

Textové súbory

Na ukladanie údajov mimo operačnej pamäte sú určené súbory. Dokážeme ich vytvárať a manipulovať s uloženými údajmi aj prostredníctvom programu. Do textového súboru môžeme zapisovať a čítať z neho znaky (`char`), ale aj viac znakov (`string`) a čísla (`integer`) podobne ako pri čítaní z klávesnice a zapisovaní na obrazovku.

Súbor = postupnosť (sekvencia) prvkov rovnakého typu – znaky usporiadané v riadkoch rôznej dĺžky. **Všetky údaje sú v textovom súbore zapísané ako ASCII znaky.**

textový súbor = postupnosť riadkov (aj prázdna)

riadok textového súboru = postupnosť znakov (aj prázdna) ukončená koncom riadka

Na konci každého riadka je v súbore neviditeľný **znak konca riadka**. Textový editor ho nezobrazuje, ale v súbore je zapísaný ako dva znaky (**#13#10**). Na konci celého súboru je tiež neviditeľný **znak konca súboru** (**#26**).

- s textovými súbormi sa dá pracovať len sekvenčným prístupom
- musí sa rozlišovať, či zo súboru čítame alebo do neho zapisujeme údaje:
 - vstupný súbor - len čítať pripravené údaje
 - výstupný súbor - len zapisovať, na koniec súboru!
 - **do textového súboru sa nedá aj zapisovať aj súčasne z neho čítať**

Súboru (`file`) musí byť priradené nejaké fyzické zariadenie, resp. musí sa nachádzať na nejakom zariadení - diskový súbor.

"Operácie" so súbormi:

- **write** = zápis informácie na koniec súboru (podobne ako do **Memo1.Lines.Add(...)**)
- **read** = čítanie nasledujúcej hodnoty zo súboru

Ako pracujeme s textovým súborom

Deklarovanie premennej typu textový súbor:

```
var t:TextFile; // premenná t je typu textový súbor
```

Priradenie fyzického súboru najčastejšie na disku:

```
AssignFile(t, 'meno_suboru.txt');
```

Dávajte si pozor na uvádzanie plnej cesty na disku k nejakému súboru - ak sa takýto program preniesie na iný počítač, s veľkou pravdepodobnosťou tieto absolútne cesty k súborom nebudú fungovať - používajte radšej relatívne cesty od momentálnej adresy projektu (tam, kde sa nachádza EXE súbor)

Otvorenie súboru

- buď na čítanie (musí už existovať): `Reset(t);`
- alebo zápis (ak už existuje, tak sa najprv vyprázdni): `Rewrite(t);`
Súbor sa otvorí. Ak už existoval jeho obsah sa zruší a čaká sa na zapisovanie nových hodnôt. Ak ešte neexistoval, vytvorí sa s prázdny obsahom.

Práca so súborom, t.j. samotné zapisovanie alebo čítanie:

```
write(t, hodnota); writeln(t);        =        writeln(t, hodnota);
```

Formátovací parameter vo write

- za **znakom** alebo **znakovým reťazcom** `write(t, '*':10);`
 - označuje, že znak sa vypíše na šírku 10, t.j. najprv 9 medzier a potom '*'
 - keď je reťazec dlhší ako formátovací parameter, zapíše sa kompletný reťazec, t.j. ignoruje sa formát
- za **celým číslom** označuje šírku, do ktorej sa má zapísať číslo, ak by nevošlo do danej šírky, formát sa ignoruje
`write(t, 25*25 :5);` zapíše dve medzery, za ktoré dá číslo 625
- za **reálnym číslom** označuje šírku, do ktorej sa má číslo zapísať, desatinné číslo sa vypíše v *semilogaritmickej* tvare; druhý formátovací parameter označuje počet desatinných miest
`write(t, 100/3 :7 :4);` zapíše 33.3333

`read(t, premenna);` prečíta sa jeden znak zo súboru z pozície ukazovateľa, priradí sa jeho hodnota do *premennej* a **ukazovateľ sa posunie za znak**

vidíme:	v skutočnosti:
ab	a b <koniec riadka>
cde	c d e <koniec riadka> f <koniec súboru>
f	^ ukazovateľ

`readln(t,premenna);` prečíta sa jeden znak zo súboru z pozície ukazovateľa, priradí sa jeho hodnota do *premennej* a **ukazovateľ** sa posunie na nový riadok

Testovanie konca súboru a konca riadka

- štandardná logická funkcia **End Of File** vráti true, ak je ukazovateľ nastavený za posledným znakom súboru `Eof(t)`
- štandardná logická funkcia **End Of LiNe** vráti true, ak je ukazovateľ nastavený na konci riadka alebo aj za posledným znakom súboru `Eoln(t)`

```
while not Eof(t) do begin
  while not Eoln(t) do begin
    read(t,a);
  end;
  readln(t);
end;
```

Zatvorenie súboru, t.j. ukončenie práce so súborom:

```
CloseFile(t);
```

Čítanie čísel

Vieme čítať aj čísla (celé aj reálne), ale v súbore musia byť tieto čísla oddelené medzerou, koncom riadka alebo tabulátorom (znak s kódom #9). Príkaz `read` preskočí všetky medzerové znaky (medzera, <Eoln> alebo #9), potom prekonvertuje znaky zo vstupu na číslo. Ak je číslo ukončené nemedzerovým znakom (napr. '.' alebo ';'), tak vyhlási chybu **Invalid numeric format**. Ak čítame koniec súboru (t.j. platí `eof`, ale aj ak sú tam len medzery) vráti hodnotu 0 - treba sa tohoto vyvarovať!

Preto takéto čítanie `read` môžeme používať, len ak je výslovne povedané, že je súbor korektný a číslo je ukončené medzerovým znakom.

Preto je lepšie používať štandardnú funkciu **SeekEof**, ktorá skôr ako odpovie na stav súboru, odfiltruje všetky medzerové znaky. Podobne existuje **SeekEoln**, ktorá testuje koniec riadka, ale najprv odfiltruje medzery a tabulátory.

POZOR: funkcie `SeekEoln` a `SeekEof` **odignorujú** všetky **medzery**, tabulátory a v prípade `SeekEof` i konce riadkov. Preto tieto funkcie **nie sú vhodné pri čítaní znakov**, používame ich hlavne pri čítaní čísel z textového súboru.

1. Daný je súbor celých čísel `cisla.txt` oddelených navzájom medzerami a koncami riadkov. Zistite súčet všetkých čísel a vypíšte ho do Memo.

<pre>98 7 48 18 165 37 36 55 138 484 335 8 5 7 75 20 46 56 123 737 75 6 8 24 5</pre>	súčet čísel v súbore je: 2616
--	--------------------------------------

2. Daný je súbor celých čísel `cisla.txt` oddelených navzájom medzerami a koncami riadkov. Zistite, koľko čísel je z intervalu $(10,30)$ a túto informáciu vypíšte do Memo. Program upravte tak, aby sme si hranice intervalu mohli zadať v Editoch.

1	
30	
<pre>98 7 48 18 165 37 36 55 138 484 335 8 5 7 75 20 46 56 123 737 75 6 8 24 5</pre>	počet čísel z intervalu <1,30) je 10

3. Daný je súbor celých čísel `cisla.txt` oddelených navzájom medzerami a koncami riadkov. Skopírujte z neho každé tretie číslo do nového súboru `kazde3.txt`.
 - a. Vytvorte súbor, v ktorom budú vybrané čísla zapísané v jednom riadku oddelené medzerou.
 - b. Vytvorte súbor, v ktorom budú vybrané čísla zapísané vždy v samostatnom riadku.

Obsah súboru `kazde3.txt` vypíšte do Memo. Použite príkaz
`Memo1.Lines. LoadFromFile('nazov_suboru.txt');`

<pre>98 7 48 18 165 37 36 55 138 484 335 8 5 7 75 20 46 56 123 737 75 6 8 24 5</pre>	<pre>48 37 138 8 75 56 75 24</pre>	<pre>98 7 48 18 165 37 36 55 138 484 335 8 5 7 75 20 46 56 123 737 75 6 8 24 5</pre>	<pre>48 37 138 8 75 56 75 24</pre>
--	------------------------------------	--	------------------------------------

4. Napíšte program, ktorý zapíše do textového súboru `ciselka.txt` náhodné čísla z intervalu $(100,300)$ tak, aby v súbore bolo aspoň 5 a najviac 10 riadkov a v každom riadku aspoň 3 a najviac 7 čísel. Čísla oddelujte medzerou. Obsah súboru `ciselka.txt` vypíšte do Memo.

ciselka.txt - Poznámkový blok Súbor Úpravy Formát Zobrazit' Pomocník <pre>243 268 220 271 208 184 224 229 176 187 159 111 292 154 176 195 258 205 195 213 178 285 267 167 117 229 173 266 291 128 274 274 295 194 259 192 204 256 235 123 244</pre>	ciselka.txt - Poznámkový blok Súbor Úpravy Formát Zobrazit' P <pre>128 207 288 251 204 182 194 152 254 247 191 213 127 103 223 129 222 144 177 288 280 236 189 222 187 280 239 112 293 233</pre>
---	--

5. Napíšte program, ktorý zistí a vypíše do Memo, koľko je v súbore `ciselka.txt` (z predchádzajúceho príkladu) párnych čísel väčších ako 200, tieto čísla tiež vypíše.

128 207 288 251 204	288
182 194 152	204
254 247 191	254
213 127 103	222
223 129 222 144	288
177 288 280 236 189	280
222 187 280 239	236
112 293 233	222
	280
	počet párnych čísel väčších ako 200
	je v súbore: 9

6. V textovom súbore `cisla1.txt` sú jedno, dvoj a trojciferné čísla. Napíšte program, ktorý tieto čísla naformátuje do 5 stĺpcov do súboru `naformatovane.txt`. Medzi jednotlivými stĺpcami je jedna medzera.

naformatovane.txt - Poznámkový				
Súbor	Úpravy	Formát	Zobraziť	Pc
898	776	548	180	765
837	386	550	138	484
335	857	367	589	75
820	426	656	123	737
75	623	508	274	5
458	489	495	156	755
176	425	739	723	179
697	653	249	618	85
315	965	132	950	119
477	728	925	8	190
578	359	911	393	137
334	3	530	184	297
47	824	639	599	288
687	278	494	630	569
694	394	102	189	125
427	822	306	990	586

7. Daný je súbor celých čísel (`cisla2.txt`) oddelených navzájom medzerami a koncami riadkov. Nájdite najmenšie číslo a vypíšte ho do Memo.

898 776 548 180 765 837 386 550 138	najmenšie číslo je: 3
484 335 857 367 589 75 820 426 656	
123 737 75 623 508 274 5 458 489	
495 156	
755 176 425 739 723 179 697 653 249	
618 85 315 965 132 950 119 477 728	
925 8 190 578 359 911 393 137 334 3	
530	
184 297 47 824 639 599 288 687 278	
494 630 569 694 394 102 189 125 427	
822 306 990 586 953 455 268 35 398	
322	
59 346 190 679 937 736 846 49 972	
580 140 525 509 764 44 413 79 667	
818 626 42 814 486 994 825 512 179	
911 867	
999 80 5 620 914 955 730 237 128	
507 75 775 240 35 797 52 281 135	
852 501 418 422 952 632 715 510 413	

8. Daný je súbor celých čísel (cisla2.txt) oddelených navzájom medzerami a koncami riadkov. Nájdite druhé najväčšie číslo a vypíšte ho do Memo.

898 776 548 180 765 837 386 550 138 484 335 857 367 589 75 820 426 656 123 737 75 623 508 274 5 458 489 495 156 755 176 425 739 723 179 697 653 249 618 85 315 965 132 950 119 477 728 925 8 190 578 359 911 393 137 334 3 ---	druhé najväčšie číslo je: 994
---	-------------------------------

9. Daný je súbor celých čísel (cisla2.txt) oddelených navzájom medzerami a koncami riadkov. Zistite priemer zo všetkých čísel a vypíšte ho do Memo.

898 776 548 180 765 837 386 550 138 484 335 857 367 589 75 820 426 656 123 737 75 623 508 274 5 458 489 495 156 755 176 425 739 723 179 697 653 249 618 85 315 965 132 950 119 477 728 925 8 190 578 359 911 393 137 334 3 530 184 297 47 824 639 599 288 687 278 494 630 569 694 394 102 189 125 427 822 306 990 586 953 455 268 35 398 322 59 346 190 679 937 736 846 49 972 580 140 525 509 764 44 413 79 667 818 626 42 814 486 994 825 512 179 911 867 -----	priemer čísel je: 489,18
---	--------------------------

10. Daný je súbor celých čísel (cisla2.txt) oddelených navzájom medzerami a koncami riadkov. Zistite koľko čísel je podpriemerných a vypíšte informáciu do Memo.

898 776 548 180 765 837 386 550 138 484 335 857 367 589 75 820 426 656 123 737 75 623 508 274 5 458 489 495 156 755 176 425 739 723 179 697 653 249 618 85 315 965 132 950 119 477 728 925 8 190 578 359 911 393 137 334 3 530 184 297 47 824 639 599 288 687 278 494 630 569 694 394 102 189 125 427 822 306 990 586 953 455 268 35 398 322	priemer čísel je: 489,18 podpriemerných čísel je: 98
---	---