

Виталий Хить, Абсолютист

РАЗНИЦА МЕЖДУ КОДИРОВАНИЕМ И ПРОГРАММИРОВАНИЕМ

ПРОГРАММИРОВАНИЕ

```
graph LR; A[ПРОЕКТИРОВАНИЕ] --> B[КОДИРОВАНИЕ]; B --> C[ОТЛАДКА]
```

ПРОЕКТИРОВАНИЕ

КОДИРОВАНИЕ

ОТЛАДКА

КОДИРОВАНИЕ



ПРОЕКТИРОВАНИЕ & КОДИРОВАНИЕ

- Проектирование
 - Объектная модель
 - Паттерны
- Кодирование
 - Реализация проектной модели на языке программирования
 - Использование программной среды
 - Языково зависимые конструкции
 - Фреймворки

МОНЕТИЗАЦИЯ КОДА

- Поддержка программы
- Создание новых версий программы
- Повторное использование
 - Уровень функций
 - Уровень классов
 - Уровень библиотек
- Open Source (уровень фреймворков)
- Продажа исходников

УРОВНИ АБСТРАКЦИИ КОДИРОВАНИЯ

- Оформление кода
- Пространство имен
- Языковые конструкции
- Фреймворки

ОФОРМЛЕНИЕ КОДА

Пример 1

```
if (a) b = 2; else b = 3;
```

Пример 2

```
if (a)
```

```
{
```

```
    b = 2;
```

```
}
```

```
else
```

```
{
```

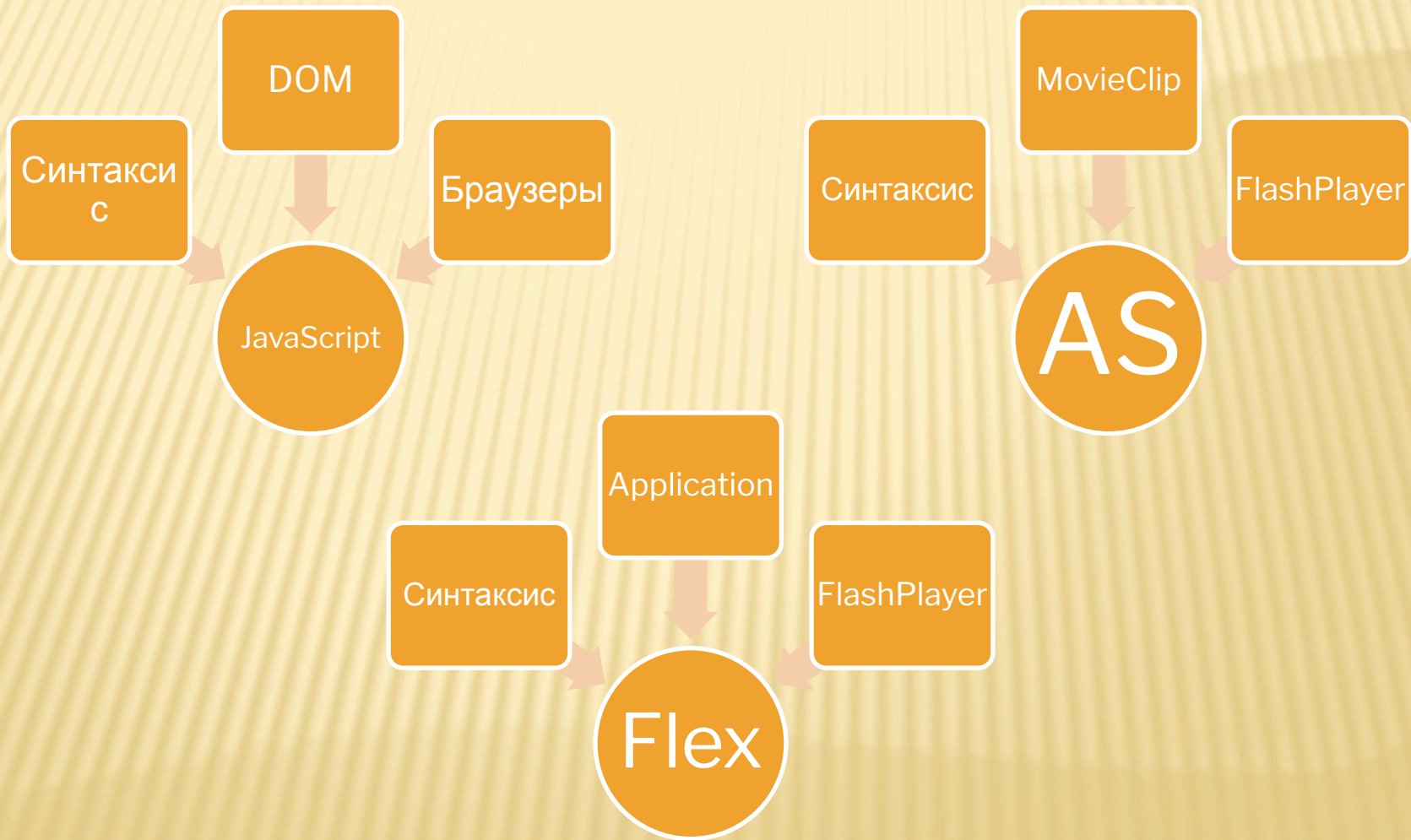
```
    b = 3;
```

```
}
```

ПРОСТРАНСТВО ИМЕН

- Стандартные правила имен в языке
- Корпоративные правила именовани
- Правила имен в фреймворке

ЯЗЫКОВЫЕ КОНСТРУКЦИИ



ЯЗЫКОВЫЕ КОНСТРУКЦИИ

▣ JavaScript

```
var Class1 = {};  
Class1.x = "Test";  
Class1.prototype.functionA = function() { return 5; }  
Class1.functionB = function() { return this.x; }  
function Class2() {}
```

▣ AS2

```
class Class1  
{  
    public var x :String = "Test";  
    function Class1() {};  
    function functionA() :Number { return 5; }  
    function functionB() :String { return this.x; }  
}
```

▣ AS3

- ▣ Приведение типов
- ▣ Полный static

ФРЕЙМВОРКИ ИЛИ ОСОБЕННОСТИ ЯЗЫКА?

□ $2 + 2 = 4$?

```
Sprite s = new Sprite();
```

```
s.x += 0.0000000001; trace(s.x); // 0
```

```
Number y = 0;
```

```
y += 0.0000000001; trace(y); //  
0.0000000001;
```

□ mxml

ЧИТАБЕЛЬНОСТЬ & ПРОИЗВОДИТЕЛЬНОСТЬ

AS3

```
var i:int = 1;      push 1
var j:int = 2;      push 2
var k:int = i + j;  add
                   convert_i
```

HAXE

```
add_i
```

```
inline
```

Tamarin (ByteCode)

```
function main      function
{
  getlocal0        {
  pushscope        pushbyte 1
                   pushbyte 2
  finddef print    add_i
  pushbyte 1       .print
  pushbyte 2       }
  add_i
  callproperty print(1)

  returnvoid
}
```

САМОДОКУМЕНТИРОВАННЫЙ КОД

- Документирование классов и членов классов
- Документирование заголовков функций
- Документирование кода
- Документирование соглашений

УНИВЕРСАЛЬНЫЕ ФРЕЙМВОРКИ

Flex

Достоинства

- Все в одном флаконе
- Индустриальный стандарт

Недостатки

- Mxml
- Большой вес
- Базовый класс (>400К текста)

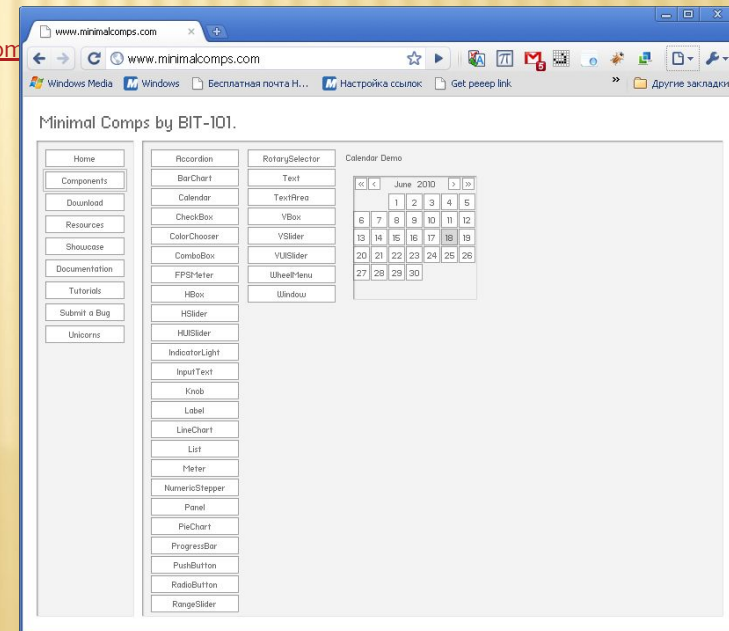
MinimalComps (bit-101.com) <http://www.minimalcomps.com>

Достоинства

- Базовый класс (<7К)
- Высокая модульность

Недостатки

- Отсутствие соглашений
- Слабая расширяемость
- Грустненький вид



ЧУКЧА - ПИСАТЕЛЬ

- Зачем необходимо уметь читать чужой код
 - Документация & исходники
 - Библиотеки в бинарниках или исходниках
- Как этому научиться
 - Гугл код
 - Портирование с другого языка

СПАСИБО ЗА ВНИМАНИЕ

**ВОПРОС
Ы?**