

里氏替换原则课件



目录

CONTENTS

- 里氏替换原则概述
- 里氏替换原则的规则和要求
- 里氏替换原则的实践应用
- 里氏替换原则的案例分析
- 里氏替换原则的挑战与解决方案
- 里氏替换原则的总结与展望

01

里氏替换原则概述





定义

定义1

如果对每一个类型为A的对象a，都有类型为B的对象b，使得为b的所有文字（在P中的）在替换a后，结果仍然是类型P的一个句子，那么，我们称对象a对程序P是可替换的，类型B的对象b称为a的里氏替换物。

定义2

如果对于每一个实例都属于A类的程序P，都可以用另一个属于B类的程序Q来替换，使得程序Q在所有可能的输入上产生的输出都与程序P产生的输出完全相同，那么我们称程序Q是程序P的一个里氏替换。



重要性

1

提高代码可维护性

里氏替换原则有助于提高代码的可维护性，因为遵循该原则的代码更容易理解和修改。

2

提高代码可读性

遵循里氏替换原则可以使代码更加清晰和易于理解，因为每个类都有明确的职责和功能。

3

降低维护成本

遵循里氏替换原则可以降低代码的维护成本，因为当某个类需要修改时，只需要修改该类本身，而不需要修改其他类。



历史背景

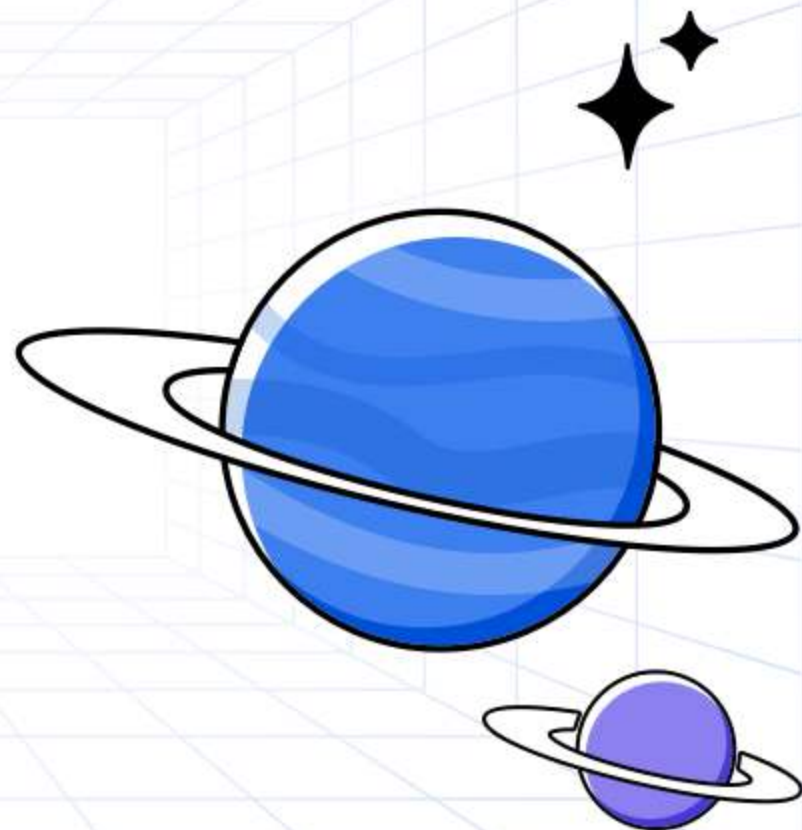


- 里氏替换原则是由美国计算机科学家理查德·里德在1988年提出的，该原则是面向对象设计的基本原则之一，旨在指导开发者如何设计和使用类和对象。



02

里氏替换原则的规则和要求





子类的函数调用父类的函数

规则

在子类中，如果一个函数在父类中已经存在，那么子类中的同名函数可以直接调用父类的函数，而不需要重新实现。这样可以减少代码重复，提高代码复用性。

描述

里氏替换原则要求子类可以替换父类而不会产生任何不期望的行为或错误。因此，当子类需要调用父类的某个函数时，可以直接使用父类的函数，而不是重新实现它。这样可以确保子类和父类之间的行为一致性。



子类对象可以替换父类对象

规则

在程序中，如果一个对象被声明为父类类型，那么它可以被赋值为子类对象。这种替换不会产生任何问题，因为子类对象可以访问其父类的属性和方法。

VS

描述

里氏替换原则要求子类对象可以完全替换父类对象，而不会破坏程序的正确性。这意味着在程序中，如果一个对象被声明为父类类型，那么它可以被子类对象所替代，而不会产生任何编译或运行时错误。这种替换的前提是子类必须实现了父类的所有方法，并且行为与父类一致。



替换后程序的行为不变



规则

在替换子类对象为父类对象后，程序的行为必须与替换前完全一致。任何与父类对象的交互都不应该察觉到其实际上是一个子类对象。

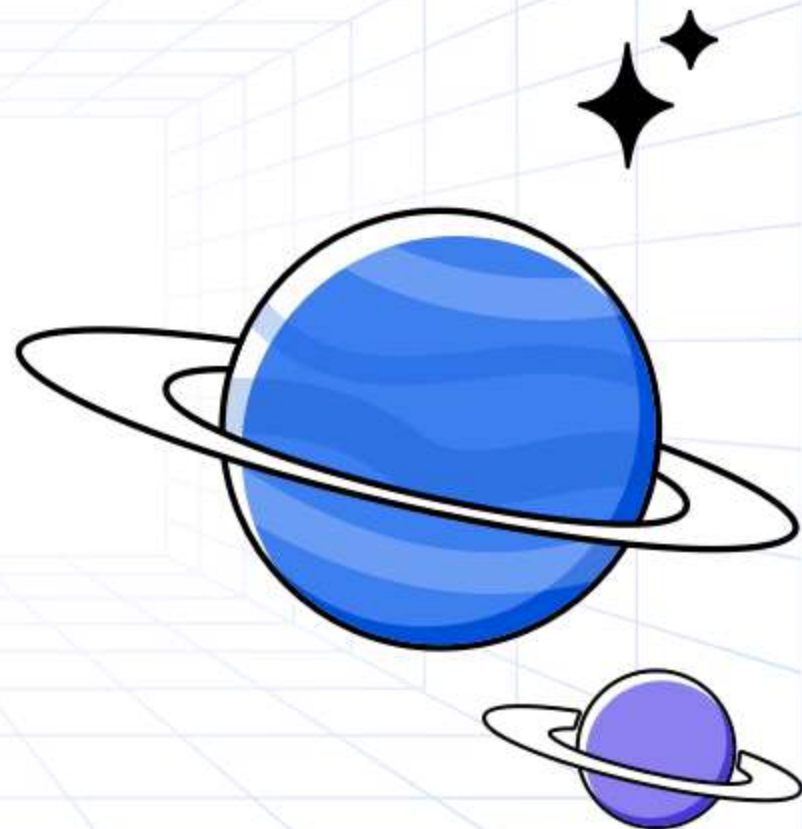


描述

里氏替换原则的核心思想是子类可以扩展父类的功能，但不会改变父类的原有行为。这意味着在程序中，如果一个子类对象被当作父类对象使用，那么它必须表现出与父类对象相同的行为，以确保程序的正确性和稳定性。这种替换的前提是子类必须严格遵守里氏替换原则的要求，即不改变父类的原有行为，只扩展其功能。

03

里氏替换原则的实践应用

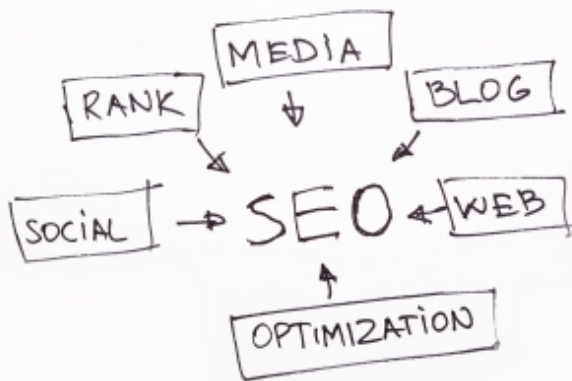




在设计模式中的应用

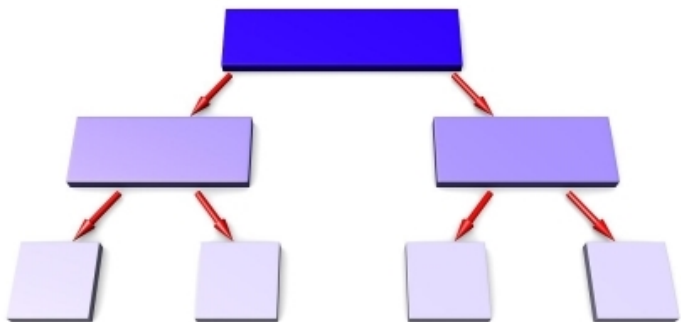
策略模式

在策略模式中，里氏替换原则要求子类必须能够替换其基类，以确保算法的灵活性和可扩展性。



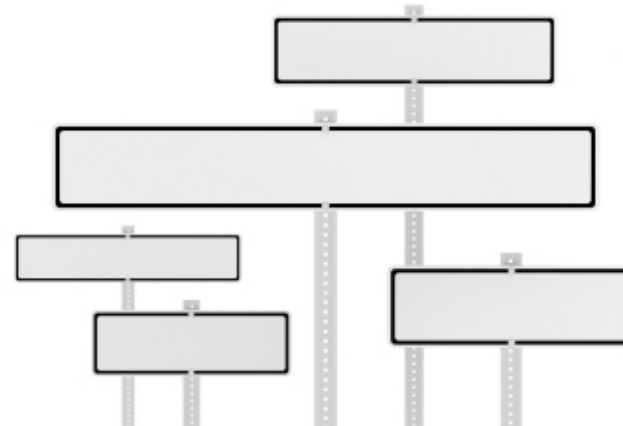
单例模式

在单例模式中，里氏替换原则要求子类的实例可以替换基类的实例，以确保整个应用程序中只有一个实例存在。



工厂模式

在工厂模式中，里氏替换原则要求子类的实例可以替换基类的实例，以便根据不同的需求创建不同的对象实例。





在单元测试中的应用



单元测试

在单元测试中，里氏替换原则要求子类的实现必须能够替换基类的实现，以便对子类进行独立的测试。

Mock对象

在单元测试中，Mock对象可以模拟基类的行为，以便对子类进行测试。Mock对象必须遵循里氏替换原则，以便在测试中正确使用。

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/957004121053006066>