

# IFJ 2020

Tým 005 - varianta I

Petr Růžanský - xruzan00 - vedoucí

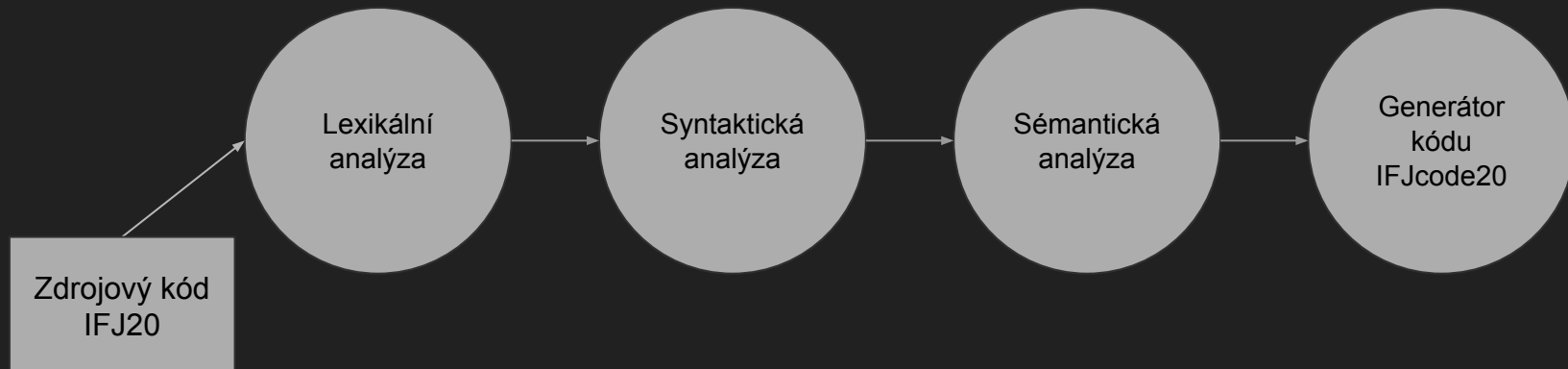
Radek Maňák - xmanak20

Adrián Bobola - xbobol00

Matúš Vráblik - xvrabl05

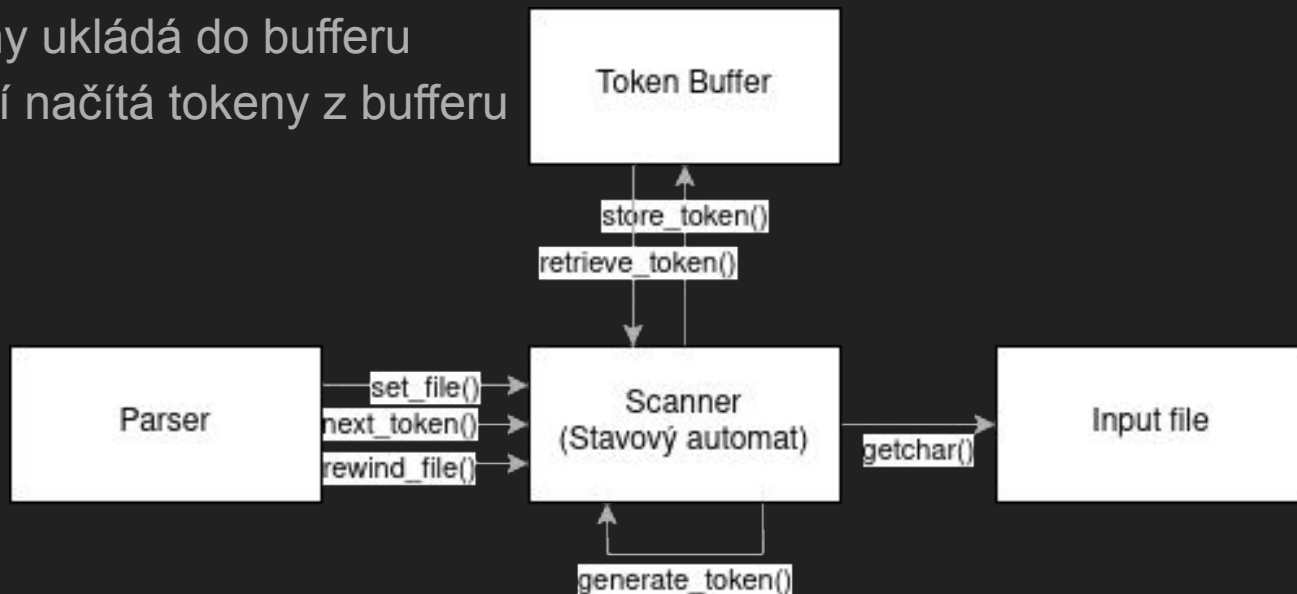


# Struktura překladače



# Lexikální analýza

- Implementovaná pomocí stavového automatu
- Vytvořené tokeny ukládá do bufferu
- Při druhém čtení načítá tokeny z bufferu



# Token

Token
+ token_type: Token_type
+ attribute: Token_attribute

Token_attribute (union)
+ string: char*
+ integer: int64_t
+ floating: double

# Syntaktická analýza (bez zpracování výrazů)

- Rekurzivní volání neterminálů
  - (metoda shora-dolů)
- Řízena LL(1) tabulkou
- Střídání dvou maker dle LL tabulky:
  - **CHECK\_AND\_LOAD\_TOKEN**( *očekávaný\_typ\_tokenu* )
    - Zkontroluje typ daného tokenu a zavolá další
  - **CHECK\_AND\_CALL\_FUNCTION**( *neterminál* )
    - Zavolá daný neterminál (funkci)
    - zkontroluje jeho návratový typ

# Zpracování výrazů

- Použití precedenční analýzy
- Přímé generování kódu při redukčních krocích
- Všechny mezivýsledky uloženy do nové proměnné
- Proměnné s výsledkem uloženy na zásobník výsledků
- Předání výsledku a řízení rekurzivnímu sestupu

# Sémantická analýza

- Dva průchody kódem
- První sbírá informace o funkcích
- Funkce uloženy v globální tabulce symbolů
- Při druhém průchodu ověřuje datové typy, deklarace
- Proměnné uloženy do tabulky symbolů na vrcholu zásobníku tabulek

Symtable item
+ token: Token
+ dataTypes: Datatype[]
+ isfunction: bool
+ return_values_count: int
+ parameters: Parameter*
+ parameter_count: int

Parameter
+ identifier: char*
+ datatype: Data_type

# Generátor kódu IFJcode20

- Přímé generování výstupního kódu
- Tvořen spolu s sémantickou analýzou =>  
Funkce generují i několik instrukcí najednou dle potřeby
- Vestavěné funkce
- Návěští a pomocné proměnné výrazů jsou očíslovány  
pro zabránění opakované deklarace



# Zdroje

- Přednášky IFJ a IAL
- Materiály na wiki stránkách IFJ

# Děkujeme za pozornost