

Úvod do programování

Lekce 4

Práce se soubory - pokračování

příklad: textový soubor `random.txt` obsahuje pět řádků po 10000 celých číslech; program vypočítá a zapíše do souboru `statistics.txt` střední hodnotu a varianci pro každý řádek

```
#include<iostream>
#include<fstream>
#include<iomanip>
using namespace std;
#define RADKU 5          // definuje format datoveho souboru
#define PO CET 10000
int main(void){
    ifstream fr; ofstream fw;
    double soucet,soucet2,n,variance,stredni;
    int radek,i;
    fr.open("random.txt"); fw.open("statistics.txt");
    fw << "radek      stredni hodnota      variance\n";
    fw << "=====\n";
    for (radek = 1; radek <= RADKU; radek++){
        cout << radek << endl; // kontrolni vypis na obrazovku
        soucet = soucet2 = 0; // pro kazdy radek zacnam s nulou
        for (i = 1; i <= PO CET; i++){
            fr >> n;
            soucet += n; // scitam data na radku
            soucet2 += n*n; // scitam druhe mocniny
        }
        stredni = soucet/PO CET; // prumer
        variance = soucet2/PO CET-stredni*stredni; // variance
        fw << fixed << setw(3) << radek;
        fw << setw(17) << stredni << setw(16) << variance << endl;
    }
    fr.close(); fw.close();
}
```

příklad: stejný datový soubor; program zjistí maximum a minimum pro každý řádek

```
#include<iostream>
#include<fstream>
#include<iomanip>
using namespace std;
#define RADKU 5
#define PO CET 10000
int main(void){
    ifstream fr;
    int radek,i,pozice_max,pozice_min,max,min,n;
    fr.open("random.txt");
    cout << "radek      max      pozice_max      min      pozice_min\n";
    cout << "=====\n";
    for (radek = 1; radek <= RADKU; radek++){
        fr >> n;
```

```

max = min = n;          // prvni cislo je pocatecni min i max
pozice_max = pozice_min = 1;
for (i = 2; i <= PO CET; i++){
    fr >> n;
    if(n > max){
        max = n;          // "n" je nove maximum
        pozice_max = i;
    }
    if(n < min){
        min = n;          // "n" je nove minimum
        pozice_min = i;
    }
}
cout << setw(3) << radek;
cout << setw(9) << max << setw(12) << pozice_max;
cout << setw(7) << min << setw (11) << pozice_min << endl;
}
fr.close();
}

```

poznámky:

- pokud se chceme vyhnout nutnosti číst první číslo na řádce zvlášť před vstupem do cyklu, nastavíme proměnnou `max` na velmi malou hodnotu např. `-9999`, o které bezpečně víme, že je menší než nejmenší číslo na řádce; podobně postupujeme u proměnné `min`
- názvy maker (symbolických konstant) se doporučuje psát velkými písmeny pro odlišení od názvů proměnných

neformátovaný vstup a výstup

při práci se soubory postupujeme obdobně jako u standardního vstupu a výstupu

příklad: kopírování souboru

```

#include<iostream>
#include<fstream>
using namespace std;
int main(void){
    ifstream fr;
    ofstream fw;
    int c;
    fr.open("vstup.txt");          // otevřím "vstup.txt" pro čtení
    if(!fr.is_open()){            // kontrola úspěšného otevření
        cout << "chyba otevření souboru! \n";
        exit(1);                  // ukončení programu
    }
    fw.open("vystup.txt");
    while((c = fr.get()) != EOF){  // čtení do konce souboru
        cout.put(c);              // výpis na obrazovku pro kontrolu
        fw.put(c);                // zápis do souboru
    }
}

```

```
    }  
    fr.close(); fw.close();  
}
```

poznámka:

- při čtení za koncem souboru je vrácena konstanta definovaná makrem EOF
- lze využít pro čtení do konce souboru

příklad: program sloučí dva soubory do jednoho

```
#include<iostream>  
#include<fstream>  
using namespace std;  
int main(void){  
    ifstream fr; ofstream fw;  
    int c;  
    fr.open("a.txt"); // oteviram "a.txt" pro cteni  
    fw.open("c.txt"); // oteviram "c.txt" pro zapis  
    while((c = fr.get()) != EOF){ // ctu "a.txt"  
        cout.put(c); // vypis na obrazovku pro kontrolu  
        fw.put(c); // zapis do souboru  
    }  
    fr.close();  
    fr.open("b.txt"); // oteviram "b.txt" pro cteni  
    while((c = fr.get()) != EOF){ // ctu "b.txt"  
        cout.put(c);  
        fw.put(c);  
    }  
    fr.close(); fw.close();  
}
```