贴到[具体问题]部分即可。

如果大模型没有提供完整代码：

你的解释很有帮助，但我需要完整的代码实现。请提供整个更新后的代码，这样我可以直接复制粘贴到编辑器中运行。

如果想了解代码的工作原理：

这个功能基本实现了，但我想更好地理解它是如何工作的。能用简单的语言解释一下这个新球类是如何实现特殊效果的吗？

请记住，即使使用AI辅助编程，理解代码的工作原理仍然非常重要。每次添加新功能时，花时间理解AI提供的代码如何实现功能以及它如何与现有系统集成。

资源和参考

p5.js资源

- p5.js官方参考文档: <https://p5js.org/reference/>
- p5.js示例: <https://p5js.org/examples/>

OOP学习资源

- Data Camp的OOP教程: <https://app.datacamp.com/learn/courses/introduction-to-object-oriented-programming-in-java>

有用的JavaScript库

- matter.js (2D物理引擎) : <https://brm.io/matter-js/>
- d3.js (数据可视化) : <https://d3js.org/>

记住，编程是一个探索的过程。不要害怕尝试、失败和学习。使用AI作为你的学习伙伴，但确保你理解每一行代码的含义。最重要的是，享受创造的过程！

祝你编程愉快！