

Programownie I

Wykład 8

dr inż. Adam Zielonka

Instytut Matematyki,
Politechnika Śląska

Gliwice 07.12.2018

- Kompilator – program tłumaczący kod źródłowy programu na język maszynowy
- Preprocesor – część kompilatora. Przetwarza plik źródłowy zgodnie z tym jak poprzez użycie odpowiednich dyrektyw określił programista. Dyrektywy poprzedza znak #. Pliki źródłowe kompilowane są do oddzielnych plików pośrednich.
- Linker(konsolidator) – łączy pliki pośrednie z odpowiednimi bibliotekami finalnie tworząc postać wykonalną programu.

Biblioteka – zbiór konstrukcji programistycznych (m. in. funkcji)

- biblioteka statyczna - łączona jest z programem w momencie konsolidacji. W systemach z rodziny Windows pliki bibliotek mają zazwyczaj rozszerzenia .lib lub .obj, natomiast w systemach z rodziny Unix .a lub .o.
- biblioteka dynamiczna - łączona jest z programem wykonywalnym dopiero w momencie jego wykonania. Łączenie odbywa się na poziomie systemu operacyjnego. Rozszerzenia: .dll, .so

Dyrektywy preprocesora

Dyrektywy są poprzedzone znakiem `#` są żądaniami by preprocesor wykonał jakąś akcję

`#define`

`#define WYRAZ ciąg_znaków_zastępujących`
w kompilowanym pliku każde wystąpienie słowa `WYRAZ` zostanie zastąpione ciągiem znaków.

`#define ILOSC_KOLUMN 20`

UWAGA

Od standardu C++11 lepiej stałe definiować za pomocą `constexpr`

```
constexpr int ilosc_kolumn=20;
```

Dyrektywy preprocesora

Dyrektywa kompilacji warunkowej

```
#if WARUNEK  
linie kompilacji warunkowej  
#endif
```

Przykład

```
#define PL 1  
#define EN 2  
#define LANG 2  
  
#if LANG==PL  
    cout << "polski";  
#else  
    cout << "angielski";  
#endif
```

Dyrektywy preprocesora

```
#ifdef
```

```
#ifdef nazwa  
    //instrukcje  
#endif
```

```
#ifndef
```

```
#ifndef nazwa  
    //instrukcje  
#endif
```

Jednokrotne dołączenie pliku

```
#ifndef _nazwa_h_  
#define _nazwa_h_  
  
...  
  
#endif
```

Visual C++

```
#pragma once
```

Nieznane dyrektywy są ignorowane.

Nazwy predefiniowane

```
__DATE__  data w momencie kompilacji  
__TIME__  godzina w momencie kompilacji  
__FILE__  nazwa pliku, który aktualnie jest kompilowany  
           przez kompilator  
__LINE__  definiuje numer linii
```

Nieznane dyrektywy są ignorowane.

typedef / using

Słowo kluczowe typedef służy do tworzenia nowych nazw określających istniejące już typy

```
typedef stara_nazwa nowa_nazwa;  
using nowa_nazwa=stara_nazwa;
```

Przykład

```
typedef double R;  
using N=unsigned long long int ;
```

Zastosowanie:

- istotnie krótsza nowa nazwa typu
- w rozważanym problemie istnieje możliwość łatwej zamiany jednego typu na inny np. char na long long.

Stałe const/constexpr

const

```
int f(int n);  
....  
int n;  
cin>>n;  
const int a=f(n);  
  
a=9; //błąd stałej nie można zmienić wartości  
const int b;// błąd stała musi zostać zainicjowana
```

constexpr

```
constexpr double E = 2.718281828459;  
constexpr double E2 = E*E;  
int n{};  
cin >> n;  
constexpr int b=n;// błąd
```

Stałe const a wskaźnik

```
const Typ* nazwa;
```

Wskaźnik do obiektu stałego:

```
const char* napis="Wykład";  
napis++;      //poprawne  
*napis='a';   // błąd
```

```
void drukujTekst(const char *tekst)  
{  
    cout<<*tekst++<<endl;  
    //*s=0; błąd  
}
```

Stały wskaźnik

Typ* const nazwa

```
char tab[10];  
char* const wsk=&tab[8];  
// tab--;    błąd  
*tab='A'
```

Przykład

```
char napis[]="informatyka";  
char* const wsk=&tab[5];  
...  
*(tab-5)='I';
```

Typy wyliczeniowe enum c++11

enum

```
enum class nazwa { poz_1,poz_2,...,poz_n};
```

Przykład 1

```
enum class Dni { pon=1, wt, sr , czw, pt ,sb , nd };  
Dni dzien=Dni::wt;  
dzien=1;                //błąd  
dzen=Dni::2;            //błąd  
if( dzien > Dzien::pt )  
    cout<< "Jest weekend\n";
```

Przykład 2

Wygrana wyg;

```
enum class Wygrana { pierwsza = 5 , druga , trzecia = 10,  
                    czwarta = trzecia * 2 , piata = 1000 };
```

Dawny zwykły enum

enum

```
enum nazwa { poz_1, poz_2, ..., poz_n};
```

Przykład 1

```
enum Dni { pon=1, wt, sr , czw, pt ,sb , nd };  
Dni dzien=wt;  
dzien=1;                //błąd  
if( dzien > pt )  
    cout << "Jest weekend\n";
```

Przykład 2

```
Wygrana w; // błąd pojawia się poza zasięgiem  
enum Wygrana { pierwsza = 5 , druga , trzecia = 10,  
               czwarta = trzecia * 2 , piata = 1000 };
```

Pliki nagłówkowe

Plik nagłówkowy - (rozszerzenie .h) zawiera deklaracje funkcji, struktur danych, stałych. Dołączenie tego pliku do innych plików zwiększa zasięg widoczności elementów w nim zadeklarowanych.

```
#include <nazwa_biblioteki.h>
#include <ścieżka\nazwa_biblioteki.h>
#include "własny_plik.h"
```

Wiele plików

main.cpp

```
#include<iostream>
#include"funkcje.h"

int main()
{
    int a=9,b=8;
    cout<< a + b<<endl;
    cout<<f( 3, 4)<<endl;
    system("pause");
}
```

funkcje.h

```
constexpr int c=10;
int f(int, int);
```

funkcje.cpp

```
#include"funkcje.h"

int d=9;

int f(int a, int b)
{
    return (a+b)*c;
}
```


main.cpp

```
#include<iostream>
#include"funkcje.h"
extern int d;
int main()
{
    int a=9,b=8;
    cout<< a + b<<endl;
    cout<<f( 3, 4)<<endl;
    system("pause");
}
```

funkcje.h

```
constexpr int c=10;
int f(int, int);
```

funkcje.cpp

```
#include"funkcje.h"

int d=9;

int f(int a, int b)
{
    return (a+b)*c;
}
```

Obiekt string

```
#include<string>

...

string napis1="kot";
string napis2("mysz");
...
napis1="jaki tam kot";
cin >> napis2;
cout << napis1 <<endl;
cout << napis1 + napis2 << endl;
```

Obiekt string

```
string a = "mysz";  
char s[] = "mysz";  
string b( s );  
string c = "To jest myszka.";  
if( a == b )  
    cout << "ok A " << endl;  
  
if( a.compare( s ) == 0 )  
    cout << "ok B " << endl;  
  
if( c.compare( 8, a.size(), a ) == 0 )  
    cout << "ok C" << endl;
```

Obiekt string

```
string a = "To jest napis.";

a.size(); // długość łańcucha

a[3] = J; // dostęp do 3 znaku

a.at( 3 ) = J; // jw. wyrzuca wyjątek

a.clear(); // czyści zawartość

a.find("jest"); // zwraca pozycję "jest" w a

a.npos // stała oznaczająca brak pozycji

a.erase( 3, 5 ); // kasuje od 3 pozycji 5 znaków
```

Obiekt string

```
string a = "To jest napis.";

a.replace( 0, 2, "Tam"); // zamienia od pozycji 0
                        // dwa znaki w a na elementy
                        // łańcucha "Tam"

cout << a << endl;      // "Tam jest napis."

a.insert( 9, "ciekawy " ); //wstawia łańcuch
                        //od 9 pozycji

cout << a << endl;      // "Tam jest ciekawy napis."
```

Obiekt string

```
string a = "to jest myszka";

unsigned int pozycja = 0;

pozycja = a.find( "mysz" );

if( pozycja != a.npos )
    cout << "Znalazlem na pozycji " << pozycja << endl;
else
    cout << "Nie znalazlem." << endl;
```