

```

struct Uzel { int Typ;
              union { struct { Uzel *prvni,*druhy,*treti,*ctvrty; } z;
              int Cislo;
              const char *Retez;
              int *Adresa; } z; };

```

Strukturovaný typ *Uzel* je použit pro generování interní formy při sémantické analýze.

```

Uzel *GenUzel(int,Uzel *,Uzel * =0,Uzel * =0,Uzel * =0);

```

Funkce je použita pro generování uzlu interní formy.

```

Uzel *GenCislo(int);

```

Funkce je použita pro generování uzlu interní formy pro číslo.

```

Uzel *GenRetez(const char *);

```

Funkce je použita pro generování uzlu interní formy pro řetěz.

```

Uzel *GenPromen(const char *);

```

Funkce je použita pro generování uzlu interní formy pro proměnnou.

```

#define TABSIZE 4

```

Velikost horizontálního tabulátoru (kolik reprezentuje mezer).

```

union YYSTYPE { unsigned c; char *r; Uzel *u; };

```

Členské proměnné *c* a *r* jsou pro uložení hodnot rozpoznaných lexikálních symbolů. Proměnná *c* je pro uložení čísel, proměnná *r* je ukazatel na uložený řetězec.

Proměnná *u* je využita pro generování interní formy po sémantické analýze.

```

#define YYSTYPE_IS_DECLARED

```

Informace pro lexikální analyzátor, že datový typ YYSTYPE je definován uživatelem.

```

enum Pozice { BEZPOZICE,RADEK,SLOUPEC };

```

Je používáno při výpisu chyb při analýze (překladu) zdrojového programu. Udává, zda text zprávy o chybě má, či nemá být doplněn údajem o pozici, kde byla chyba zjištěna.

BEZPOZICE – text zprávy je bez udání pozice.

RADEK – u textu chyby je uvedeno, na kterém řádku ve zdrojovém programu se vyskytla.

SLOUPEC – u textu chyby je uvedeno, na kterém řádku a dále ve kterém sloupci byla chyba zjištěna.

```

void Chyba(const char *,Pozice=SLOUPEC);

```

Funkce pro výpis zprávy o chybě.

```

void LexInic();

```

Funkce pro inicializaci proměnných před spuštěním lexikální analýzy.

```

void Flush();

```

Funkce pro vyprázdnění (vymazání) vstupní vyrovnávací paměti před otevřením souboru s popisem lexikálního analyzátoru při generování lexikálního analyzátoru.

```
#include "MikroC.tab.hpp"
```

Soubor s popisem označení lexikálních symbolů, které mají délku větší než jeden znak.