

XII международная конференция
CEE-SECR / РАЗРАБОТКА ПО

28 - 29 октября, Москва



О SOLID-ности

Евгений Тюменцев

HWdTech, LLC

hwdtech.ru

The Open-Closed Principle (OCP)

Программные объекты должны быть открыты для расширения, но в тоже время закрыты для модификации.

The Liskov Substitution Principle

Функции, которые используют ссылки на базовые классы, должны использовать объекты производных классов, не зная об этом.

The Dependency Inversion Principle

Высокоуровневые компоненты не должны зависеть от низкоуровневых компонент. И те, и те должны зависеть от абстракций.

Абстракции не должны зависеть от деталей. Детали должны зависеть от абстракций.

The Interface Segregation Principle

Класс не должен зависеть от интерфейсов, которые он не использует.

The Single Responsibility Principle

Должна быть ровно одна причина для изменения класса.

Не SOLID-ные конструкции

switch

If /else – if/else

enum

new

Операторы приведения типа

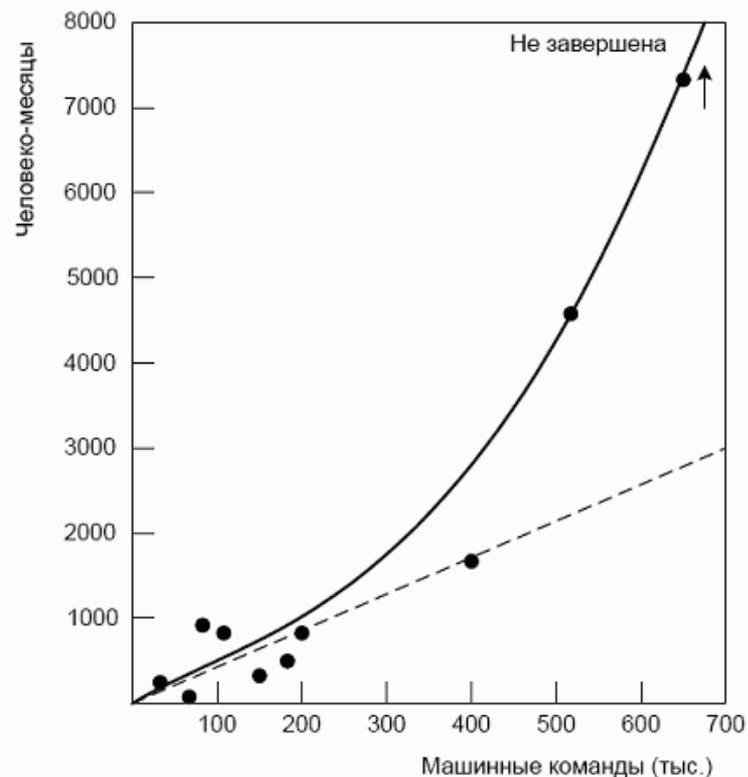
Магические константы



Что-о-о? switch,
new? Ты кто
такой?

ОСР – достаточное условие линейной
производительности

Особенности разработки ПО



Фредерик Брукс
«Мифический человеко-
месяц»

Статический полиморфизм

```
template <class It, class Op>
void for_each(It begin, It end, Op op)
{
    for(; begin != end; ++begin)
        op(*begin);
}
int arr[] = {1, 2, 3, 4, 5};
for_each(arr, arr+6, max<int>());
```

Динамический полиморфизм

- указатель на функцию

```
typedef void (*f) (int i);
```

```
f pf = ...;
```

```
pf(5);
```

- виртуальный метод

```
class Shape {  
    public:  
        virtual void Draw() = 0;  
};
```

```
shape->Draw();
```

Метапрограммирование

```
void Counter::setValue(int value)
{
    if (value != m_value)
    {
        m_value = value;
        emit valueChanged(value);
    }
}
```

Выводимость

Пусть L – множество формул, V – формула.

Тогда $L \vdash V$, если $\exists V_1, V_2, \dots, V_n$, что

1. V_n – это V ,

2. V_i – это

либо формула из L ,

либо аксиома,

либо общезначимая формула,

либо формула полученная при

помощи правила вывода

Тройка Хоара

{pred} statement {post}

Пример: $\{x == 42\} y=x+1; \{y == 43 \wedge x == 42\}$

Аксиома пустого оператора

{P} skip {P}

Аксиома оператора присваивания

$$\{P[E/x]\} x := E \{P\}$$

Аксиома оператора цикла

$\{P \wedge B\} S \{P\} \models \{P\} \text{ while } B \text{ do } S \text{ done } \{B' \wedge P\}$

Аксиома условного оператора

$\{B \wedge P\} S \{Q\}, \{B' \wedge P\} T \{Q\} \models \{P\} \text{ if } B \text{ then } S \text{ else } T \text{ endif } \{Q\}$

$\{B \wedge P\} S \{Q\} \models \{P\} \text{ if } B \text{ then } S \text{ endif } \{Q\}$

Аксиома композиции

$$\{P\} S \{Q\}, \{Q\} T \{R\} \vdash \{P\} S;T \{R\}$$

Аксиомы вывода

$$P1 \rightarrow P, \{P\} S \{Q\}, Q \rightarrow Q1 \vdash \{P1\} S \{Q1\}$$

$$\begin{array}{l} P1 \rightarrow P, \{P\} S \{Q\} \vdash \{P1\} S \{Q\} \\ \{P\} S \{Q\}, Q \rightarrow Q1 \vdash \{P\} S \{Q1\} \end{array}$$

Чем грозят правки кода?

B_1, B_2, \dots, A_1

B_3, A_1, \dots, A_2

B_5, A_2, \dots, A_3

A_2, B_6, \dots, A_4

A_2, A_4, \dots, A_5

Может все сломаться!

B_1, B_2, \dots, A_1

B_3, A_1, \dots, A_2

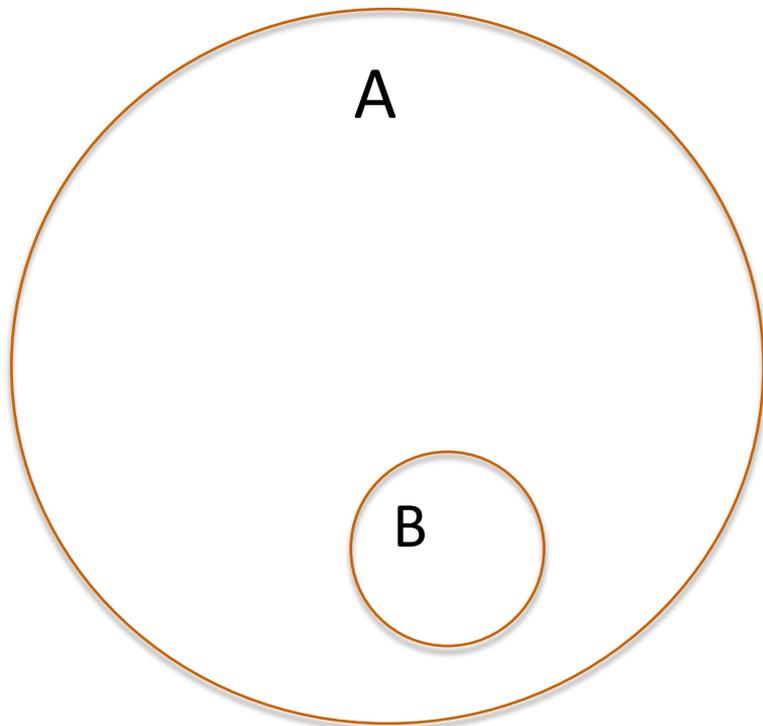
B_5, A_2, \dots, A_3

A_2, B_6, \dots, A_4

A_2, A_4, \dots, A_5



Импликация на множествах



Пусть $P1 \rightarrow P$,

$B = \{x \mid P1(x) = 1\}$,

$A = \{x \mid P(x) = 1\}$.

Тогда $B \subset A$.

$P \vDash P1$. Говорят, что P – **более слабое** условие, $P1$ – **более сильное**.

Построим вывод

$\{S\} \text{ cb } \{P1\}, \{P1\} \text{ c } \{Q1\}, \{Q1\} \text{ ca } \{R\} \vdash \{S\} \text{ cb}; \text{ c}; \text{ ca } \{R\}$

Известно, что $P1 \rightarrow P, Q \rightarrow Q1$

Тогда

$\{P\} \text{ c } \{Q\} \vdash \{P1\} \text{ c } \{Q1\}$ (аксиома вывода)

Получаем

$\{S\} \text{ cb } \{P1\}, \{P\} \text{ c } \{Q\}, \{Q1\} \text{ ca } \{R\} \vdash \{S\} \text{ cb}; \text{ c}; \text{ ca } \{R\}$

Критерий расширения

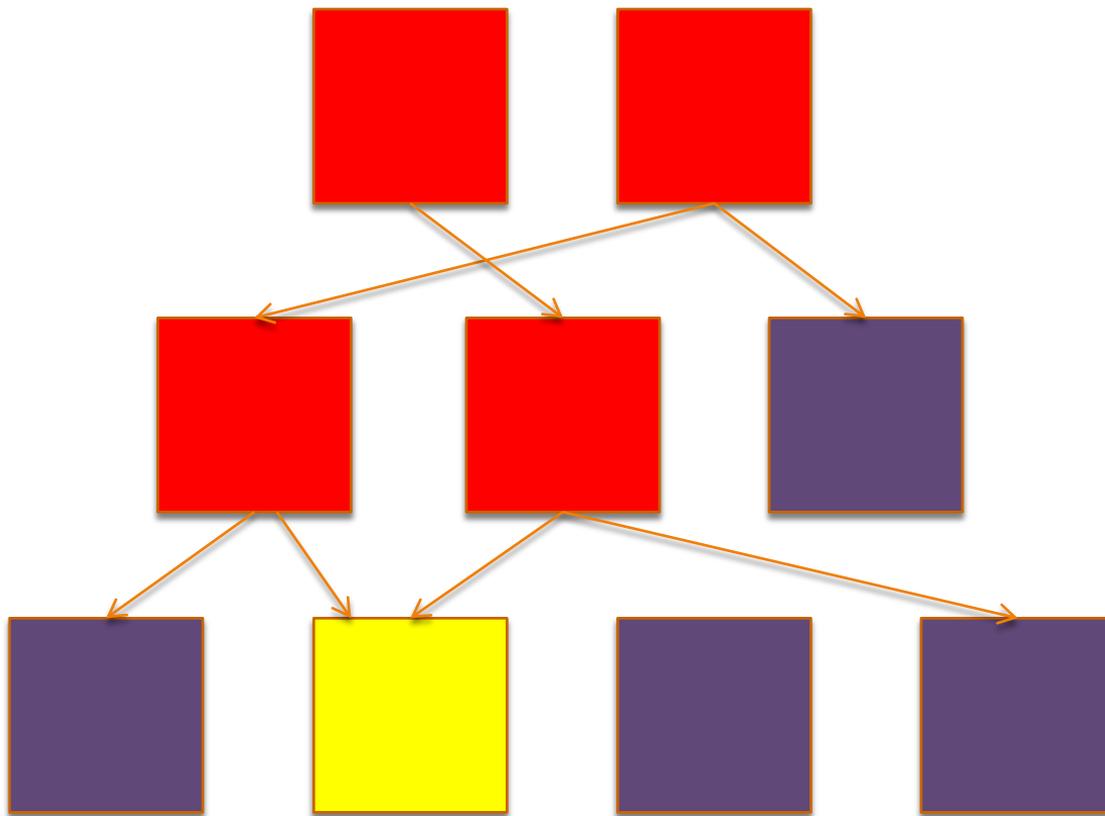
Оператор расширения должен допускать:

1. Ослабление предусловий
2. Усиление постусловий

Метафора для расширения



Зависимости



Так ли страшен dependency hell?

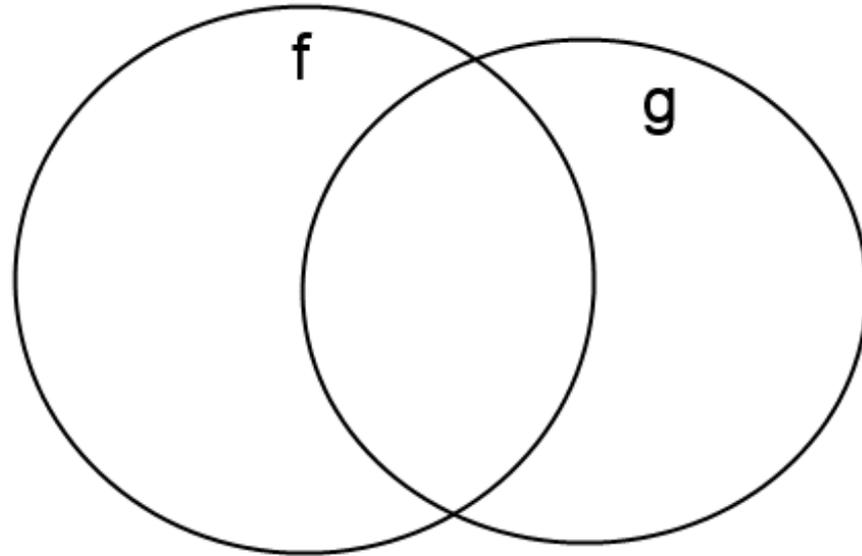
24 марта 2016 <https://habrahabr.ru/post/280099/>

left-pad

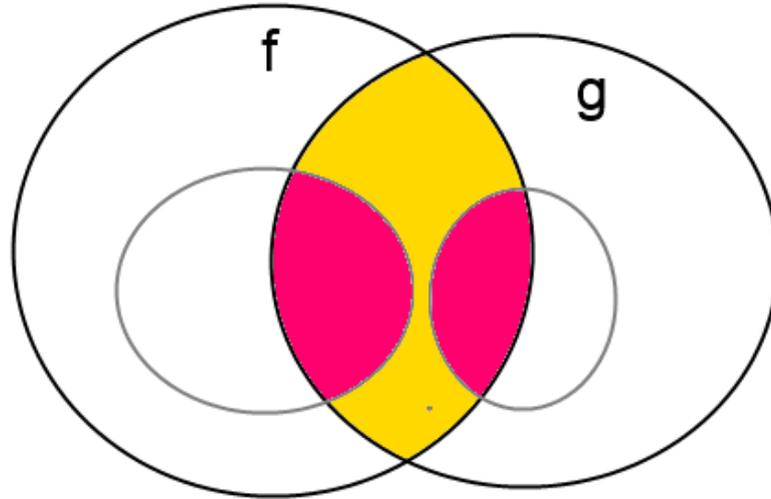
Проблема 2000 года

Переход с 16 разрядной архитектуры на 32 разрядную

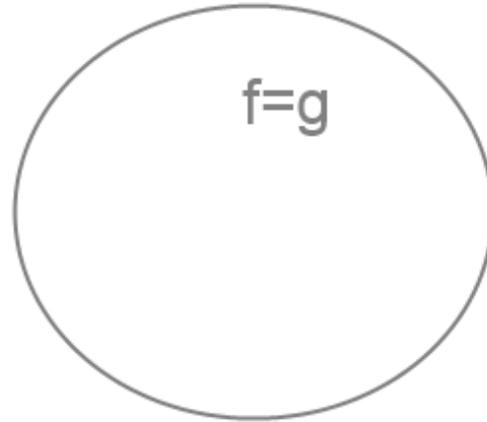
Несколько операторов расширения



Жирный интерфейс



Как избежать жирного интерфейса



Спасибо за внимание!

Тюменцев Евгений

Звоните: +7 913 150 22 04

Пишите: etyumentsev@hwdtech.ru