## Introduction to Haskell

Artem Ohanjanyan

LvivHaskell

24.11.2019

Artem Ohanjanyan (LvivHaskell)

## Haskell

Artem Ohanjanyan (LvivHaskell)

#### Introduction to Haskell

 ▲ ■
 ■
 ∽ へ 

 24.11.2019
 2/12

・ロト ・ 同ト ・ ヨト ・ ヨト

- Pure functional
- Lazy
- Statically typed
- With strong types and type inference
- Compiled (with an interpreter)

Artem Ohanjanyan (LvivHaskell)

< ■ ▶ ■ • ○ < ○</li>24.11.2019 3/12

イロト 不得 トイヨト イヨト

• Probably not to find a job (0 vacancies in Ukraine ③)

э 3/12 24.11.2019

▶ ∢ ⊒

- Probably not to find a job (0 vacancies in Ukraine ③)
- It's a challenge

→ ∢ Ξ

▶ ∢ ⊒

- Probably not to find a job (0 vacancies in Ukraine (3)
- It's a challenge
- Expand your mind with no illegal substances

- Probably not to find a job (0 vacancies in Ukraine ③)
- It's a challenge
- Expand your mind with no illegal substances
- Familiarize yourself with FP before everyone else

- Probably not to find a job (0 vacancies in Ukraine  $\odot$ )
- It's a challenge
- Expand your mind with no illegal substances
- Familiarize yourself with FP before everyone else
- Write programs (ecosystem is surprisingly good)

Artem Ohanjanyan (LvivHaskell)

▲ ■ ▶ ■ ∽ へ C
 24.11.2019 4 / 12

• Expressive

Artem Ohanjanyan (LvivHaskell)

▲ ■ ▶ ■ ∽ へ C
 24.11.2019 4 / 12

- Expressive
- Reliable

- Expressive
- Reliable
- Easy to support

▶ < E > < E</p>

## <u>Used</u> for

Artem Ohanjanyan (LvivHaskell)

3 5/12 24.11.2019

・ロト ・ 同ト ・ ヨト ・ ヨト

990

• Compilers

Artem Ohanjanyan (LvivHaskell)

▲ ■ ▶ ■ • ○ < ○</li>24.11.2019 5/12

イロト 不得 トイヨト イヨト

- Compilers
- Code analyzers

イロト イヨト イヨト

- Compilers
- Code analyzers
- Blockchain ¯\\_(い)\_/¯

Artem Ohanjanyan (LvivHaskell)

▲ ■ ▶ ■ ∽ Q C
 24.11.2019 5/12

< □ > < 四

▶ < E ▶ < E</p>

- Compilers
- Code analyzers
- Blockchain  $(\gamma)_{-}$
- Web backend

→ < Ξ

< □ > < 同

- Compilers
- Code analyzers
- Blockchain ¯\\_(い)\_/¯
- Web backend
- . . .

(4) (3) (4) (4) (4)

< □ > < 同

# Functional

Artem Ohanjanyan (LvivHaskell)

▲ ■ ▶ ■ • • • • • •
24.11.2019 6 / 12

・ロト ・ 同ト ・ ヨト ・ ヨト

• Broadly speaking, functional = Haskell

< E ► 문 ∽ < @ 24.11.2019 6/12

- Broadly speaking, functional = Haskell
- Functions are first-class citizens

▶ ∢ ⊒

- Broadly speaking, functional = Haskell
- Functions are first-class citizens
- Program is an expression rather than a list of instructions

## Pure

Artem Ohanjanyan (LvivHaskell)

#### Introduction to Haskell

24.11.2019 7/12

990

・ロト ・四ト ・ヨト ・ヨト 三日

### • Immutability

Artem Ohanjanyan (LvivHaskell)

▲ ■ ▶ ■ • ○ < ○</li>24.11.2019 7/12

イロト 不得 トイヨト イヨト

- Immutability
- No side effects

Artem Ohanjanyan (LvivHaskell)

▲ E ● E • つ Q C
 24.11.2019 7/12

イロト イヨト イヨト

- Immutability
- No side effects
- Result of the function is always the same



- Immutability
- No side effects
- Result of the function is always the same



- Immutability
- No side effects
- Result of the function is always the same

• Equational reasoning



- Immutability
- No side effects
- Result of the function is always the same

- Equational reasoning
- Refactoring



- Immutability
- No side effects
- Result of the function is always the same

- Equational reasoning
- Refactoring
- Parallelism



- Immutability
- No side effects
- Result of the function is always the same

- Equational reasoning
- Refactoring
- Parallelism
- Easier reasoning



### Artem Ohanjanyan (LvivHaskell)

#### Introduction to Haskell

### 24.11.2019 8 / 12

(日)

990



• User-defined control structures

< E ト ヨ のへで 24.11.2019 8/12

- User-defined control structures
- Infinite data structures

▶ ∢ ⊒

Image: A math a math

- User-defined control structures
- Infinite data structures
- Efficient functional programming for free (e.g. efficient higher-order functions)

- User-defined control structures
- Infinite data structures
- Efficient functional programming for free (e.g. efficient higher-order functions)
- However, sometimes it makes things more complicated



### Artem Ohanjanyan (LvivHaskell)

#### Introduction to Haskell

#### 9/12 24.11.2019

・ロト ・ 同ト ・ ヨト ・ ヨト

990



• Algebraic Data Types

Artem Ohanjanyan (LvivHaskell)

▲ ■ ▶ ■ ∽ へ C
 24.11.2019 9/12

イロト 不得 トイヨト イヨト



- Algebraic Data Types
- Powerful parametric polymorphism

▶ ∢ ⊒



- Algebraic Data Types
- Powerful parametric polymorphism
- Higher-kinded types



- Algebraic Data Types
- Powerful parametric polymorphism
- Higher-kinded types
- Type classes

э



- Algebraic Data Types
- Powerful parametric polymorphism
- Higher-kinded types
- Type classes
- Type inference



- Algebraic Data Types
- Powerful parametric polymorphism
- Higher-kinded types
- Type classes
- Type inference
- Many more



- Algebraic Data Types
- Powerful parametric polymorphism
- Higher-kinded types
- Type classes
- Type inference
- Many more
- $\bullet \ \mathsf{FP} + \mathsf{Types} = \mathsf{Win}$

< E ト ヨ つへの 24.11.2019 9/12

Artem Ohanjanyan (LvivHaskell)

#### Introduction to Haskell

24.11.2019 10 / 12

э

イロト 人間ト イヨト イヨト

- Treap a.k.a. randomized binary search tree
- Regular expression engine

## https://www.haskellstack.org

э 11/12 24.11.2019

э

# End

Artem Ohanjanyan (LvivHaskell)

24.11.2019 12 / 12

996

◆□ → ◆□ → ◆臣 → ◆臣 → □ 臣

Thanks for your attention!

▲ ■ ▶ ■ • つへで 24.11.2019 12 / 12

イロト 不同 トイヨト イヨト