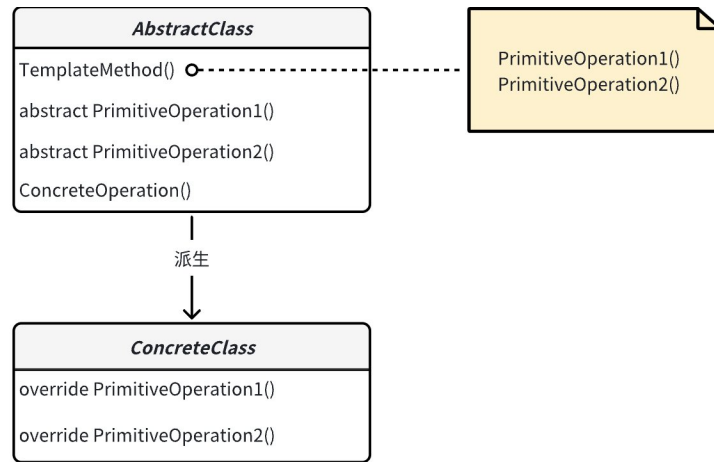


模板方法模式

概念

- 模板方法模式：定义了一个操作中的算法的骨架，而将一些步骤延迟到子类中。模板方法使得子类可以在不改变算法结构的情况下，重新定义算法中的某些步骤。
- 设计意图：通过把不变的行为搬移到超类，去除子类中的重复代码。它提供了一种代码复用的有效方式。
- 使用场景
 - 多个子类有共有的方法，并且逻辑基本相同时。
 - 重要的、复杂的方法，可以考虑作为模板方法。
 - 重构时，模板方法模式是一个经常使用的模式。
- 构成
 - 抽象类（AbstractClass）：定义模板方法和基本方法。模板方法定义了算法的骨架，而基本方法是算法的组成部分。
 - 具体类（ConcreteClass）：实现抽象类中的基本方法。
- 优点
 - 提高代码复用性：将相同部分的代码放在抽象的父类中，而将不同的代码放在不同的子类中。
 - 提高扩展性：增加新的行为时，只需要新增一个子类即可。
 - 符合开闭原则：通过增加子类来实现扩展，而不是修改已有的代码。
- 缺点
 - 类数量增加：每一个抽象类都需要一个具体类来实现，导致类的数量增加。
 - 类间关系复杂：增加了类间的耦合。



实例

- 抽象类

```
1 abstract class AbstractClass
2 {
3     // 模板方法
4     public void TemplateMethod()
5     {
6         PrimitiveOperation1();
7         PrimitiveOperation2();
8         ConcreteOperation();
9     }
10
11     // 基本方法（抽象）
12     protected abstract void PrimitiveOperation1();
13     protected abstract void PrimitiveOperation2();
14
15     // 基本方法（具体）
16     private void ConcreteOperation()
17     {
18         Console.WriteLine("ConcreteOperation: This is a concrete operation.");
19     }
20 }
```

- 具体类

```
1 class ConcreteClassA : AbstractClass
2 {
3     protected override void PrimitiveOperation1()
4     {
```

```

5         Console.WriteLine("ConcreteClassA.PrimitiveOperation1");
6     }
7
8     protected override void PrimitiveOperation2()
9     {
10         Console.WriteLine("ConcreteClassA.PrimitiveOperation2");
11     }
12 }
13
14 class ConcreteClassB : AbstractClass
15 {
16     protected override void PrimitiveOperation1()
17     {
18         Console.WriteLine("ConcreteClassB.PrimitiveOperation1");
19     }
20
21     protected override void PrimitiveOperation2()
22     {
23         Console.WriteLine("ConcreteClassB.PrimitiveOperation2");
24     }
25 }

```

- 客户端

```

1 static void Main(string[] args)
2 {
3     AbstractClass classA = new ConcreteClassA();
4     classA.TemplateMethod();
5
6     AbstractClass classB = new ConcreteClassB();
7     classB.TemplateMethod();
8 }

```

- 打印结果

```

1 ConcreteClassA.PrimitiveOperation1
2 ConcreteClassA.PrimitiveOperation2
3 ConcreteOperation: This is a concrete operation.
4 ConcreteClassB.PrimitiveOperation1
5 ConcreteClassB.PrimitiveOperation2
6 ConcreteOperation: This is a concrete operation.

```