

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green color. They are positioned diagonally, with the blue one in front of the green one.

Design Patterns

by Versus 2020

Immutable Interface

Неизменяемый интерфейс

Фундаментальный шаблон

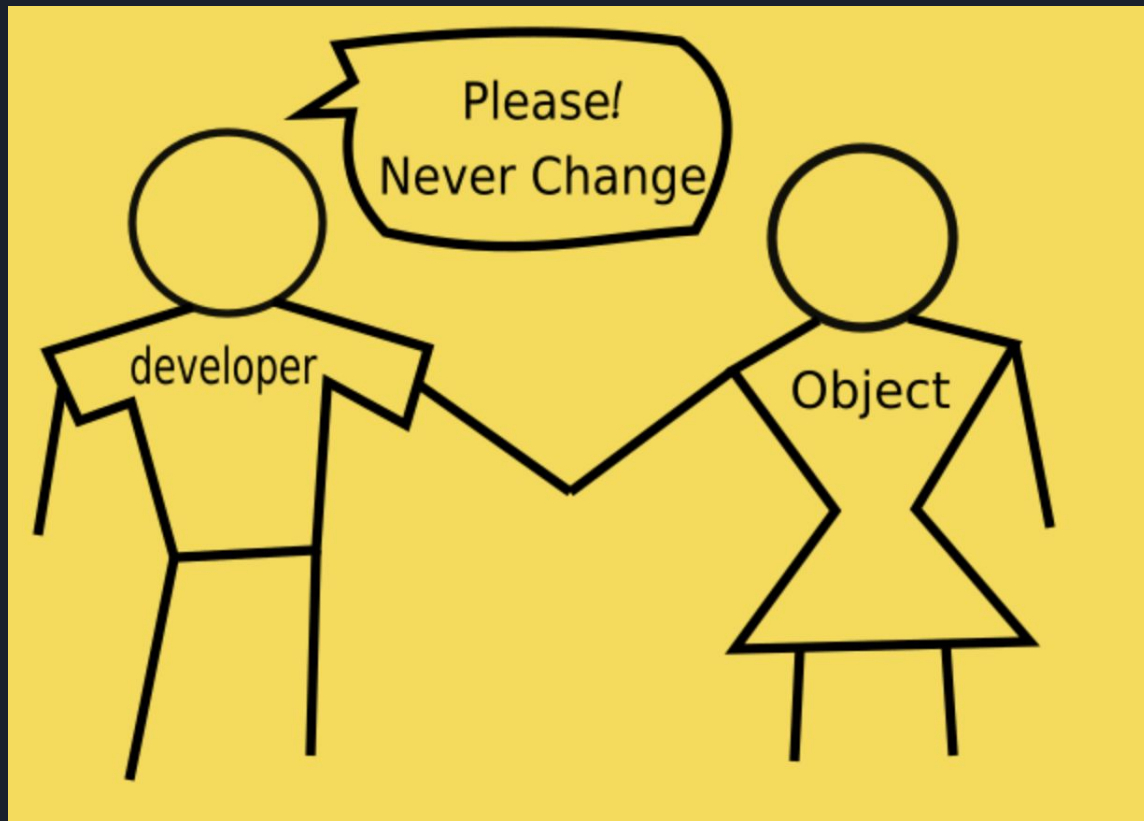




Суть шаблона

Служит для проектирования неизменяемого объекта.

Проблема





Решение

Определить и реализовать неизменяемый интерфейс.

Решение на Java

```
public class Point2D {
    private int x;
    private int y;
    public Point2D(int x, int y) { this.x = x; this.y = y; }

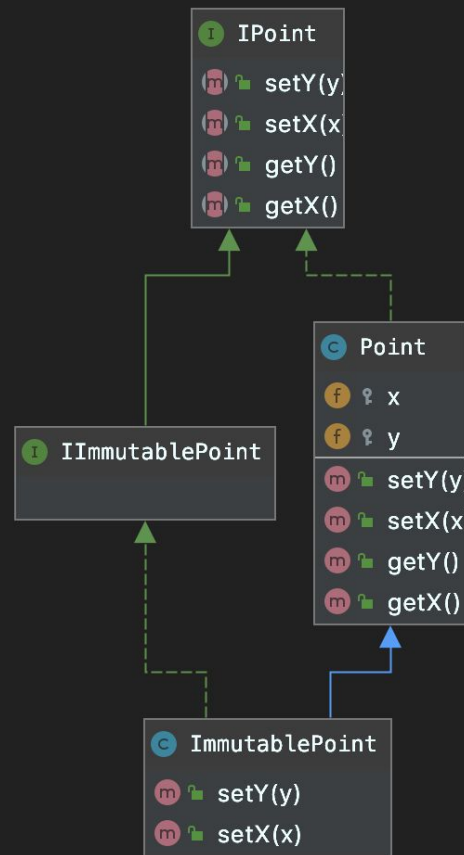
    public int getX() { return this.x; }
    public int getY() { return this.y; }

    public void setX(int newX) { this.x = newX; }
    public void setY(int newY) { this.y = newY; }
}

public interface ImmutablePoint2D {
    public int getX();
    public int getY();
}
```

```
ImmutablePoint2D point = new Point2D(0,0); // на конкретный экземпляр Point2D ссылается неизменяемый интерфейс
int x = point.getX(); // допустимый вызов метода
point.setX(42); // ошибка компиляции: метод setX() не существует в типе ImmutablePoint2D
```

Структура



Пример кода





Применимость

Нужно скрыть от изменения свойства объекта класса.



Шаги реализации

1. Определить базовый интерфейс вашего класса.
2. Определить неизменяемый интерфейс.
3. Унаследовать неизменяемый интерфейс от базового.
4. Реализовать неизменяемый интерфейс в потомке вашего базового класса.



Преимущества

Четко передает намерения о неизменяемости типа.



Недостатки

- Конкретные классы должны явно объявлять в их реализации неизменяемый интерфейс. Это не может быть возможно, если конкретный класс «принадлежит» стороннему коду.
- Объект не является на самом деле неизменяемым и, следовательно, не подходит для использования в структурах данных, требующих неизменяемость, таких как хеш-таблицы.

ИТОГ

