

F# в продакшне! Но почему?

РОМАН ЛИМАН

Сегодня в меню

2

- ▶ Как я пришел к F#
- ▶ Немного о нашем проекте
- ▶ Введение в синтаксис и основные принципы
- ▶ Продакшн кейс

В погоне за качеством кода

3

- ▶ Эта фишка нам нужна вчера!

В погоне за качеством кода

4

- ▶ Эта фича нам нужна вчера!
- ▶ Сначала пофикси баги!

В погоне за качеством кода

5

- ▶ Эта фича нам нужна вчера!
- ▶ Сначала пофикси баги!
- ▶ Где тот, кто это написал?!

В погоне за качеством кода

6

- ▶ Эта фича нам нужна вчера!
- ▶ Сначала пофикси баги!
- ▶ Где тот, кто это написал?!
- ▶ Опять null ref

В погоне за качеством кода

7

- ▶ Эта фича нам нужна вчера!
- ▶ Сначала пофикси баги!
- ▶ Где тот, кто это написал?!
- ▶ Опять null ref
- ▶ Тесты напиши

В погоне за качеством кода

8

- ▶ Эта фича нам нужна вчера!
- ▶ Сначала пофикси баги!
- ▶ Где тот, кто это написал?!
- ▶ Опять null ref
- ▶ Тесты напиши
- ▶ Тесты обнови

Все средства хороши?

9

- ▶ Шаблоны проектирования

Все средства хороши?

10

- ▶ Шаблоны проектирования
- ▶ Архитектурные подходы

Все средства хороши?

11

- ▶ Шаблоны проектирования
- ▶ Архитектурные подходы
- ▶ Конвенции

Все средства хороши?

12

- ▶ Шаблоны проектирования
- ▶ Архитектурные подходы
- ▶ Конвенции
- ▶ Процессы разработки

Не сработало

Proffsmagasinet

14

- ▶ <https://www.proffsmagasinet.se/>
- ▶ E-commerce
- ▶ F#, Azure, Azure Functions, Cosmos DB, Elastic Search etc.
- ▶ Legacy: C#, Azure, Virto Platform

Почему F#?

15

- ▶ Меньше кода
- ▶ Код более читаемый
- ▶ Код более надежный

Никакой магии

16

- ▶ Дополнительные абстракции

Никакой магии

17

- ▶ Дополнительные абстракции
- ▶ Удобные инструменты работы с ними

Никакой магии

18

- ▶ Дополнительные абстракции
- ▶ Удобные инструменты работы с ними
- ▶ Лаконичный синтаксис (Signal to noise ratio)

Никакой магии

19

- ▶ Дополнительные абстракции
- ▶ Удобные инструменты работы с ними
- ▶ Лаконичный синтаксис (Signal to noise ratio)
- ▶ Новые средства организации кода

F# функции

20

```
module FSharpIntro =  
  
    let add x y = x + y // x is int  
  
    // x and y are float  
    let add2 (x: float) y = x + y  
  
    let addWithPrint x y =  
        printfn "Adding %i and %i" x y  
        x + y
```

Partial application

21

```
... let add5 = add 5

... let seven = add5 2

... let doSomething (log: ILogger) arg =
... |   log.Log("Doing something")
... |   printfn "%A" arg
```

Оператор Pipe

22

```
... let pipeSample() =  
... | ... Console.WriteLine "some string"  
... | ... "some string" |> Console.WriteLine  
... | ... 4 |> add 3
```

Оператор Pipe

23

```
... let anotherPipeSample() =  
... |> [|1; 2; 3; 4; 5|]  
... |> Seq.sortByDescending (fun x -> x)  
... |> Seq.map(add 4) // partial application  
... |> Array.ofSeq
```

Discriminated Unions

24

```
···type CreditCardPaymentMethod =  
···| ··· OnSite of CreditCardInfo  
···| ··· Courier
```

```
···type PaymentMethod =  
···| ··· Cash  
···| ··· CreditCard of CreditCardPaymentMethod  
···| ··· WebMoney of string
```


Discriminated Unions

25

```
... let webMoney = WebMoney "1234123412341234"  
  
... let cash = Cash  
  
... let creditCardCourier = CreditCard Courier  
  
... let creditCardOnSite =  
... | ... OnSite creditCardInfo |> CreditCard  
...  
  
... let compileError = Cash "123"
```

Discriminated Unions

26

```
let processPayment paymentMethod amount =  
    match paymentMethod with  
    | Cash | CreditCard | Courier ->  
        async.Return (Ok())  
    | WebMoney acc ->  
        payWithWebMoney acc amount  
    | CreditCard (OnSite creditCardInfo) ->  
        payWithCreditCard creditCardInfo amount
```

Простейший аналог на C#

27

```
public enum PaymentMethodType
{
    WebMoney,
    CreditCardOnSite,
    CreditCardCourier,
    Cash
}

public class PaymentMethod
{
    public string WebMoney { get; set; }
    public CreditCardInfo CreditCard { get; set; }
    public PaymentMethodType Type { get; set; }
}
```

Простейший аналог на C#

28

```
...public class PaymentMethodValidator
...{
...    : AbstractValidator<PaymentMethod>
...    {
...        public PaymentMethodValidator()
...        {
...            RuleFor(x => x.WebMoney)
...                .NotNull()
...                .When(x => x.Type == PaymentMethodType.WebMoney);
...            RuleFor(x => x.WebMoney)
...                .Null()
...                .When(x => x.Type != PaymentMethodType.WebMoney);
...
...            RuleFor(x => x.CreditCard)
...                .NotNull()
...                .When(x => x.Type == PaymentMethodType.CreditCardOnSite);
...            RuleFor(x => x.CreditCard)
...                .Null()
...                .When(x => x.Type != PaymentMethodType.CreditCardOnSite);
...        }
...    }
}
```

Другой вариант на C#

29

```
...public abstract class PaymentMethodAdvanced
...{
...    public class Cash : PaymentMethodAdvanced { }

...    public class CreditCardCourier : PaymentMethodAdvanced { }

...    public class WebMoney : PaymentMethodAdvanced
...    {
...        public string AccountNumber { get; }
...        public WebMoney(string account)
...        {
...            AccountNumber = account;
...        }
...    }
...}
```

```
...    public class CreditCardOnSite : PaymentMethodAdvanced
...    {
...        public CreditCardInfo CreditCardInfo { get; }
...        public CreditCardOnSite(CreditCardInfo creditCardInfo)
...        {
...            CreditCardInfo = creditCardInfo;
...        }
...    }

...    public class Number : PaymentMethodAdvanced
...    {
...        public int Value { get; private set; }
...        public Number(int value)
...        {
...            Value = value;
...        }
...    }
...}
```

```
...public interface IAccountService
...{
...    void CreateAccount(string userId, AccountInfo accountInfo);
...    AccountInfo GetAccount(string accountId);
...    void UpdateAccount(AccountInfo accountInfo);
...    void DeleteAccount(string accountId);
...}
```

Option and Result

31

```
type Option<'T> =  
    | Some of 'T  
    | None  
  
type Result<'T, 'Error> =  
    | Ok of 'T  
    | Error of 'Error
```

```
type AccountOpenError =  
  | LowFicoScore of score: int  
  | CurrencyForbidden of currencyCode: string  
  | HighCreditLimit of limit: decimal  
  
type AccountCreationError =  
  | DataError of DataAccessError  
  | RuleViolation of AccountOpenError  
  | ValidationFailed of message: string
```


Продакшн кейс! Наканецта

Продакшн кейс! Наканецта

34

- ▶ У товара есть спецификации

Продакшн кейс! Наканецта

35

- ▶ У товара есть спецификации
- ▶ Спецификации да/нет (Наличие bluetooth)

Продакшн кейс! Наканецта

36

- ▶ У товара есть спецификации
- ▶ Спецификации да/нет (Наличие bluetooth)
- ▶ Числовые спецификации (Кол-во оборотов в минуту)

Продакшн кейс! Наканецта

37

- ▶ У товара есть спецификации
- ▶ Спецификации да/нет (Наличие bluetooth)
- ▶ Числовые спецификации (Кол-во оборотов в минуту)
- ▶ Текстовые спецификации

Продакшн кейс! Наканецта

38

- ▶ У товара есть спецификации
- ▶ Спецификации да/нет (Наличие bluetooth)
- ▶ Числовые спецификации (Кол-во оборотов в минуту)
- ▶ Текстовые спецификации
- ▶ Фиксированный список значений

Продакшн кейс! Наканецта

39

- ▶ У товара есть спецификации
- ▶ Спецификации да/нет (Наличие bluetooth)
- ▶ Числовые спецификации (Кол-во оборотов в минуту)
- ▶ Текстовые спецификации
- ▶ Фиксированный список значений
- ▶ Мульти значения

Доменная модель

```
...type AttributeValueType =
...| ... | ListOfValues
...| ... | Number
...| ... | FreeText

...type ComplexAttributeType =
...{
...    ValueType : AttributeValueType
...    IsMultiValued : bool
...    UnitId : string option
...}

...[<RequireQualifiedAccess>]
...type AttributeType =
...| ... | Boolean
...| ... | Rich of ComplexAttributeType

...type Attribute =
...{ Id : string
...  Type : AttributeType
...  Name : LocalizedString list }
```


Доменная модель

```
...type NumericSpecValue =  
...| JustNumber of float  
...| NumericValue of float * Unit * LocalizedString list  
  
...type RichTechSpecValue =  
...| NumericSpec of NumericSpecValue  
...| Text of (string * LocalizedString list)  
  
...type RichTechSpec =  
...{  
...    SpecValue: RichTechSpecValue list  
...    IsMultiValue: bool  
...}  
  
...type TechSpecValue =  
...| Boolean of bool  
...| Rich of RichTechSpec  
  
...type TechSpec =  
...{ Name: string  
...    Value: TechSpecValue  
...    DisplayNames: LocalizedString list }
```

Итоги перехода

42

- ▶ Меньше кода, он читаемый и стабильный

Итоги перехода

43

- ▶ Меньше кода, он читаемый и стабильный
- ▶ Написали и забыли

Итоги перехода

44

- ▶ Меньше кода, он читаемый и стабильный
- ▶ Написали и забыли
- ▶ 0 null reference

Итоги перехода

45

- ▶ Меньше кода, он читаемый и стабильный
- ▶ Написали и забыли
- ▶ 0 null reference
- ▶ Код легко адаптировать под новые требования

Итоги перехода

46

- ▶ Меньше кода, он читаемый и стабильный
- ▶ Написали и забыли
- ▶ 0 null reference
- ▶ Код легко адаптировать под новые требования
- ▶ 0 mocks

Как начать

47

- ▶ <https://fsharpforfunandprofit.com/>
- ▶ https://t.me/fsharp_chat
- ▶ <https://github.com/atsapura/CardManagement>
- ▶ https://t.me/fsharp_jobs