Девятая независимая
научно-практическая конференция
«Разработка ПО 2013»

23 - 25 октября, Москва

Принцип наименьшего удивления в разработке API приложения

Гордиенков Андрей

Немного о докладчике

- Фанат программирования и рассказов о программировании
- Ведет свой блог 3 года: статьи и видео – http://softblog.violet-tape.net
- Очень ленив, поэтому ищет пути как писать меньше, а делать больше

Почему хорошее API так важно?

- API может быть основным активом
  - Быстрое создание экосистемы
  - Удержание потребителей
- API является обязательством
  - Неудачное API генерирует поток заявок на поддержку
- Публичное API – навсегда. Почти.

Неочевидность как есть

- API для Альфа-банка. Неофициальное.

```
var api = new AlfabankAPI();
api.Connect("k000000", "qwerty");
Console.WriteLine("Balance: {0}", api.GetBalance());

var date1 = new DateTime(2013, 9, 1);
var date2 = DateTime.Now;
var account = "40700000000000000000";
var operations = api.GetMovementOnAccount(account, date1, date2);
```

Тяжелое наследие

- Задача: создать пункт меню «на лету». В таблице.

```
private void tralala() {
    var contextMenu = new ContextMenu();
    contextMenu.Items.Add("MyNewItem");
    gridColumn.ContextMenu = contextMenu;
}
```

- Ответ:

```
e.MenuInfo.View.RowCellMenuCustomizations
```

  - Нарушения закона Деметры

Тяжелое наследие

- Задача: получение коллекции багов из трекера

```
class Program
{
    static void Main(string[] args) {
        var server = new TfsTeamProjectCollection(new Uri("SEE-SEC(R) 2013"));
        server.GetService()
    }
}
```

```
(<no parameters>) : T
(Type serviceType):object
Gets the service object of the specified type.
serviceType: An object that specifies the type of service object to get.
```

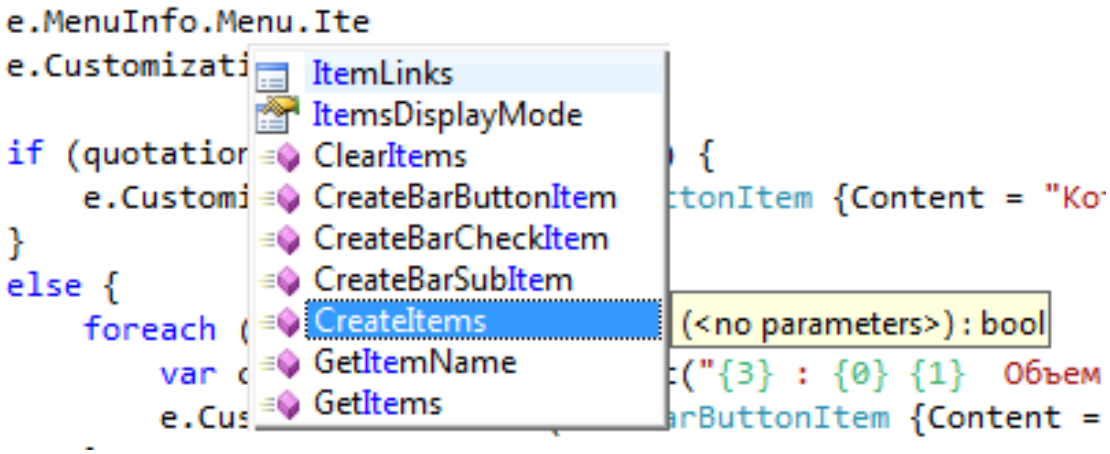
- Ответ:

```
class Program
{
    static void Main(string[] args) {
        var server = new TfsTeamProjectCollection(new Uri("SEE-SEC(R) 2013"));
        var service = server.GetService(typeof(WorkItemStore));
        // and so on...
    }
}
```

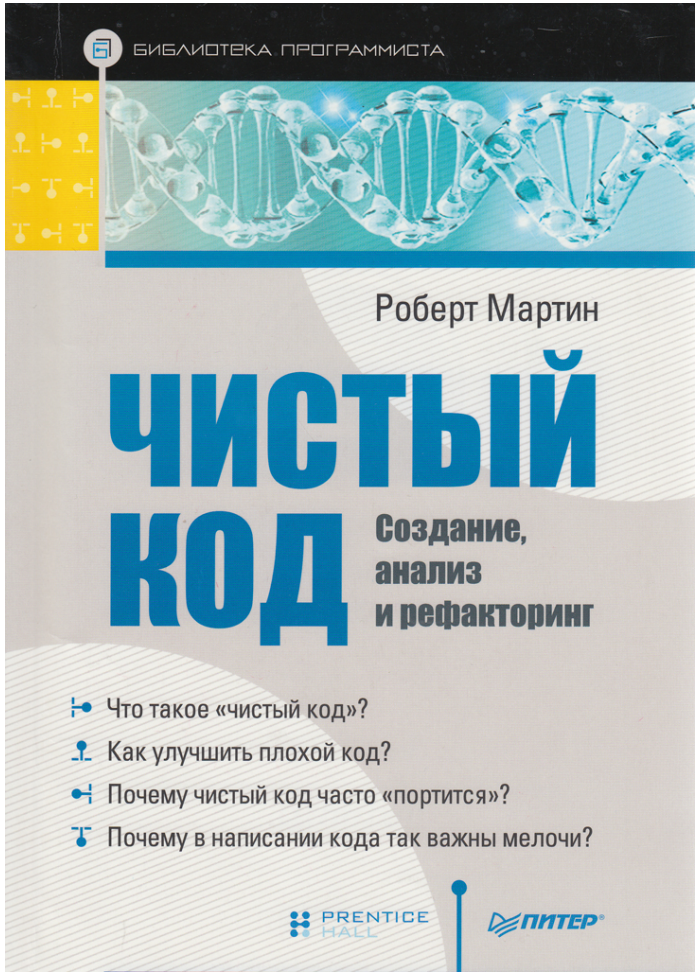
  - Ранняя проверка ошибок не производится

Тяжелое наследие

- Неочевидность использования



Многие решения здесь



Хорошее API

- Легко запомнить
- Легко использовать даже без документации
- Трудно использовать не по назначению
- Легко читать и поддерживать код
- Легко расширить
- Достаточно мощное для основных нужд
- Соответствует основным потребителям

Имена

Дан ScrollBox, какая функциональность у методов?

- void ShiftDown(float distance)
- void ShiftDownRelative(float distance)

Ответ:

- ShiftDown – сдвиг на N пикселей относительно текущей точки
- ShiftDownRelative – сдвиг на N процентов с начала документа

Имена

Переименовать:

- void ShiftDown(float distance)
- void ShiftDownFromStart(float distance, Measure measure)

Использование:

```
scrollBox.ShiftDown(400);
scrollBox.ShiftDownFromStart(400);
scrollBox.ShiftDownFromStart(25, Measure.Percent);
```

Излишняя сложность

- Система состояний UI компонента. «В лоб»

```
var visualState = new VisualState("StateName");
var transition = new VisualTransition();
var keySetter = new KeySetter("Background", Colors.Red);
transition.PropertySetters.Add(keySetter);
visualState.Transitions.Add(transition);
```

- Использование Fluent Interface

```
new VisualStateBuilder("StateName")
    .WithTransition()
    .WithProperty("Background", Colors.Red)
    .Build();
```

Излишняя сложность

- Анимация сдвига вниз

```
var storyboard = new Storyboard();
var propertyChange
     = new PropertyChange(1d, EasingMode.Linear, 500);
var propertyChangeSequence
     = new PropertyChangeSequence("Top", new []{propertyChange});
var uiAnimation = new UIAnimation(propertyChangeSequence);
storyboard.Add(uiAnimation, new []{border});
```

- Анимация сдвига вниз (с заботой о пользователе)

```
border.GlideDown(500);
```

Облегчение работы

Можно все настроить вручную…

```
map.controls
    .add('trafficControl')          // пробки
    .add('searchControl')           // поиск
    .add('zoomControl')             // зум-контрол
    .add('typeSelector')            // слои
    .add('geolocationControl')      // геолокация
    .add('fullscreenControl')       // фуллскрин
    …
```

Но можно положиться на разработчиков

```
map.controls.add('default');
```

Излишняя сложность

- Помогайте пользователю в изучении системы с помощью DSL, FluentInterface
- Упрощайте вызов частых методов

Кроссплатформенная согласованность

Яндекс.Диск названия методов:

- C#
  - `RemoveAsync`
  - `TrashAsync`
- ObjectiveC
  - `removeItemAtPath`
  - `trashItemAtPath`
- Java
  - `delete`

Правило трех для плагинов

- Напишите несколько плагинов до релиза
  - Если вы написали один, возможно другой не заработает
  - Если вы написали два, поддержка будет лучше, но с трудностями
  - Если вы написали три, все должно работать как надо

Основные практики

- Коридорное тестирование
  - Выявляет самые топорные методы и ситуации
- Автоматические тесты
  - Первые практические примеры работы кода
- Написание документации в коде, внешней
  - Позволяет выявить вещи посложнее, так как описывать бред тяжело
- Написание программ с вашим API другой группой
  - Eating your own dog food

K.O.

- Составление API тяжелая задача
  - Это не единичная активность в проекте
  - Совершенство недостижимо, но пытаться стоит!

Вопросы

Контакты

- Письма писать на my@violet-tape.net
- Поболтать можно в skype violet-tape