

# Функционално програмиране

упражнение 1, група 2

Християн Невелинов (Христов), email: nevelinov@uni-sofia.bg

## Парадигми за програмиране:

- процедурна
  - императивен стил
  - процедури със странични ефекти
- обектно-ориентирана
  - императивен стил
  - методи със странични ефекти
  - обектите имат *поведение* (методи) и *състояния* (член-данни)
- функционална
  - декларативен стил
  - без странични ефекти (при чисто функционалните езици)

## Особености на функционалната парадигма:

- по-кратки програми
- липсват странични ефекти -> липсва състояние -> няма необходимост от синхронизация при многонишков достъп
- доказателство на свойства и коректност на програмите
- програмите могат да се разглеждат като трансформация на входните данни
- липсва оператор за цикъл, симулира се чрез рекурсия
- по-добра организация на кода се постига чрез дефинирането на функции

## Програмиране на Haskell:

- ще използваме за редактор SciTE (произнася се *скайт*, от англ. Scintilla based Text Editor)
  - **Важно:** когато създавате файлове, те трябва да бъдат записани с разширение .hs
  - **Важно:** индентацията е от значение
  - компилиране на програмата: F5
  - изчистване на изхода: shift + F5

## Типове:

- цели числа: 2, 3, 8, ..
- реални числа: 3.14, 36.7, ..
- булеви променливи: True, False
- символи (*заграждат се с единични кавички*): 'a', 'b', ..
- низове (*заграждат се с двойни кавички*): "haskell", "programming", ..

## Основни оператори:

- аритметични операции: +, -, \*
- деление:
  - частно при целочислено деление: div
  - остатък при целочислено деление: mod
- степенуване:
  - повдигане в степен естествено число: ^ -- 2 ^ 3
  - повдигане в степен реално число: \*\* -- 2 \*\* 0.5
- абсолютна стойност: abs
- логически оператори:
  - конюнкция: &&
  - дизюнкция: ||
  - отрицание: not

## Условни изрази:

- if <condition> then <if-condition-is-true> else <otherwise>

Пример:

```
if (37 `mod` 2 == 0) then "Even" else "Odd"
```

## Оператори за сравнение:

- >
- >=
- ==
- /=
- <=
- <

## Декларация и дефиниция на функция:

- общ вид

## Задачи:

**Задача 1.** Да се дефинира функция `inside`, която проверява дали числото  $x$  принадлежи на интервала  $[a, b]$ .

**Задача 2.** Да се дефинира функция `sumSquares`, която намира сумата на квадратите на числата  $a$  и  $b$ .

**Задача 3.** Да се дефинира функция `myfact`, която пресмята факториела на число. Да се напише и итеративно решение.

**Задача 4.** Да се дефинира функция `myfib`, която връща  $n$ -тото число от редицата на Фибоначи (редицата е 1, 1, 2, 3, 5, ... и е индексирана от 0). Да се напише и итеративно решение.

**Задача 5.** Да се дефинира функция `myGcd`, която връща НОД( $a, b$ ).