

Java Design Patterns

Антипаттерны

Java Design Patterns

Тема

Антипаттерны

Антипаттерны

Антипаттерны

Считаю важным знать не только удачные способы решения задач, но и возможные ошибки при их решении.

1. Интерфейс для констант
2. Внесенная сложность
3. Мягкое кодирование
4. Ненужная инициализация

Интерфейс для констант

Интерфейс для констант

При программировании очень часто приходится использовать константы. Константа, описанная в некотором классе, иногда может понадобиться и в другом классе.

Конечно, идея создать огромную кучу констант в одном интерфейсе с гибкой и прозрачной системой префиксирования подкупает своей новизной. Но это ни что иное как **антипаттерн проектирования**. Ведь в объектно-ориентированном программировании на Java интерфейсам была уготована более важная роль, чем просто хранить константы.

Внесенная сложность

Внесенная сложность

Усложнить можно любую задачу - это факт. Иногда программисты, и не только начинающие, при написании кода начинают применять все свои знания в сфере ООП. В результате вместо одного простого класса получается целая иерархия.

Мягкое кодирование

Мягкое кодирование

О хардкоде знают все. На ряду с хардкодом есть понятие "мягкое кодирование". Это маниакальное, граничащее с психозом, желание избежать хардкода.

Ненужная инициализация

Ненужная инициализация

Это скорее что-то среднее между антипаттерном проектирования и кодирования. Рассмотрим следующий программный код:

```
public class SomeClass {  
    private boolean active = false;  
    private int counter = 0;  
    private Object reference = null;  
  
    public SomeClass {  
        ...  
    }  
    ...  
}
```

Вероятно, разработчик кода хотел быть уверенным, что все поля класса проинициализированы. Но в данном конкретном случае это не нужно. В спецификации JLS говорится, что поля классов инициализируются значениями по-умолчанию - null для ссылок, 0 для целых типов и так далее.

В приведенном выше примере во время создания экземпляра класса будет добавлен ненужный шаг перед выполнением конструктора. Мелочь, но неприятно.

Хотя можно сделать чуть хуже - после такой явной инициализации полей добавить еще их инициализацию внутри тела конструктора.

Информационный видеосервис для разработчиков программного обеспечения

