Volání hodnotou (1)

- Skutečným parametrem musí být výraz
- Substituce skutečného parametru za formální probíhá takto:
  - výraz (skutečný parametr) se vyhodnotí
  - formální parametr se stane lokální proměnnou, které je při volání procedury (funkce) vyhrazeno paměťové místo do něhož je zkopírována hodnota skutečného parametru

Volání hodnotou (2)

  - po ukončení procedury (funkce) se toto místo zase uvolní
- Podprogram může hodnotu této své lokální proměnné měnit, aniž by se tím měnila hodnota skutečného parametru
- Parametr-hodnota tedy nemůže v žádném případě reprezentovat výsledek výpočtu

Volání odkazem (1)

- Skutečným parametrem musí být proměnná
- Odpovídajícímu formálnímu parametru musí v hlavičce procedury předcházet klíčové slovo var
- Substituce skutečného parametru za parametr formální probíhá takto:
  - jsou vyhodnoceny případné indexy skutečného parametru

Volání odkazem (2)

  - formální parametr (jeho identifikátor) se ztotožní s adresou paměťového místa v němž je uložena hodnota odpovídajícího skutečného parametru
- Vyvoláním podprogramu se tak akce popsaná jeho deklarací realizuje přímo nad skutečným parametrem
- Používá se u všech výstupních parametrů

Volání odkazem (3)

- Volání odkazem se používá i u vstupních parametrů, a to v případě, kdy vstupním parametrem je proměnná strukturovaného typu (např. pole)
- Kopírování strukturovaných hodnot skutečných parametrů do paměťových míst vyhrazených lokálním proměnným při použití volání hodnotou má za následek časové ztráty a zvýšení paměťových nároků

Vnořené procedury a funkce

- Z popsané definice bloku vyplývá, že procedury a funkce mohou být vzájemně vnořeny
- Proto se také uvádí, že programovací jazyk Pascal má tzv. blokovou strukturu (narozdíl např. od jazyka C, který má modulární strukturu)

Globální, lokální a nelokální objekty (1)

- Objekty (návěští, konstanty, datové typy, proměnné, procedury a funkce) deklarované a definované uvnitř podprogramu se nazývají lokální
- Lokální objekty jsou přístupné pouze z daného podprogramu a z podprogramů do něj vnořených. Z ostatních částí programu jsou nepřístupné.

Globální, lokální a nelokální objekty (2)

- V deklaraci podprogramu resp. v příkazové části se mohou vyskytovat i objekty, které nejsou lokální (nejsou deklarovány nebo definovány v bloku daného podprogramu)
- Takovýmto objektům říkáme:
  - nelokální: jsou deklarovány (definovány) v některé z nadřazených procedur nebo funkcí
  - globální: jsou deklarovány (definovány) v hlavním programu

Procedury bez parametru (1)

- Výše uvedených objektů (zejména nelokálních a globálních proměnných) je možné využít pro komunikaci podprogramu se svým okolím namísto komunikace pomocí formálních a skutečných parametrů
- Jsou-li uvnitř podprogramu nelokální (globální) proměnné měněny nežádoucím způsobem, může dojít k tzv. vedlejšímu efektu (side effect)

Procedury bez parametru (2)

- Proto se pro zajištění vazby procedur a funkcí na své okolí preferuje užití formálních a skutečných parametrů
- Podprogramy bez parametru se užívají hlavně v případech, kdy se nějaký složený příkaz beze změny opakuje

Rozšířená syntax (1)

- Borland (Turbo) Pascal dovoluje použití tzv. rozšířené syntaxe, která se zapíná direktivou {$X+}
- V rámci rozšířené syntaxe je možné volat funkce stejně jako procedury
- Např.:

  function ReadKey:char;

  může být volána:

Rozšířená syntax (2)

```
{$X+}
var c:char;
begin

    c := ReadKey;   (* klasické volání fce *)

    ReadKey;      (* volání s rozšířenou syntaxí *)

end.
```