



MINISTERSTVO ŠKOLSTVÍ,
MLÁDEŽE A TĚLOVÝCHOVY



**OP Vzdělávání
pro konkurenceschopnost**

INVESTICE
DO ROZVOJE
VZDĚLÁVÁNÍ

Střední průmyslová škola a Vyšší odborná škola technická Brno, Sokolská 1

Šablona: Inovace a zkvalitnění výuky prostřednictvím ICT

Název: Základy programování a algoritmizace úloh

Téma: Typové a netypové soubory

Autor: Ing. Hodál Jaroslav, Ph.D.

Číslo: VY_32_INOVACE_26-09

Anotace: *Materiál popisuje jak lze v jazyce Pascal obsluhovat a zpracovávat typové a netypové soubory. Materiál je určen pro 3. a 4. ročník oboru strojírenství a technické lyceum. Vytvořeno: únor 2013.*

9. Typové a netypové soubory

- **typové** (s udaným typem) i **netypové** (bez udaného typu) soubory jsou obvykle používány pro ukládání dat, která nemají povahu čistého textu
- stejně jako u textových souborů i zde je nutné dodržet posloupnost **3 kroků** práce – propojení, otevření, uzavření
- narozdíl od textových souborů (viz 26-08) však data v obou typech souborů nejsou organizována v řádcích, ukládána jsou jako **lineární posloupnost jednotlivých hodnot**
- každá z těchto hodnot má svým umístěním v souboru jednoznačně určeno **číselné označení** počínaje 0 pro první hodnotu až po N-1 u poslední (je-li počet N)

Základy práce s netextovými soubory

- základní princip práce (3 kroky viz 26-07 – Práce se soubory) i použití příkazů pro propojení (**assign**) a uzavření (**close**) jsou stejné jako u textových souborů
- pro otevření můžeme použít dva příkazy (**reset** a **rewrite**)
- příkaz **append** (nám známý z textových souborů) zde nelze použít
- typové i netypové soubory nerozlišují žádné režimy otevírání souboru (u textových pouze pro zápis x pouze pro čtení)
- **reset** i **rewrite** tedy otevřou soubory pro zápis i čtení
- **reset** zachová původní data, **rewrite** soubor vyprázdní

Ukazatel aktuální pozice

- ukazatel aktuální pozice v souboru je po otevření souboru nastaven **vždy na začátek** souboru
- čtení i zápis dat vždy probíhá na pozici tohoto ukazatele
- soubor je zpracováván sekvenčně podobně jako textové soubory, což znamená, že po každém přečtení/zápisu dat se ukazatel pozice **automaticky posune** na další záznam (položku) v souboru (zvýší se tedy o 1)
- na rozdíl od textových souborů lze pozici (číslo záznamu), na níž se ukazatel nachází, zjistit příkazem **filepos (f)**; a počet záznamů v souboru příkazem **filesize (f)**;
- ukazatel pozice lze navíc i bez čtení či psaní posunout na libovolnou pozici příkazem **seek (f, pozice)**;

Čtení a zápis dat

- **čtení dat** se provádí výhradně příkazem **read** (v typových i netykových souborech nejsou žádné řádky, proto nelze použít `readln`)
- **read(f, x)** přečte jednu položku z místa, kde se nachází ukazatel aktuální pozice v souboru a uloží její hodnotu do proměnné `x`
- **zápis dat** se provádí výhradně příkazem **write** (`writeln` opět použít nelze)
- **write** zapíše jednu položku do místa, kde se nachází ukazatel aktuální pozice v souboru (přepíše stávající položku, pokud nejsme na konci souboru, přidá novou, pokud jsme na konci souboru)

Typové soubory

- **typový soubor** (soubor s udaným typem) je takový druh souboru, do kterého ukládáme **jediný druh informací** (jednoduchý nebo strukturovaný kromě textového řetězce)
- informace jsou do něj zapisovány v **binární podobě** (při otevření souboru např. v textovém prohlížeči nelze vidět zapsaná data ve snadno čitelné podobě jako u textového souboru)
- zapisovat do něj lze pouze obsah proměnné nebo výsledek výrazu **identického typu** jako jsou údaje v souboru
- typ informací, které v souboru půjde uchovávat se uvádí v deklaraci proměnné soubor za názvem dat. typu **file** a slůvkem **of**

Příklad práce s typovým souborem

```
var f : file of byte; x, i : byte;
begin
  assign(f, 'C:\data.dat');
  rewrite(f);

  write(f, 3, 5, 6);

  seek(f, 0);
  while not eof(f) do begin
    read(f, x); writeln(x);
  end;

  seek(f, 1); write(f, 9);

  seek(f, 0);
  for i := 1 to 3 do begin
    read(f, x); writeln(x);
  end;

  close(f);
end.
```

zkuste okomentovat, co dělají jednotlivé příkazové bloky

Příklad práce s typovým souborem

```
var f : file of byte; x, i : byte;
begin
  assign(f, 'C:\data.dat');
  rewrite(f);

  write(f, 3, 5, 6);

  seek(f, 0);
  while not eof(f) do begin
    read(f, x); writeln(x);
  end;

  seek(f, 1); write(f, 9);

  seek(f, 0);
  for i := 1 to 3 do begin
    read(f, x); writeln(x);
  end;

  close(f);
end.
```

zkuste okomentovat, co dělají jednotlivé příkazové bloky

**propojení souboru
otevření souboru**

zápis 3 hodnot do souboru

přesun na začátek souboru

načtení a výpis všech dat ze souboru (1. metoda)

**přesun na druhou hodnotu a její přepsání na novou (9)
přesun na začátek souboru**

načtení a výpis všech dat ze souboru (2. metoda)

uzavření souboru

Netypové soubory

- **netypový soubor** (soubor bez udaného typu) je takový druh souboru, do kterého ukládáme **libovolné informace** po blocích jednotné velikosti
- informace jsou do něj zapisovány v **binární podobě**
- v deklaraci proměnné se uvádí pouze typ **file** bez dalších doplňků
- při otevírání netypového souboru příkazy **reset** nebo **rewrite** lze uvést i velikost zapisovaného/čteného bloku v bytech – např. **reset (f, 50)** ; tuto velikost dat pak při zápisu i čtení musíme dodržet
- pokud velikost bloku v příkazu neuvedeme, použije se výchozí hodnota **128**