ХАРДКОДИНГ, СОФТКОДИНГ

АНТИПАТТЕРНЫ

ХАРДКОДИНГ

- Хардкодинг, hardcoding - прямое указание значений каких-либо теоретически параметризуемых сущностей в тексте программы или построение её логики как если бы эти сущности имели постоянное значение. Обычно применяется для ускорения разработки ПО, часто - ленивыми программистами.

ЧЕМ ОПАСЕН

При изменении каких-либо данных, "вшитых" прямо в программный код, понадобится изменять код.

Если забыть о каком-то захардкоженном куске кода может случиться так, что программа будет работать корректно только на компьютере разработчика. Для устранения этой ошибки понадобится досконально изучить код.

ПРИМЕРЧИК

```
public class LoansController {
    User user;

    public void loanApplication() {
        if (user.age >= 23) {
            // можно получить кредит
        }
    }
}
```

- Когда однажды выяснится, что для получения кредита нужно стать старше 23 лет, да ещё и найти работу, нам придётся найти в коде все места, где прописано `if (Age >= 23)` и поменять их на `if (age > 23 && employed)`. Но как найти все знаки `>=`? Их же тысячи! Вот это ручная работа на столетия!
- Но самое страшное, что в коде могут быть и выражения вида:
  - `if (!(age < 23))`, и
  - `if (23 > age)`,

...и даже такие места, которые совсем нереально обнаружить:

```
if (age < 23)
{
   // 100500 строк кода
}
else
{
   // можно получить кредит
}
```

ЧТО ЖЕ ДЕЛАТЬ?

Вот почему важно выносить не только константы, а ещё и логику. Важно следить, чтобы любое знание в коде было прописано ровно в одном месте. В данном случае - знание о том, в каких случаях клиент может взять кредит (то самое >= 23) должно быть вынесено в отдельный метод.

ОДИН ИЗ ПЛЮСОВ ВЫНЕСЕНИЯ ЛОГИКИ В МЕТОДЫ - ЕЁ ЛЕГКО ТЕСТИРОВАТЬ

СОФТКОДИНГ

ПОЧЕМУ ЭТО ПЛОХО

- Если система становится сильно настраиваемой, проблемой может стать обработка ошибок
- Модификация кода становится сложнее
- Само по себе использование этого кода может стать достаточно сложным