

Programowanie komputerowe

Zajęcia 6

Strumienie

Strumienie to obiekty służące do obsługi ciągu danych. Podział:

- ze względu na kierunek przepływu danych:
 - wejściowe
 - wyjściowe
- ze względu na źródło/cel danych
 - terminal (cin, cout)
 - plik (ifstream, ofstream)
 - napis (istringstream, ostringstream)
- ze względu na typ danych – zazwyczaj char

Czytanie plików

Do odczytu plików służą obiekty klasy `ifstream` deklarowane w bibliotece `fstream`.

Tworzenie strumienia:

- `ifstream s; // tworzy strumień`
- `ifstream s("plik.txt"); // tworzy strumień i łączy z plikiem o podanej nazwie`

Jeśli podajemy pełną ścieżkę do pliku musimy użyć podwójnych ukośników, np.

```
ifstream s("C:\\katalog\\plik.txt");
```

Czytanie plików (2)

Możemy połączyć strumień z plikiem już po stworzeniu go przy użyciu funkcji `open` np.

```
s.open("plik.txt");
```

Po użyciu należy odłączyć strumień przy użyciu funkcji `close`:

```
s.close();
```

Przypisanie pliku do strumienia może się nie powieść; żeby sprawdzić czy strumień jest połączony można użyć funkcji `is_open`.

Żeby sprawdzić, czy wczytane zostały wszystkie znaki z pliku używamy funkcji `eof`.

Czytanie plików (3)

Znaki z pliku można wczytywać używając operatora `>>`, tak jak w przypadku strumienia `cin`. Inne sposoby (x jest pewnym strumieniem wejściowym)

- `x.get()` – wczytuje jeden znak i go zwraca,
- `x.get(char& c)` – wczytuje znak i umieszcza na zmiennej `c`
- `x.get(char* s, int n)` – wczytuje znaki aż do końca linii do tablicy `s`, ale nie więcej niż `n-1`
- `x.get(char* s, int n, char d)` – jak wyżej, ale zamiast końca linii szukamy znaku `d`
- `getline(istream& x, string& s)` – wczytuje linię do napisu.

Czytanie plików – przykład

```
int main() {  
    ifstream x("a.txt");  
    if(!x.is_open()) {  
        cout << "Nie udalo sie otworzyc pliku";  
        return -1;  
    }  
    string s;  
    while(!x.eof()) {  
        getline(x, s);  
        cout << s << endl;  
    }  
}
```

Pisanie do plików

Do zapisu do pliku służą obiekty klasy `ofstream` deklarowane w bibliotece `fstream`.

Tworzenie strumienia:

- `ofstream s; // tworzy strumień`
- `ofstream s("plik.txt"); // tworzy strumień i łączy z plikiem o podanej nazwie; jeśli trzeba tworzy ten plik; dane są nadpisywane`
- `ofstream s("plik.txt", ios::app); // tworzy strumień i łączy z plikiem o podanej nazwie; jeśli plik nie jest pusty dopisuje znaki na końcu`

Pisanie do plików (2)

Do pliku piszemy przy pomocy operatora `>>`, tak jak w przypadku `cout`. Inną możliwością jest użycie następujących funkcji:

- `ostream& put(char c)`
wpisuje znak
- `ostream& write(const char* s, size_t n)`
wpisuje nie więcej niż `n` znaków z napisu `s`

Pisanie do plików – przykład

```
int main() {  
    ofstream x("qqq.txt", ios::app);  
    if(!x.is_open()) {  
        cout << "Nie udalo sie otworzyc pliku"; return -1;  
    }  
    string s;  
    do {  
        getline(cin, s);  
        x << s << endl;  
    } while (!s.empty());  
}
```

Formatowanie

Dla strumieni wyjściowych możemy ustalić formatowanie. Np.

- `cout.width(int n);` // ustawia szerokość wydruku
- `cout.fill(char c);` // znak użyty do wypełnienia
- `cout.setf(ios::left, ios::adjustfield);` // ustawia wyrównanie do lewej strony
- `cout.precision(int n);` // ustawia liczbę cyfr znaczących
- i wiele innych

Ćwiczenia

1. Napisać program, który pyta użytkownika o nazwę pliku, a następnie drukuje ten plik na ekranie z pominięciem spacji.
2. Napisać program, który zapisuje w pliku wszystkie liczby pierwsze mniejsze od podanego parametru.
3. Napisać program, który dla podanego pliku nazwa.cpp tworzy kopię `_nazwa.cpp` z usuniętymi komentarzami.
4. Plik <http://www.mimuw.edu.pl/~ziemians/pk2016/12375.txt> zawiera pomiary temperatury na stacji Warszawa Okęcie z pewnego okresu. Napisać program, który drukuje rekordowo niską i rekordowo wysoką temperaturę w tym okresie.