


面向对象特征讲解课 件



- 
- 面向对象编程简介
 - 面向对象特征
 - 面向对象编程语言特性
 - 面向对象编程实践
 - . 1 面向对象编程的总结
 - . 2 面向对象编程的未来发展

CHAPTER

面向对象编程简介



面向对象编程的定义



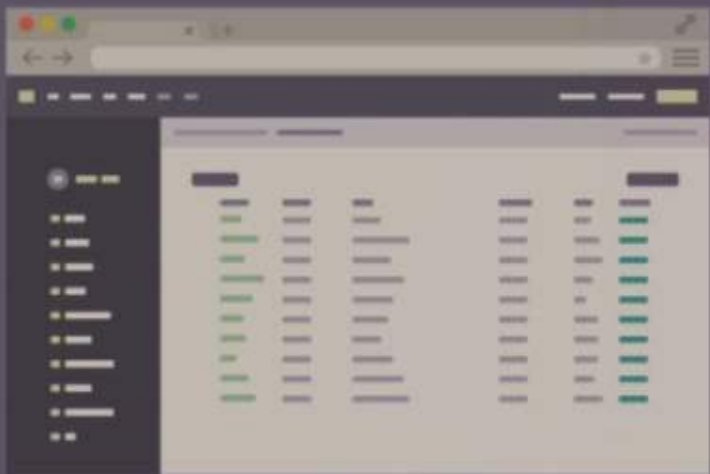
面向对象编程（Object-Oriented Programming，简称OOP）是一种编程范式，它以对象为基础，将数据和操作封装在一起，通过类和继承等机制实现代码的复用和扩展。

在面向对象编程中，对象是现实世界中事物的抽象，具有属性和行为。属性存储在对象内部，行为则通过方法来定义。





面向对象编程的起源与发展





面向对象编程的优势与局限性

优势

面向对象编程可以提高代码的可重用性、可维护性和可扩展性。通过封装和抽象，可以隐藏对象的内部细节，提高代码的安全性和稳定性。此外，面向对象编程还支持继承和多态等机制，使代码更加灵活和易于扩展。

局限性

面向对象编程也存在一些局限性。例如，过度使用面向对象编程可能会导致代码过于复杂和难以维护。此外，面向对象编程对于一些简单的问题可能过于复杂，导致开发效率降低。因此，在实际开发中需要根据实际情况选择合适的编程范式。

CHAPTER

面向对象特征



封装



封装是将对象的属性和方法封装在一起，形成一个独立的实体。



通过封装，对象的属性和方法被隐藏在内部，只通过外部接口与外界交互，保护对象的内部状态不被外部随意修改。



封装可以提高代码的安全性和可维护性，降低错误操作的可能性。

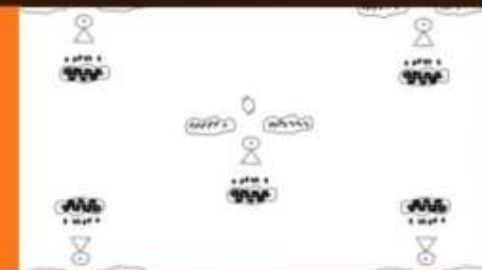


继承



继承是子类继承父类的属性和方法，子类可以继承父类的所有非私有属性和方法。

通过继承，子类可以重用父类的代码，减少重复的代码量，提高代码的复用性。



继承可以实现代码的层次结构，使得代码更加清晰和易于维护。



多态



多态是指一个接口可以有多种实现方式，或者一个对象可以有多种形态。



通过多态，可以实现同一个接口或类在不同情况下的不同行为，提高代码的灵活性和可扩展性。



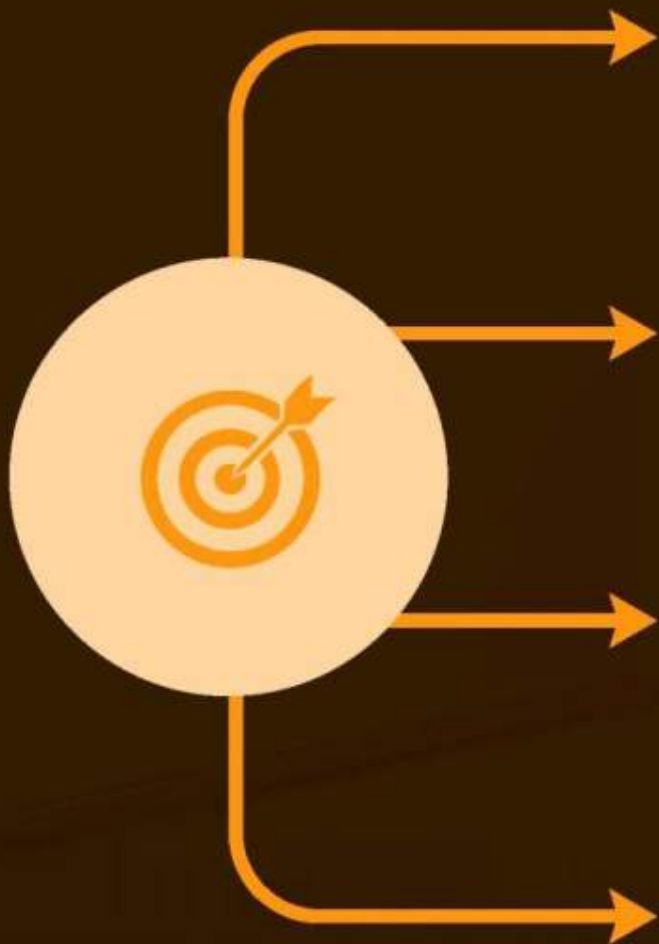
多态可以提高代码的可读性和可维护性，使得代码更加简洁和易于理解。

CHAPTER

面向对象编程语言特性



Java面向对象编程特性



封装

Java中的封装是指将对象的属性和行为封装在一个独立的单元中，通过访问控制符来控制对属性和行为的访问。

继承

Java中的继承是指一个类可以继承另一个类的属性和行为，从而形成一个更具体的类。

多态

Java中的多态是指一个接口或类可以有多种形态，即可以根据上下文的不同，将同一个接口或类实例当作不同的类型来处理。

抽象

Java中的抽象是指定义一个只提供了部分实现的类或接口，让子类或实现类来补充完整。



C面向对象编程特性



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/748012006043006067>