# Syntaktický a sémantický analyzátor

Soubor funkcí kontrolující syntaktickou a sémantickou správnost vstupního kódu IFJ17.

## Princip

Syntaktický a sémantický analyzátor se skládá z několika funkcí, které odpovídají jednotlivým možným strukturám v jazyce IFJ17. Každá daná funkce načítá tokeny z lexikálního analyzátoru, které následně kontroluje a dále zpracovává, popřípadě volá další funkce syntaktického a sémantického analyzátoru, které zpracovávají potřebné použité struktury (výrazy, vnořené příkazy apod.).

## Řešení

Syntaktický a sémantický analyzátor se skládá z 19 hlavních funkcí, které slouží ke kontrole syntaktické a sémantické správnosti vstupního kódu a několika dalších pomocných funkcí, které jsou používány pro zjednodušení opakovaných akcí napříč všemi hlavními funkcemi.

### Spuštění překladu

Zpracování začíná v hlavní funkci **Parser\_ParseInitial,** která vytvoří výchozí úvodní instrukce výstupního programu a dále cyklicky zpracovává příchozí tokeny. Tato funkce dovoluje použití následujících struktur: *deklarace funkce*, *definice funkc*e, *definice scope*. Tato úvodní funkce pracuje na stejném principu jako funkce pro zpracování vnořeného kódu, přijímá pouze značně omezený seznam typů tokenů

Po přijetí očekávaných klíčových slov volá odpovídající funkce, které načítají další tokeny, kontrolují jejich správnost a dále je zpracovávají. Při přijetí tokenu neočekávaného typu analyzuje chybu a vrací odpovídající chybový kód hlavní funkci celého programu.

### Vnořený kód

Funkce pro zpracování obecného vnořeného kódu (**Parser\_ParseNestedCode**) slouží jako křižovatka, která na základě prvního příchozího tokenu volá odpovídající funkci pro zpracování očekávané struktury. Dále také rozhoduje, zda je vůbec možné v dané situaci danou strukturu použít – tato rozhodnutí dělá především na základě seznamu zanoření, který určuje, v jaké struktuře se aktuálně vnořený kód zpracovává (zda se jedná o větev podmínky, cyklus či funkci).

V případě, že tato funkce obdrží token, který by mohl mít speciální význam – klíčové slovo, které nebylo očekáváno a pro vnořený kód nemá žádný smysl (ovšem může se jednat o ukončení nadřazené struktury – konec definice funkce, cyklu apod.), vrací se o úroveň výše, a funkci, která jej zavolala, tento token předává. Ta rozhodne o jeho správnosti.

### Zpracování funkcí

O zpracování funkcí v programu se starají 3 funkce, kde každá z nich má konkrétní účel – jedna funkce pro deklaraci funkce (**Parser\_ParseFunctionDeclaration**), jedna pro její definici (**Parser\_ParseFunctionDefinition**) a jedna zpracovává volání těchto funkcí (**Parser\_ParseFunctionCall**).

Funkce pro deklaraci a definici funkcí jsou volány z hlavní úvodní funkce syntaktického a sémantického analyzátoru. Funkce pro volání funkcí jsou volány z funkcí pro zpracování výrazů.

### Zpracování podmínek

Podmínky zpracovávají dvě funkce – první (**Parser\_ParseCondition**), která zpracovává prvotní podmínku a ELSE bez další následující podmínky, a druhá (**Parser\_ParseSubCondition**), která zpracovává další libovolný počet navazujících ELSEIF podmínek.

Druhá funkce pro zpracování navazujících ELSEIF podmínek zpracovává pouze tyto typy – v případě, že obdrží token značící kompletní ukončení podmínky či začátek ELSE větve, vrací se postupně na nejvyšší úroveň, dokud se nevrátí do první funkce, která se o daný token postará sama.

### Zpracování cyklů

### Zpracování výrazů

## Pomocné datové struktury

### Zásobník

### Seznam

#### Postfixový seznam

#### Seznam zanoření