

# Паттерны объектно-ориентированного проектирования

Алексей Владыкин

СПбГУ ИТМО

16 ноября 2009

- Design Pattern — паттерн (шаблон, образец) проектирования.
- «Каждый паттерн описывает некую повторяющуюся проблему и ключ к ее разгадке, причем таким образом, что этим ключом можно пользоваться при решении самых разнообразных задач.» — Christopher Alexander
- В отличие от идиом, паттерны универсальны, т.е. не завязаны на определенный язык программирования.
- Джонсон Р., Хелм Р., Гамма Э. Приемы объектно-ориентированного проектирования: Паттерны проектирования. — СПб.: Питер, 2001. — 368 с.

# Проектируем игру «Крестики-нолики»

По-английски: Tic-tac-toe, Noughts and Crosses, X's and O's.

Объектная декомпозиция задачи:

- игровое поле;
- две противоборствующих стратегии;
- пользовательский интерфейс;
- контроллер.

## Игровое поле

- Пассивная сущность.
- Хранит состояние поля.
- Ничего не знает ни о стратегии игры, ни о пользовательском интерфейсе.

```
enum NCMark { NONE, NOUGHT, CROSS };  
  
class NCField {  
public:  
    virtual int getWidth() const = 0;  
    virtual int getHeight() const = 0;  
    virtual NCMark getMark(int i, int j) const = 0;  
    virtual bool setMark(int i, int j, NCMark mark) = 0;  
};
```

# Стратегия игры

- Задает алгоритм совершения одного хода.
- Знает только об игровом поле.
- Паттерн Strategy.

```
class NCStrategy {  
public:  
    virtual bool makeTurn(NCField& field) = 0;  
};
```

# Контроллер

- Активная сущность.
  - Инициализирует модель: поле и две стратегии.
  - Инициализирует представление: графический интерфейс.
  - По очереди дает каждой стратегии сделать ход.
  - Проверяет выигрыш одной из стратегий.
  - Обновляет интерфейс.
- Модель + представление + контроллер =  
паттерн Model—View—Controller