

Разработка приложений для OS X

Лекция II
Еще немного ООП. Свойства. Наш браузер.

Hexlet University, 2012

Изменение формата

- Тесты
- Оценки
- Подкаст

Создание объектов

```
NSString *title = [[NSString alloc] init];
```

vs.

```
NSString *title = [NSString string];
```

```
NSNumber* value = [[NSNumber alloc] initWithFloat:1.0];
```

Освобождение памяти

Если ARC выключен то освобождать память нужно будет вручную.

```
// string1 will be released automatically  
NSString* name = [NSString string];
```

```
// must release this when done  
NSString* song = [[NSString alloc] init];  
[string2 release];
```



@property

`@property (attributes) type name;`

`@property float value;`

`==`

- `(float)value;`
- `(void)setValue:(float)newValue;`

Попробуем!



Несколько аргументов

```
[myFraction setNumerator:10];  
[myFraction setDenominator:22];
```

Несколько аргументов

```
[myFraction setNumerator:10];  
[myFraction setDenominator:22];
```

```
[myFraction setNumerator:10 setDenominator:22];
```

Несколько аргументов

```
[myFraction setNumerator:10];  
[myFraction setDenominator:22];
```

```
[myFraction setNumerator:10 setDenominator:22];  
[myFraction setNumerator:10 andDenominator:22];
```


Несколько аргументов

```
[myFraction setNumerator:10];  
[myFraction setDenominator:22];
```

```
[myFraction setNumerator:10 setDenominator:22];  
[myFraction setNumerator:10 andDenominator:22];
```

```
[myFraction setTo:10 over:22];
```



Self

как 'this' в C++
указатель на текущий instance

Self

как 'this' в C++
указатель на текущий instance

Свой init

Категории

- Разделение определения класса на модули
- Расширение существующих классов

Категории

```
#import "ClassName.h"
```

```
@interface ClassName (CategoryName)  
    // method definitions  
@end
```

```
@implementation ClassName (CategoryName)  
    // method implementation  
@end
```

Категории

- Сначала нужно импортировать определение класса
- Нет лимита на категории
- Пара Класс-Категория должна быть уникальной



Расширение класса

- Категория без имени
- Позволяет задавать новые переменные

Расширение класса

```
#import "ClassName.h"

@interface ClassName (
    // variables definitions
    // method definitions
@end

@implementation ClassName
    // method implementation
@end
```


Расширение класса

- Категория может перезаписать оригинальный метод из класса. Не делайте этого! Метод класса будет потерян
- Если перезагрузка необходима – создайте подкласс. Оригинальный метод будет доступен через `super`



Методы NSObject

```
[myFraction isKindOfClass: [Fraction class]];
```

```
[myFraction isKindOfClass: [Fraction class]];
```

```
[myFraction respondsToSelector: @selector (setNumerator:)];
```

```
[Fraction instancesRespondToSelector:@selector (setNumerator:)];
```

```
[Fraction isKindOfClass: [NSObject class]];
```

Наследование

- В отличие от С – нет множественного наследования
- В целом, ничего нового

Обработка ИСКЛЮЧЕНИЙ

```
@try {  
    // something  
}  
  
@catch (NSException *exception) {  
    NSLog(@"Caught %@%@", [exception name], [exception reason]);  
}
```



**Давайте сделаем
брауер!**