

# Декларативное программирование

Семинар №1, группа 22215


---

Завьялов А.А.

5 сентября 2022 г.

Кафедра систем информатики ФИТ НГУ

## Завьялов Антон (Алексеевич)

- a.zavyalov@g.nsu.ru
-  [https://t.me/arx\\_dukalis](https://t.me/arx_dukalis)
- + 7 (960) 956 61 14



[🔗 https://bit.ly/haskell22215](https://bit.ly/haskell22215)

## Знакомство с языком программирования Haskell

- Установка и настройка инструментария
- Введение в синтаксис и базовые типы данных

# Вводные данные о Haskell

## Немного про язык

- Больше академический, чем промышленный
  - девиз – "avoid **success at all costs**"<sup>1</sup>
  - или всё-таки "**avoid success** at all costs"?
- Назван в честь математика и логика Хаскелла Карри
- Математическая база – типизированное  $\lambda$ -исчисление 2-го порядка
- Стандартизованный<sup>2</sup>, одна официальная реализация – Glasgow Haskell Compiler
- Компилируемый
- Интерпретируемый

---

<sup>1</sup>пер. – "избегать успеха любой ценой"

<sup>2</sup>см. <https://www.haskell.org/onlinereport/haskell2010/>

Набор инструментов Haskell состоит из

- Glasgow Haskell Compiler (GHC) – компилятор
- Cabal – пакетный менеджер и система сборки
- Stack – (еще один) пакетный менеджер и система сборки

Рекомендуемый способ установки тулчейна – GHCup

<https://www.haskell.org/ghcup/>

- Visual Studio Code
  - есть хороший **плагин**
- Emacs
- (Neo)Vim
- ...
- Leksah
- `http://repl.it/`
  - онлайн-редактор

- `ghc` – компилирует исходный код в объектный/исполняемый файл
- `runghc`, `runhaskell` – выполняет переданный код
- `ghci` – запускает сессию REPL, интерактивное окружение для разработки



## Что может делать GHCi

- вычислять выражения
  - печатать информацию об их типах
- выполнять программы в режиме интерпретации
- загружать модули, *скомпилированные* GHC в машинный код

---

<sup>3</sup>REPL (от англ. "read-eval-print loop") – простейшая организация интерактивной среды программирования в рамках интерфейса командной строки, выраженная в повторении последовательности чтения пользовательского кода, его исполнения и печати результатов в поток вывода.

## Hackage ([hackage.haskell.org](http://hackage.haskell.org))

- Хранилище пакетов Haskell (Как PyPI для Python)
- Можно, например, изучить документацию к базовому набору библиотек

## Hoogle ([hoogle.haskell.org](http://hoogle.haskell.org))

- Поисковый движок для кода на Haskell

# Вводные данные о Haskell

## Язык

- Функциональный
  - любая программа представлена комбинацией *функций* (в математическом смысле)
- Чистый
  - функции **детерминированные**<sup>4</sup>
  - функции не имеют **побочных эффектов**<sup>5</sup>
- Ленивый (по умолчанию)
  - *aka нестрогие вычисления, отложенные вычисления*
  - Вычисления производятся только по необходимости

---

<sup>4</sup> для заданных входных данных функция всегда возвращает одинаковый результат

<sup>5</sup> не ссылаются на данные, значения которых могут измениться в процессе работы программы и не производит таких изменений

## Типизация

- Статическая
  - типы проверяются до исполнения программы
- Сильная
- Неявная
  - вывод типов

## Полиморфизм

- параметрический
- ad-hoc

# Что почитать?

- Изучай Haskell во имя добра! – Миран Липовача
  - Бесплатно, на английском:  
`http://learnyouahaskell.com/`
- Програмируй на Haskell – Уилл Курт
- **Programming with Lambda Calculus** – Helmut Brandl

## Q&A

---