

Pracovní list 9: Textové soubory

Co už máme znát

- operace pro vstup a výstup standardních souborů;
- formátování textového výstupu;
- knihovnu a operace pro otevření a zpracování uživatelského souboru;
- práci s textovými daty (řádky) souboru;
- práci s polem záznamů.

Kontrolní otázky

- 9.1 Jakým způsobem se čte ze standardního vstupu?
- 9.2 Jakým způsobem se vypisuje na standardní výstup a na standardní chybový výstup?
- 9.3 K čemu slouží operace `setw()`, `right`, `setfill()`, `setprecision()` a v jaké knihovně se nacházejí?
- 9.4 K čemu dochází při výměně dat mezi standardními soubory a operační pamětí?

Příprava na cvičení

Ve cvičení budeme potřebovat editor a překladač pro programy v jazyce C++. V některých příkladech budeme zpracovávat soubor `zamestnanci.txt`, který se nachází na adrese

<https://akela.mendelu.cz/~xturcin0/alg0/zamestnanci.txt>

Stáhněte si jej do pracovního adresáře. Dále si připravte textový soubor `dopis.txt` obsahující alespoň 10 vět. V některých příkladech se nebudeme ohlížet na specifika textových dat s národními znaky (například řazení podle české abecedy).

Řešené příklady

Příklad 9.1 Vytvořte kopii souboru `dopis.txt`. Kopie souboru se bude jmenovat `kopie.txt`.

Řešení: Pro práci se soubory je potřeba připojit knihovnu `<fstream>`. Pro čtení je vytvořena proměnná `vstup` třídy `<ifstream>`. To, že se jedná o objekt, není v tento okamžik vůbec podstatné. Obdobně je pro zápis vytvořena proměnná `vystup` třídy `<ofstream>`. Již při deklaraci proměnných se uvádí fyzický název souboru, se kterým budete pracovat prostřednictvím této proměnné. Kromě toho lze také soubory otevřít metodou `open`, do jejíhož parametru napíšete fyzické jméno souboru. Skutečnost, že se soubor správně otevřel, lze zjistit pomocí metody `is_open`. Tato metoda vrací hodnotu pravda v případě, že otevření proběhne v pořádku. Aby bylo možné otevřít soubor pro

čtení, musí takový soubor existovat. Pokud neexistuje soubor, který otvíráte pro zápis, je vytvořen automaticky. Pokud takový soubor existuje, je přepsán. Tento program načítá ze vstupního souboru jednotlivé řádky do proměnné `radek` a následně je zapisuje do výstupního souboru. Toto se děje tak dlouho, dokud na vstupu nějaké řádky jsou.

```
731 #include <iostream>
732 #include <fstream>
733 using namespace std;
734
735 int main () {
736     string radek;
737     ifstream vstup ("dopis.txt");
738     ofstream vystup ("kopie.txt");
739     if (vstup.is_open() and vystup.is_open()){
740         while (getline (vstup, radek))
741             vystup<<radek<<endl;
742         vstup.close();
743         vystup.close();
744     }
745     else cerr << "Nepodařilo se otevřít některý ze souborů.";
746     return 0;
747 }
```

Příklad 9.2 Z textového souboru `zamestnanci.txt` vypíšte jména a příjmení všech žen s platem nad 30000.

Řešení: Nejprve je potřeba si soubor stáhnout do stejného adresáře, jako máte svůj program. Opět zde bude využita knihovna `<fstream>`. Pro načítání jednotlivých položek jsou vytvořeny proměnné `jmeno`, `prijmeni`, `poh` (pro pohlaví), `rok` a `plat`. Pokud se v pořádku otevře vstupní soubor, jsou do proměnných postupně načítány údaje. Vše probíhá tak dlouho, dokud se nedojde na konec souboru. To určuje řídicí podmínka cyklu. V každém průběhu cyklu se z jednoho řádku načtou všechny údaje. Soubor musí být správně vytvořený, jinak může dojít k chybě při načítání. Po načtení dat se ověří, zda se jedná o ženu s platem vyšším než 30 000. Pokud ano, vypíše se její jméno a příjmení.

```
748 #include <iostream>
749 #include <fstream>
750 using namespace std;
751
752 int main () {
753     string jmeno;
754     string prijmeni;
755     char poh;
756     int rok, plat;
757     ifstream vstup ("zamestnanci.txt");
758     if (vstup.is_open()){
759         while (!vstup.eof()){
```

```
760     vstup>>jmeno>>prijmeni>>rok>>poh>>plat;  
761     if(poh=='z' and plat>30000)  
762         cout<<jmeno<<" "<<prijmeni<<endl;  
763     }  
764     vstup.close();  
765 }  
766 else cerr << "Nepodařilo se otevřít soubor."  
767 return 0;  
768 }
```

Příklad 9.3 Z textového souboru zamestnanci.txt vypište všechny zaměstnance podle výše platu od největšího po nejmenší.

Řešení: V tomto příkladu se použijí prvky z pracovního listu Řazení a záznam. Každý zaměstnanec je uložen do záznamu v poli. Pro zpřehlednění je řazení v samostatném podprogramu. Stejně tak i výpis, který je převzat z uvedeného pracovního listu. Jak vidíte, jde pouze o aplikaci již známých postupů.

```
769 #include <iostream>  
770 #include <fstream>  
771 using namespace std;  
772 typedef struct {  
773     string jmeno;  
774     string prijmeni;  
775     char pohlavi;  
776     int rok;  
777     int plat;  
778 } TypZamestnanec;  
779 typedef TypZamestnanec TypPoleZamestnancu[20];  
780  
781  
782 void vypisZamestnance(TypZamestnanec clovek){  
783     cout<< "jmeno: "<<clovek.jmeno<<endl;  
784     cout<< "prijmeni: "<<clovek.prijmeni<<endl;  
785     cout<< "rok narozeni: "<<clovek.rok<<endl;  
786     cout<< "pohlavi (m/z): "<<clovek.pohlavi<<endl;  
787     cout<< "plat: "<<clovek.plat<<endl;  
788 }  
789  
790 void vypisSeznam(TypPoleZamestnancu seznam, int pocet){  
791     for (int i=0;i<pocet;i++){  
792         vypisZamestnance(seznam[i]);  
793         cout<<"-----"<<endl;  
794     }  
795 }
```

```
796
797 void seradPodlePlatu(TypPoleZamestnancu pole, int pocet) {
798     int j = 0;
799     for (int i = 0; i < pocet - 1; i++) {
800         j = i + 1;
801         TypZamestnanec pom = pole[j];
802         while (j > 0 and pom.plat > pole[j-1].plat) {
803             pole[j] = pole[j-1];
804             j--;
805         }
806         pole[j] = pom;
807     }
808 }
809
810 int main () {
811     TypPoleZamestnancu zam;
812     int pocet=0;
813     ifstream vstup ("zamestnanci.txt");
814     if (vstup.is_open()){
815         while (!vstup.eof()){
816             vstup>>zam[pocet].jmeno>>zam[pocet].prijmeni
817                 >>zam[pocet].rok>>zam[pocet].pohlavi
818                 >>zam[pocet].plat;
819             pocet++;
820         }
821         vstup.close();
822         seradPodlePlatu(zam,pocet);
823         vypisSeznam(zam,pocet);
824     }
825     else cerr << "Nepodařilo se otevřít soubor.";
826     return 0;
827 }
```

Příklady

Příklad 9.4 V přichystaném textovém souboru dopis.txt spočítejte poměr velkých a malých písmen.

Příklad 9.5 V přichystaném textovém souboru dopis.txt spočítejte počet vět a počet slov.

Příklad 9.6 V přichystaném textovém souboru dopis.txt spočítejte absolutní a relativné četnost písmen malé anglické abecedy.

Příklad 9.7 Na základě textového souboru `zamestnanci.txt` vytvořte dva nové soubory. Jeden bude obsahovat pouze ženy a druhý pouze muže.

Příklad 9.8 Z textového souboru `zamestnanci.txt` vypište jména a příjmení všech mužů narozených před rokem 1985 a vypočítejte jejich průměrný plat.

Příklad 9.9 Z textového souboru `zamestnanci.txt` vypište příjmení a jména mužů seřazených podle roku narození od nejstaršího.

Příklad 9.10 Předchozí dva příklady modifikujte tak, aby programy zpracovávaly standardní vstup. V čem se budou od předchozí varianty lišit?

Co máme po cvičení umět

- použít knihovnu s operacemi pro práci s uživatelskými soubory;
- použít operace pro otevření uživatelského souboru pro čtení/zápis;
- použít operace zjišťující stav souboru (otevřený, není na konci);
- použít operaci pro uzavření uživatelského souboru.

Kontrolní otázky

- 9.5 Jak se liší čtení standardního vstupního souboru od čtení uživatelského textového souboru?
- 9.6 Jaké jsou výhody standardních souborů oproti souborům se zadaným názvem?
- 9.7 Co se stane, neuzavřeme-li po zpracování uživatelský soubor?
- 9.8 Co se děje při výměně dat mezi uživatelským textovým souborem a proměnnými v operační paměti?