



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: Textové soubory

Autor: Ing. Hodál Jaroslav, Ph.D.

Číslo: VY_32_INOVACE_26-08

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

8. Textové soubory

- z pohledu jazyka Pascal je textovým souborem jakýkoliv soubor, který obsahuje **smysluplný text bez dodatečného netextového obsahu**
- textovým souborem tedy může být čistý textový soubor (obvykle s příponou txt v názvu), ale také třeba zdrojový kód Pascalu (soubor *.pas), webová stránka (*.html), ...
- textovým souborem na druhou stranu rozhodně není dokument MS Word (*.doc), protože obsahuje netextový obsah popisující formátování dokumentu, nebo třeba PDF dokument (může obsahovat např. binární obrazová data)

Základy práce se textovými soubory

- základní princip práce s textovými soubory je stejný jako u dalších typů souborů
- vždy vyžaduje 3 kroky – **propojení, otevření a uzavření** prováděné přesně v tomto pořadí
- proměnná, která zastupuje při všech operacích textový soubor bude typu **text** (jedná se o název datového typu, pozor, neplést s datovým typem textový řetězec – `string`)
- otevření textového souboru je možné provést třemi různými příkazy – **reset, rewrite a append**
- na volbě příkazu pro otevření bude zásadně záviset, co všechno budeme moci s textovým souborem dělat

Režimy práce s textovým souborem

- textová data mají narozdíl od ostatních druhů dat poněkud odlišné vlastnosti, a proto u textových souborů rozlišujeme **dva režimy práce**
- volbu režimu, ve kterém budeme se souborem pracovat, provádíme nepřímo tím, jaký příkaz pro otevření souboru použijeme – soubor tedy můžeme otevřít
 - **pouze pro čtení**
 - **pouze pro zápis**
- z uvedených informací vyplývá, že textový soubor **nelze** otevřít tak, aby z něj bylo možné číst a zároveň do něj zapisovat

Otevření pro čtení

- otevření textového souboru pro čtení je možné provést pouze příkazem **reset**
- takto otevíraný soubor musí na disku počítače existovat
- je otevřen v režimu pouze pro čtení – z takto otevřeného souboru je tedy možné pouze číst data, nelze je do něj zapisovat
- pozice pro čtení (ukazatel, kde zrovna v souboru čteme) je po otevření nastavena na začátek souboru
- příkaz **reset** neslouží ke čtení dat, pouze zpřístupňuje soubor, číst data je možné příkazy **readln** nebo **read**
- u obou příkazů je nutné jako první parametr uvést, z jakého souboru se mají data číst **readln (f , x) ; read (f , x) ;**

Čtení dat z textového souboru (1)

- protože příkazy `read/readln` umožňují načítat **3 různé druhy informací** (text, znak, číslo) dostáváme **6** různých možných průběhů procesu čtení dat z textového souboru
- vždy se čte informace od ukazatele pozice v souboru
- významnou roli při čtení dat hrají konce řádků (jsou tvořeny dvěma znaky `#10` a `#13`), které nejsou běžně zobrazovány
- největší blok dat, který je možné načíst naráz je jeden řádek
- **`x:string + readln(f,x)`**
 - načte se vše až po konec řádku
 - data se uloží do proměnné `x`
 - pozice pro čtení se posune na začátek nového řádku

Čtení dat z textového souboru (2)

- **`x:char + readln(f,x)`**
 - načte se jeden znak
 - data se uloží do proměnné `x`
 - pozice pro čtení se posune na začátek nového řádku
- **`x:integer/real + readln(f,x)`**
 - načte se vše až po konec řádku
 - je-li to možné, data se převedou na číslo a uloží do proměnné `x`
 - pozice pro čtení se posune na začátek nového řádku

Čtení dat z textového souboru (3)

- **`x:string + read(f,x)`**
 - načte se vše až po konec řádku a uloží do proměnné `x`
- **`x:char + read(f,x)`**
 - načte se jeden znak a uloží se do proměnné `x`
 - pozice pro čtení se posune na začátek nového řádku
- **`x:integer/real + read(f,x)`**
 - přeskočí se všechny bílé znaky (mezery, tabulátory, konce řádků) po první běžný znak a načte se vše až po první další bílý znak
 - pokud to lze, převedou se data na číslo a uloží do `x`

Zápis dat do textového souboru

- pomocí příkazů **write/writeLn** lze do souboru zapisovat stejné druhy informací jako na obrazovku počítače (text, znak, číslo, logická hodnota) a výsledný výstup je obrazovce velmi podobný (viz 25-7 – Jednoduché příkazy jazyka Pascal)
- stejně jako u obrazovky **write** pouze vypíše data, **writeln** vypíše a odřádkuje
- na rozdíl od obrazovky nejsou texty automaticky zalamovány po 80 znacích (na koncích řádků obrazovky), ale mohou tvořit delší řádky