

# Жизнь мобильного разработчика

архитектура,  
тестирование,  
кросс-платформ



**“Архитектура программного обеспечения**  
(англ. *software architecture*) — это структура  
программы или вычислительной системы,  
которая включает программные компоненты,  
видимые снаружи свойства этих компонентов, а  
также отношения между ними.”

– [ru.wikipedia.org](https://ru.wikipedia.org)

**“Архитектура ПО** является реализацией  
нефункциональных требований к системе”

– [ru.wikipedia.org](https://ru.wikipedia.org)



# Примеры архитектурных моделей

- ✦ Монолитное приложение
- ✦ Клиент-серверная модель
- ✦ Распределенные вычисления
- ✦ Событийная архитектура
- ✦ Сервис-ориентированная архитектура



Напишем калькулятор! :-)



Текстовое  
поле

...

Кнопка 0

Кнопка 1

...

Кнопка +

Кнопка =

...



Текстовое  
поле

...

Кнопка 0

Кнопка 1

...

Кнопка +

Кнопка =

...

Логика  
сложения

...

Логика  
вычитания

...

Память



?

?

Текстовое  
поле

...

Кнопка 0

Кнопка 1

...

Кнопка +

Кнопка =

...

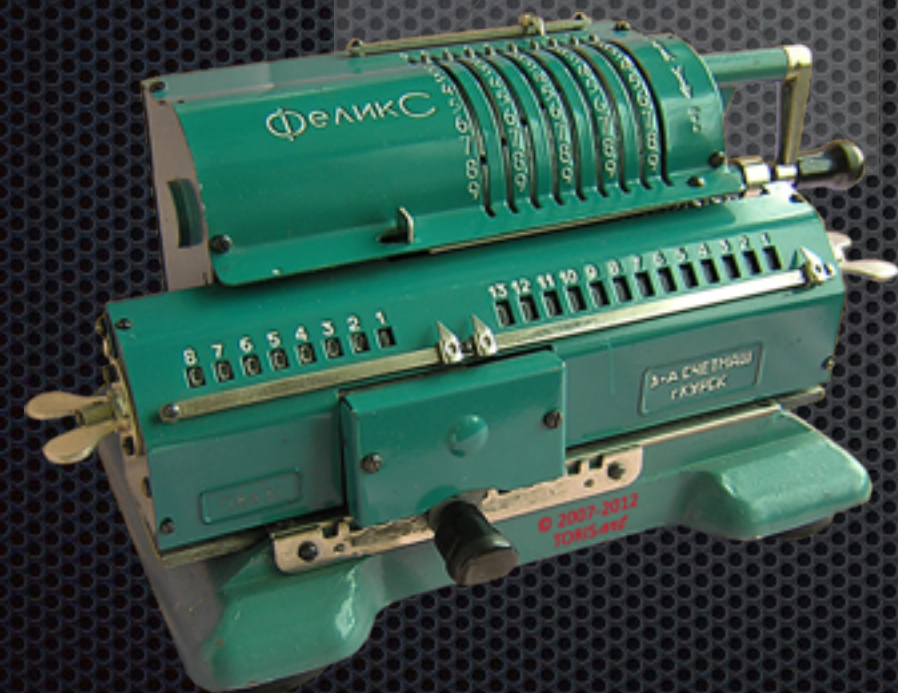
Логика  
сложения

...

Логика  
вычитания

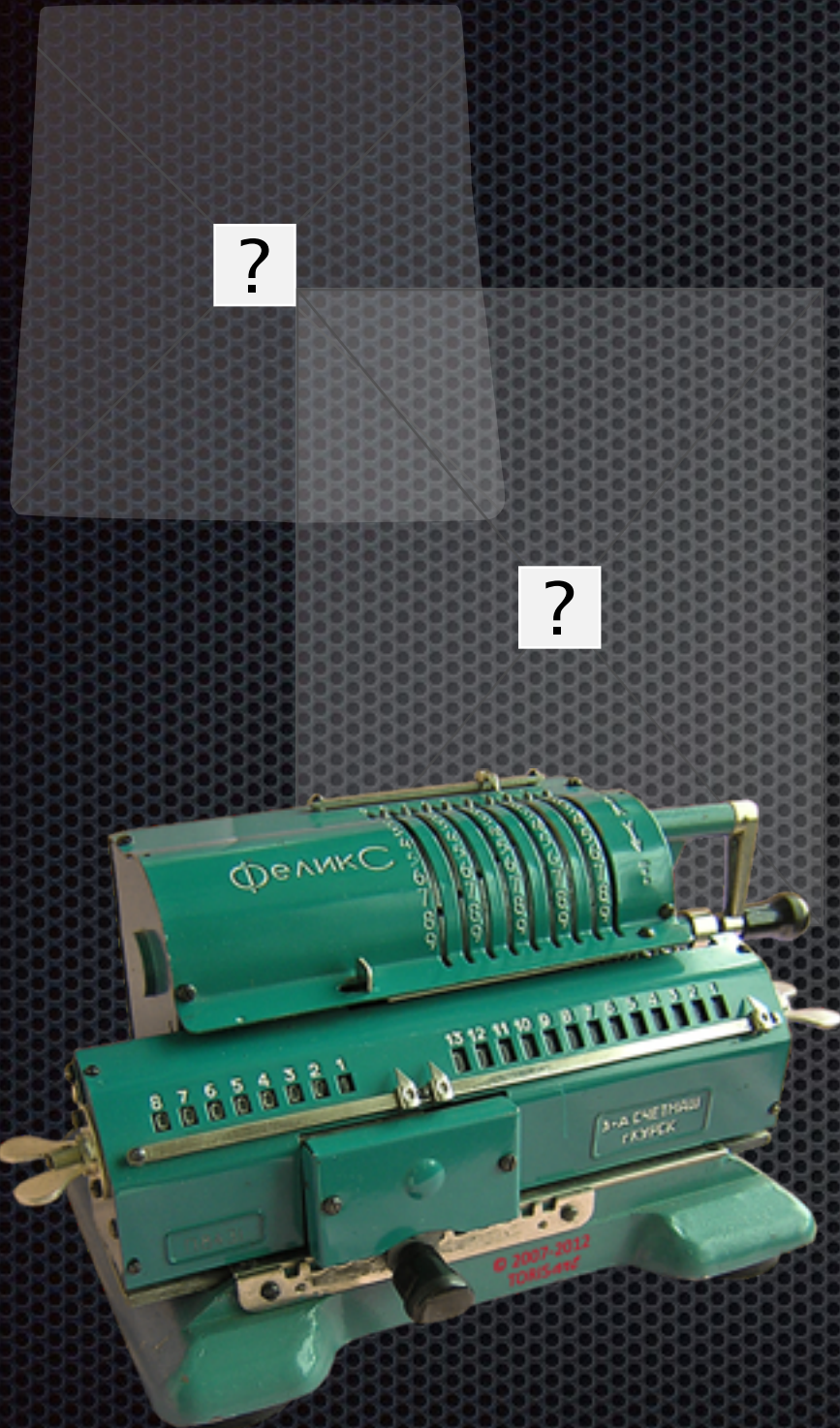
...

Память





# View



# ViewModel

Текстовое  
поле

...

Кнопка 0

Кнопка 1

...

Кнопка +

Кнопка =

...

# Model

Логика  
сложения

...

Логика  
вычитания

...

Память



# Model View ViewModel

- ✦ Модель - фундаментальные данные, необходимые для работы приложения
- ✦ Представление - графический интерфейс
- ✦ Модель представления - абстракция представления



И чо?



**“Тести́рование програ́ммного обеспéчéния** — процесс исследования, испытания программного продукта, имеющий две различные цели:

- продемонстрировать разработчикам и заказчикам, что программа соответствует требованиям;
- выявить ситуации, в которых поведение программы является неправильным, нежелательным или не соответствующим спецификации.”



# Тестирование и мы

- ✦ Модульное тестирование
- ✦ Приемочное тестирование



**“Модульное тестирование, или юнит-тестирование** (англ. *unit testing*) — процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы.”



# Модульное тестирование

- ✦ Поощрение изменений
- ✦ Упрощение интеграции
- ✦ Документирование кода
- ✦ Отделение интерфейса от реализации



**“Приёмочные тесты** (англ. *customer tests, acceptance tests*) — тесты, проверяющие функциональность приложения на соответствие требованиям заказчика. Приёмочные тесты проходят на стороне заказчика. Это помогает ему быть уверенным в том, что он получит всю необходимую функциональность.”



- ✦ Модульное тестирование позволяет определить, что вы сделали вещь «правильно».
- ✦ Приемочное тестирование позволяет определить, что вы сделали «правильную» вещь.



# Calabash

реальный код



**Feature:** Messenger

**Scenario:** User should be able to send messages

Given I am in the system

And I press the menu key

And I wait to see "Messages"

And I tap on "Messages"

And I wait to see "New message"

When I tap on "New message"

And I enter "Calabash test" into "Title" input field

And I enter timestamp into "Message" input field

And I tap on "Send"

Then I wait to see "Send" tab

When I tap on "Sent" tab

Then I wait to see entered timestamp



Говоришь-говоришь, а где же  
мобильный кросс-платформ?



The background is a dark blue gradient with a network of small, semi-transparent dots and thin lines in shades of blue and green, creating a digital or molecular aesthetic.

# Xamarin





appcelerator



21-й слайд ;-P