Úkoly ke kapitole 4

1. Je dán programovací jazyk s proměnnými těchto datových typů:

   - celá čísla,
   - racionální čísla,
   - řetězce,
   - pravdivostní hodnoty (boolean).

   Naprogramujte tabulku symbolů a její přístupové funkce pro tento jazyk (předpokládejme, že nemá blokovou strukturu). Jazyk má být interpretován, proto do tabulky ukládejte také hodnoty proměnných.

2. K jazyku z úkolu 1 přidejme názvy funkcí, které mají návratovou hodnotu, parametry se nepoužívají. Funkce mohou být vnořené, každá může mít vlastní lokální proměnné. Naprogramujte tabulku symbolů a její přístupové funkce.

3. Převeďte výraz $x = 3 - 2 + 1$ do všech tří základních druhů intermediálního kódu. Zkontrolujte výpočtem v intermediálním kódu, zda máte operátory ve správném pořadí.

4. Převeďte výraz $x = 25 - y * (8 + z)/2 + 5 * x$ do všech tří základních druhů intermediálního kódu.

5. Převeďte výraz IF $x < y + 2$ THEN $x := y + 2$ ELSE $x := z$ do všech tří základních druhů intermediálního kódu.

6. Jak víme, při konstrukci tabulky symbolů se často využívají metody známé z databází. Promyslete si, jak tímto způsobem (pomocí primárních klíčů) lze co nejvíce eliminovat neustálé porovnávání řetězců při procházení tabulky symbolů.

   Předpokládejme, že při zpracování deklarace proměnné (případně funkce nebo dalších možných prvků, které řadíme do tabulky symbolů) je této proměnné přiřazena nová hodnota primárního klíče reprezentovaná datovým typem, který se snadno porovnává (zřejmě číslem). V tabulce musíme během syntaktické analýzy evidovat také řetězcovou reprezentaci názvu proměnné, kterou používáme při zjištění proměnné mimo deklaraci, a nebo použijeme indexový soubor (seznam).

   Podle výše naznačeného řešení upravte kód tabulky symbolů pro interpretaci v příkladu 4.3 na straně 100. Přinese toto řešení zrychlení výpočtu i v případě, že v jazyce nejsou cykly a tedy nedochází k opakovanému vyhodnocování kódu dynamickou sémantikou?