

# Разработка приложений для OS X

Лекция 8

Core Data. NSPanel. Пользовательские настройки.

Hexlet University, 2012

# Проект 3

На выбор

- Вариант 1: продолжайте собственный проект
- Вариант 2: конкретное задание – iDNA
- Единый дедлайн – **25 декабря** (Merry Christmas!)

# Вариант I

- Бета-версия приложения
- Оно должно запускаться
- Большая часть функций должна быть реализована
- По большей части субъективная проверка

# Вариант 2: iDNA

- Эволюция для достижения Goal DNA
- Настройка размера популяции, ДНК и процента мутации
- Undo/redo изменений
- Сохранение/загрузка популяции
- Загрузка Goal DNA из текстового файла

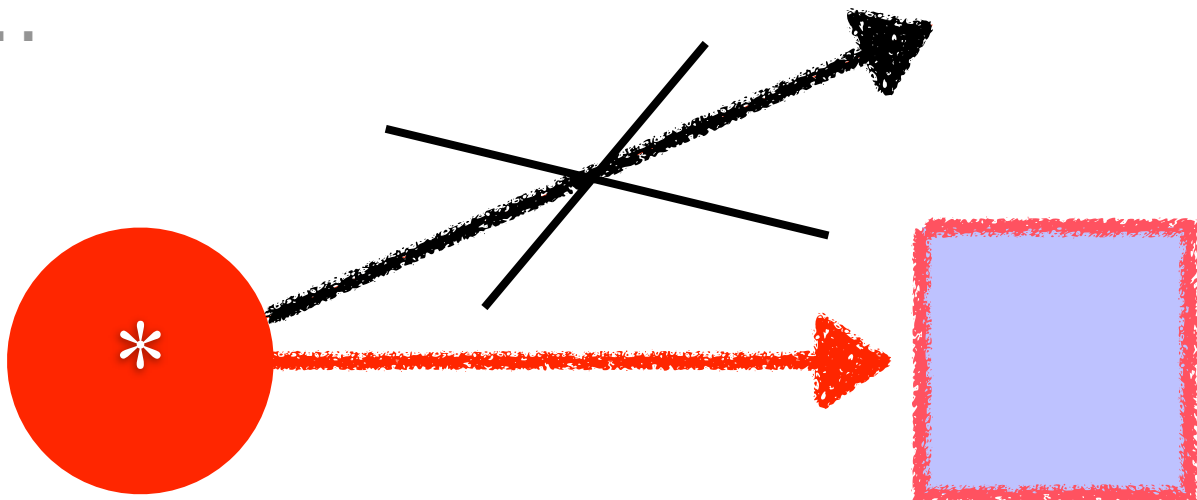




# Двойной указатель из прошлой лекции

```
- (BOOL)readFromData:(NSData *)data ofType:(NSString *)typeName error:(NSError **)outError
{
    ...
    @try {
        ...
    }
    @catch (NSException *e) {

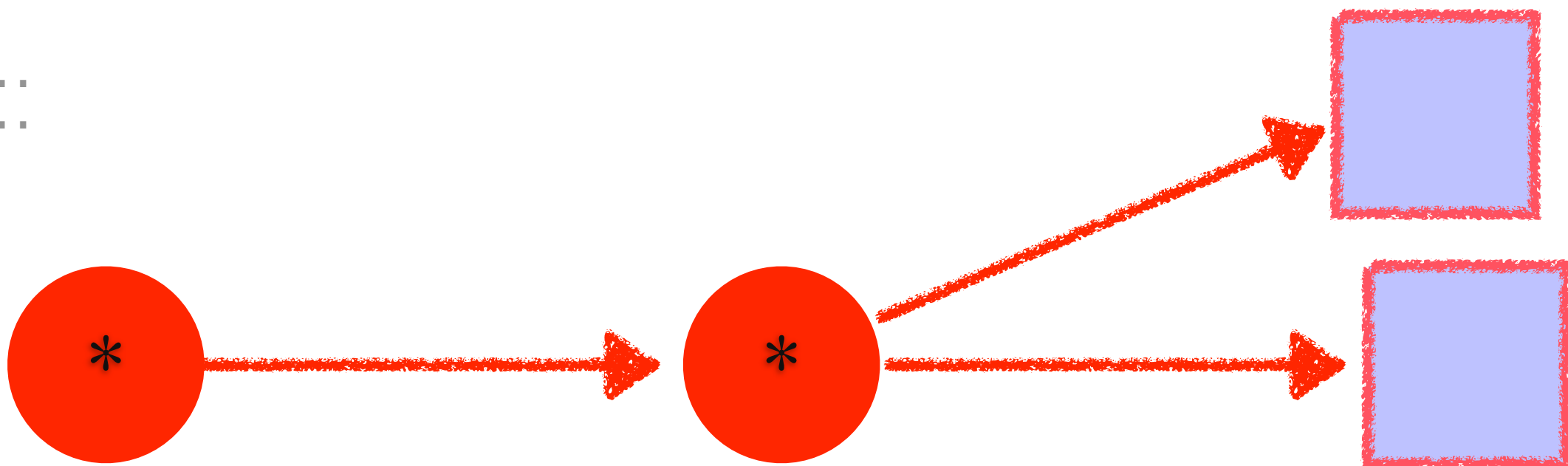
        ...
        *outError = [NSError errorWithDomain:NSOSStatusErrorDomain code:unimpErr userInfo:d];
        ...
    }
    ...
    ...
}
```



# Двойной указатель из прошлой лекции

```
- (BOOL)readFromData:(NSData *)data ofType:(NSString *)typeName error:(NSError **)outError
{
    ...
    @try {
        ...
    }
    @catch (NSEException *e) {

        ...
        *outError = [NSError errorWithDomain:NSOSStatusErrorDomain code:unimpErr userInfo:d];
        ...
    }
    ...
}
```



# Что не так с нашим приложением?

- NSDocument
- Классы для модели
- Переменные для свойств моделей
- Массив для хранения
- ArrayController



# Core Data

- **Persistent**-фреймворк
- Сериализация (архивация) данных в XML, SQL или двоичный формат
- Работает с SQLite
- Mac OS 10.4 (Tiger), iOS 3.0



# Создадим подкласс NSArrayController'a

- вместо контроллера для модели –  
контроллер для **нашей** модели
- чтобы контролировать создание объекта



# NIB-файлы и NSPanel

- NIB это скомпилированный XIB
- NSPanel это подкласс NSWindow
  - может быть key window не бывая main window
  - закрывается с помощью Esc



# User defaults database

- Приложения записывают пользовательские настройки в систему
- `~/Library/Preferences`
- Там хранятся настройки отличные от заводских
- `NSUserDefaults`

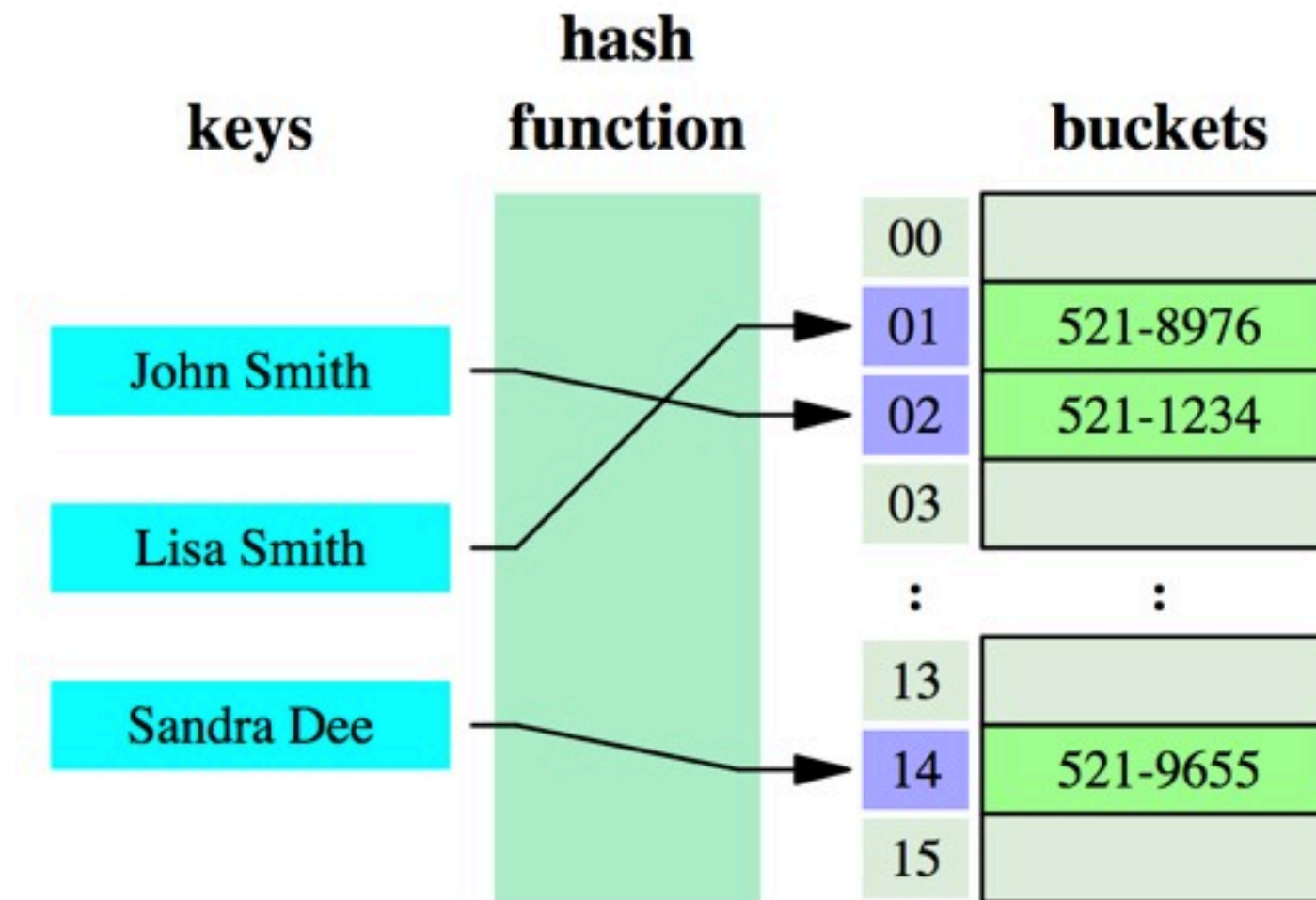
# Dictionaries (словари)

- KEY (ключ, string) – VALUE (указатель на объект)
- Ключ уникален. Получить объект по ключу:  

```
anObject = [myDictionary objectForKey:@"foo"];
```
- Если такого ключа нет – вернется nil
- NSMutableDictionary это подкласс NSDictionary

# NSDictionary

- реализован как hash table



# Некоторые методы NSDictionary

- (NSArray \*)allKeys; //массив всех ключей
- (unsigned)count; //количество пар ключ–значение
- (id)objectForKey:(NSString \*)aKey; //объект по ключу или nil

```
for (NSString *key in dict) {  
    NSLog(@"%@ -> %@", key, [dict objectForKey:key]);  
}
```

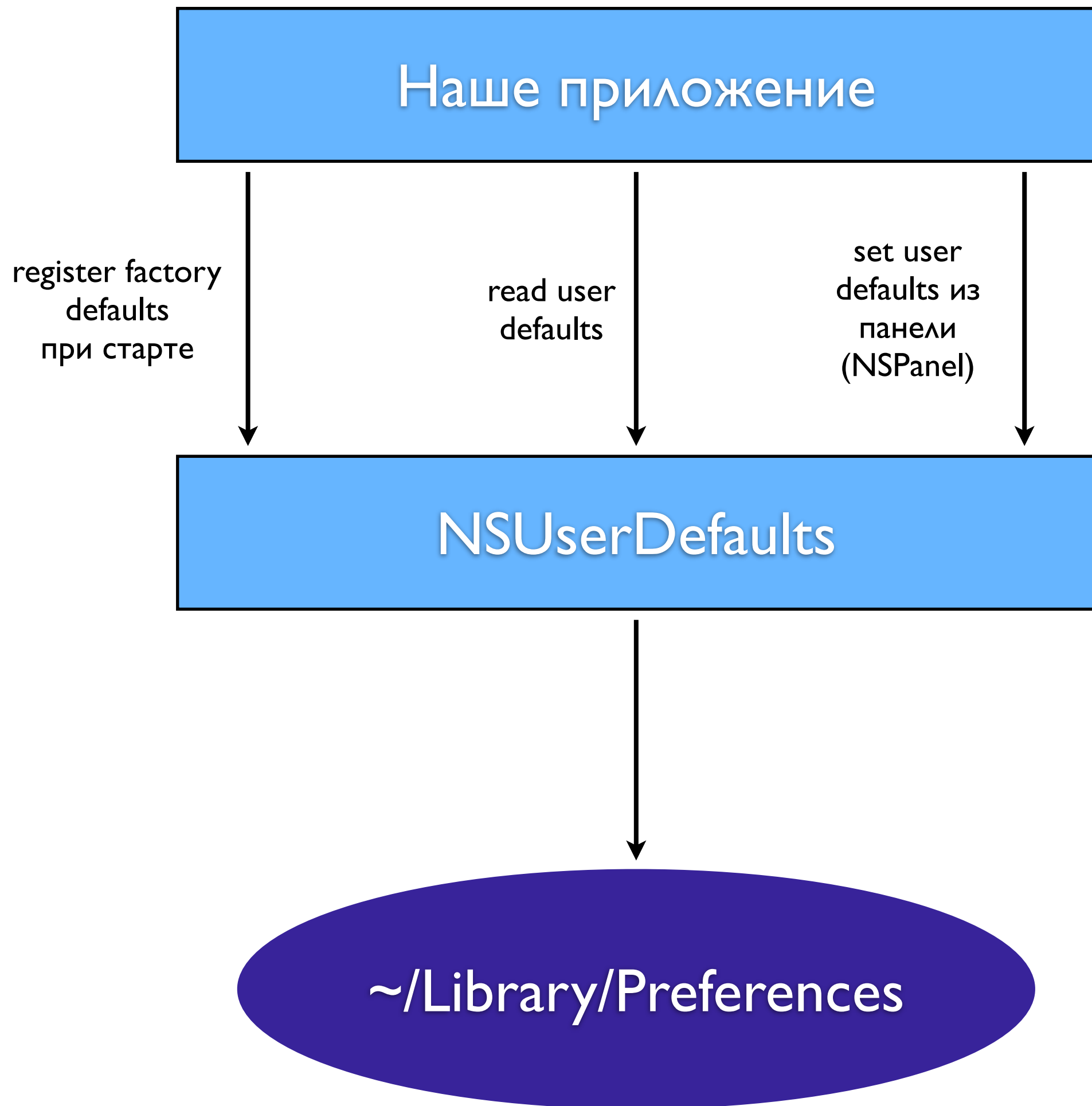
# Некоторые методы в NSMutableDictionary

- + (id)dictionary; // пустой словарь
- (void)removeObjectForKey:(NSString \*)aKey; // удалить объект по ключу и ключ
- (void)setObject:(id)anObject forKey:(NSString \*)aKey; // добавить объект по ключу



# NSUserDefaults

- Заводские настройки поставляются с приложением
- User settings это то, что отличается от заводских настроек
- Нужно получить заводские настройки при старте. Это **registering defaults**
- После этого нужно получить пользовательские. Это **reading and using the defaults**
- Наша панель будет вносить правки в пользовательские настройки. Это **setting defaults**



# Методы UserDefaults

```
+ (NSUserDefaults *)standardUserDefaults
// общий объект

- (void)registerDefaults:(NSDictionary *)dictionary
// Registers the factory defaults

- (void)setBool:(BOOL)value forKey:(NSString *)defaultName
- (void)setFloat:(float)value forKey:(NSString *)defaultName
- (void)setInteger:(NSInteger)value forKey:(NSString *)defaultName
- (void)setObject:(id)value forKey:(NSString *)defaultName
// Изменение пользовательских значений

- (BOOL)boolForKey:(NSString *)defaultName
- (float)floatForKey:(NSString *)defaultName
- (NSInteger)integerForKey:(NSString *)defaultName
- (id)objectForKey:(NSString *)defaultName
// Чтение настроек. Если нет пользовательских — возвращаются заводские

- (void)removeObjectForKey:(NSString *)defaultName
// удаление пользовательской настройки. Приложение будет использовать заводскую.
```

# Области

Application: user defaults database

Registered defaults: factory settings

# Области

Arguments: command line arguments

Application: user defaults database

Global: user global settings

Language: user language

Registered defaults: factory settings

