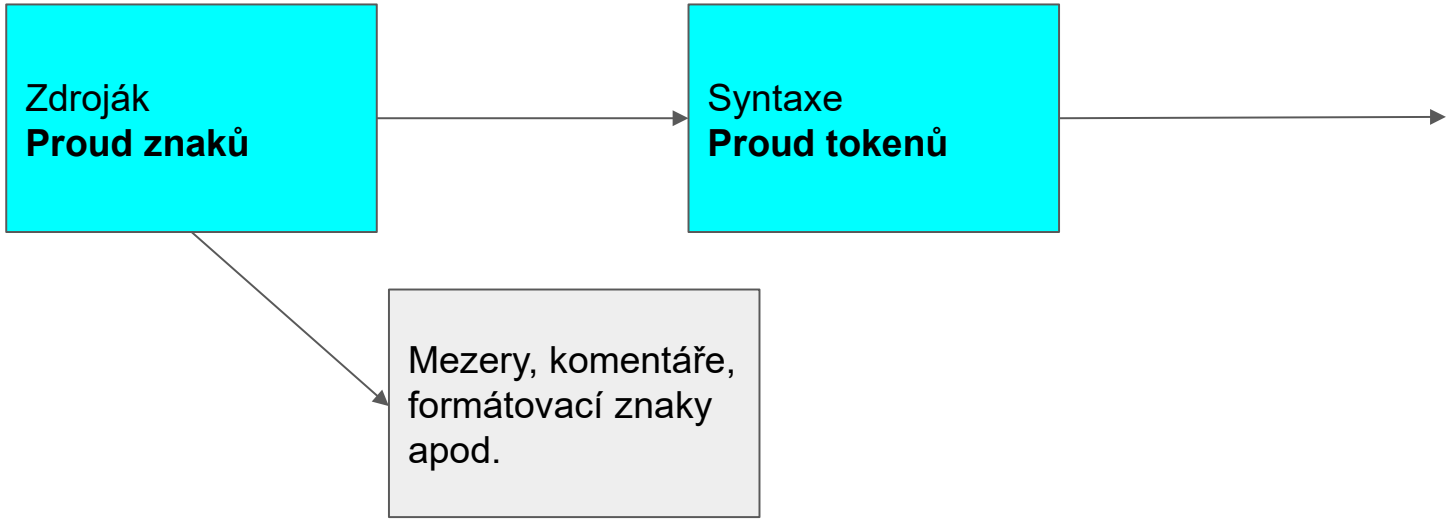
Lexikální analýza

NTI/PRK, LS 2022

Co je cílem


Zdroják
Proud znaků
Syntaxe
Proud tokenů
Mezery, komentáře, formátovací znaky apod.

Co je “token” (“lexem”)?

“V programovacích jazycích jsou slovní druhy objekty jako identifikátory, číselné hodnoty, klíčová slova a podobně. Těmto entitám se tradičně říká tokeny.”

(ICD, str. 1)

Tokeny odpovídají nějakým vzorcům; jsou nahrazovány identifikátory tokenů => Regulární výrazy.

Tokeny jsou terminální symboly gramatiky jazyka.

Regulární výraz

- Definice vzorů
- Regulární jazyky
- Konečné automaty

Syntaktický vzorec $(^n)^n$ popisuje neregulární jazyk => závorky jsou samostatné tokeny.

Co se musí udělat?

- Najít ve zdrojáku vzorce:
  - Identifikátory
  - Hodnoty (řetězce, čísla…)
  - Klíčová slova
  - Volání funkcí
  - Interpunkci (závorky, středníky…)
  - Operátory
  - Další tokeny

Lexikální analýza

Hledej a nahraď:

var MyIdentifier = 13;

KeyVar VarDef1 AssignmentOperator IntegerVal1;

Výstupy lexikální analýzy

1. **Proud tokenů** pro syntaktickou analýzu.
2. První verze **tabulky symbolů**.
3. **Chybová hlášení**

Tabulka symbolů

Propojení vašich identifikátorů s tokeny

- Názvy proměnných a konstant
- Názvy procedur a funkcí
- Konstanty
- Překladačem generované dočasné symboly
- Návěstí ve zdrojovém jazyce (label) - např. assembler

Co si musíme pamatovat?

- Identifikátor a datový typ
- Deklaraci procedury
- Offset v úložišti
- Ukazatel na začátek (objekty, struktury)
- Zda se parametry předávají hodnotou či odkazem
- Počet a typy argumentů ve volání funkcí
- Základní adresu (funkce) - kvůli skoku a návratu

Lexikální analýza - sofware

- Definujte vzorce
- Nástroje vždy konfigurujeme
  - Regulární výrazy a vzory
  - Co se má udělat, při zachycení vzoru
- Skoro vždy C++ nebo Java
  - Lex/Flex (C++, mostly used in Unix/Linux systems)
  - ANTLR (Java)
  - JavaCC (Java)
  - ANTLR a JavaCC jsou novější a inspirovány programem Lex