

Programownie I

Wykład 10

dr inż. Adam Zielonka

Instytut Matematyki,
Politechnika Śląska

Gliwice 21.12.2018

Operacje wejścia/wyjścia na plikach

Do programu należy włączyć plik nagłówkowy biblioteki `fstream`

```
#include<fstream>
```

Strumienie

- `ofstream` (*output file steam*) – zapis do plików
- `ifstream` (*input file stream*) – odczyt z plików
- `fstream` (*file stream*) –zapis i odczyt jednocześnie

Uwaga. Nazwy zadeklarowane w pliku `fstream` należą do przestrzeni nazw `std`, zatem użycie `using namespace std`; pozwala na niepisanie kwalifikatora zakresu `std::`

Zapis do pliku

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
    ofstream plik("tekst.txt");
    //tryb tekstowy domyślnie nadpisuje zawartość pliku

    plik << "Zielonka";
    plik.close();

    plik.open("tekst.txt")
    plik << "Adam";
    plik.close();
}
```

Tryby otwarcia

Tryb	Skrót od	Opis
in	input (wejście)	otwórz plik do czytania
out	output (wyjście)	otwórz plik do pisania
ate	at end (na końcu)	otwórz i ustaw się na końcu zawartości
app	append (dołączenie)	otwórz plik i pozwól na dopisanie na końcu pliku
trunc	truncate (odcięcie)	otwórz a jeśli plik istnieje to skasuj jego zawartość
binary		zapis w trybie binarnym

```
fstream plik;
```

```
plik.open("proba2.txt", ios::out | ios::app);
```

```
if(plik.is_open())  
{  
    plik << "Napis";  
    plik.close();  
}  
else  
    cout<<"Nie można otworzyć pliku."<<endl;
```

Odczyt z pliku

```
char z;
ifstream plik("tekst.txt");
if (plik.is_open())
{
    while (!plik.eof())
    {
        plik.get(z);
        cout << z;
    }
}
else
    cout << "Nie mozna otworzyc plik." << endl;
plik.close();
```

Odczyt z pliku

```
fstream plik("tekst.txt", ios::out);  
plik << "Taki sobie\n tekst";  
plik.close();
```

```
string s;  
plik.open("tekst.txt",ios::in);  
if (plik.is_open()){  
    while (!plik.eof()){  
        plik >> s;  
        cout << s<<endl;  
    }  
}  
else  
    cout << "Nie mozna otworzyc plik." << endl;  
plik.close();
```

Odczyt z pliku

```
fstream plik("tekst.txt", ios::out);  
plik << "Taki sobie\n tekst";  
plik.close();
```

```
string s;  
plik.open("tekst.txt", ios::in);  
if (plik.is_open()){  
    while (!plik.eof()){  
        getline(plik, s);  
        cout << s<<endl;  
    }  
}  
else  
    cout << "Nie mozna otworzyc plik." << endl;  
plik.close();
```

Konieczny w przypadku systemów operacyjnych, które inaczej obchodzą się z plikami tekstowymi niż binarnymi np. system operacyjny Windows. Dla strumienia nie ma znaczenie co nim płynie, czy są to dane binarne czy tekstowe, ale ... w przypadku trybu tekstowego (domyślnego) dla systemu Windows:

- każde wystawiony do strumienia znak `'\n'` zostanie zastąpiony na dwa znaki `'\n'` `'\r'`
- w przypadku pobrania ze strumienia znaki `'\n'` `'\r'` zastąpione zostaną na znak `'\n'`

Tryb binarny

```
fstream plik;
plik.open("tekst.txt", ios::out);
plik << "AB\nCD";
plik.close();
plik.open("tekst.txt", ios::in | ios::binary);
char z;
if (plik.is_open())
    while (true){
        plik.get(z); // z=plik.get();
        if (plik.eof()) break;
        cout << (int)z << ' ';
    }
else
    cout << "nie można otworzyć pliku" << endl;
plik.close();
//65 66 13 10 67 68
```

Ustawienie pozycji w pliku i jej określenie

Dla ifstream

```
plik.seekg(2 , ios::beg); //2 znak od początku
plik.seekg(6 , ios::cur); //6 znaków dalej
                        //od bieżącego
plik.seekg(-3 , ios::end); //3 znaki przed początku

plik.tellg(); //aktualna pozycja w pliku
```

Dla ofstream

```
plik.seekp(2 , ios::beg); //2 znak od początku
plik.seekp(6 , ios::cur); //6 znaków dalej
                        //od bieżącego
plik.seekp(-3 , ios::end); //3 znaki przed początku

plik.tellp(); //aktualna pozycja w pliku
```

Określenie ilości znaków w pliku do odczytu

```
ifstream plik("plik.bin" , ios::binary | ios::ate );  
int n = plik.tellg();
```

albo inaczej

```
ifstream plik("plik.bin" , ios::binary );  
...  
plik.seekg(0 , ios::end );  
int n = plik.tellg();
```

Jeszcze zapis i odczyt binarny

Metody read write

```
plik.read( wskaźnik , rozmiar );
```

```
plik.write( wskaźnik , rozmiar);
```

Przykład

```
char z;
```

```
plik.read( &z , sizeof( z ));
```

```
int n=10;
```

```
int tab[n]{};
```

```
plik.read( tab , n*sizeof(int));
```

```
plik2.write( (char*)tab, n*sizeof(int) );
```

Jeszcze zapis i odczyt binarny

Zapis binarny

```
const int  n = 100;
int  tab[n] {}, tab2[n] {};

for (int i = 0; i < 100; i++)
    tab[i] = INT32_MAX-i;

ofstream plik( "liczby.bin" , ios::binary );

if (plik.is_open())
{
    plik.write( (char*)tab , n*sizeof(int) );
    plik.close();
}
else
    cout << "nie mozna ..." << endl;
```

Jeszcze zapis i odczyt binarny

Odczyt binarny

```
ifstream plik2("liczby.bin", ios::binary | ios::ate );
int  n = plik2.tellg();
plik2.seekg(0,ios::beg);

char *tab2 = new char[n];
if (plik2.is_open()){
    plik2.read( tab2 , n );
    plik2.close();
    int n1=n/sizeof(int);
    int *t=(int*) tab2;
    for(int i=0 ; i< n1 ; i++)
        cout << t[i] <<endl;
}
else  cout << "nie mozna ..." << endl;
delete [] tab2;
```