

Programowanie

Dariusz Wardecki, wyk.VII

Powtórzenie

```
#include<iostream>
using namespace std;

int main(int argc, char* argv[])
{
    cout << argv[0] << endl;
    char *t = argv[1];
    cout << *(t+=2) << endl;
    cout << argv[3][5] << endl;
    return 0;
}
```

```
./prog param1 param2 param3
```

Powtórzenie

```
void f(char *t, int n)
{
    for(int i = 0; i <= n; i++)
        cout << t[n-i] << endl;
}
main()
{
    char *str = "abcd";
    f(str, 4);
}
```

Funkcje atoi() i atof()

```
#include<iostream>
#include<cstdlib>

using namespace std;

int main(int argc, char* argv[ ])
{
    int n = atoi(argv[1]);
    double x = atof(argv[2]);

    cout << 2*n << endl;
    cout << x*x << endl;
    return 0;
}

./prog 3 5 10
```

Klasa string

Przykład:

```
#include<iostream>
#include<string>
using namespace std;

int main ()
{
    string str = "Hello";
    cout << str << endl;
    return 0;
}
```

Klasa string

Przykład: dodawanie stringów

```
#include<iostream>
#include<string>
using namespace std;

int main ()
{
    string str1 = "Hello ";
    string str2 = "World";
    cout << str1 + str2 << endl;
    return 0;
}
```

Klasa string

Operacje na łańcuchach - insert

```
#include<iostream>
#include<string>
using namespace std;

int main ()
{
    string str1 = "Raz dwa cztery";
    string str2 = " trzy";
    str1.insert(7,str2);
    cout << str1 << endl;
    return 0;
}
```

Klasa string

Operacje na łańcuchach

- erase

```
#include<iostream>
#include<string>
using namespace std;

int main ()
{
    string str1 = "Raz dwa trzy cztery";
    string str2 = "dwa";
    str1.erase(4,3);
    cout << str1 << endl;
    return 0;
}
```

Klasa string

Operacje na łańcuchach - compare

```
#include<iostream>
#include<string>
using namespace std;

int main ()
{
    string str1 = "Raz dwa cztery";
    string str2 = "dwa";
    if(!str1.compare(4,3,str2))
        cout << "Sukces" << endl;
    return 0;
}
```

Dynamiczna alokacja pamięci

Wspierane w g++

```
cout << "Podaj wielkosc tablicy" << endl;  
int n;  
cin >> n;  
char tab[n];
```

Zgodne z ISO C++

```
cout << "Podaj wielkosc tablicy" << endl;  
int n;  
cin >> n;  
char *tab = new char[n];
```

Dynamiczna alokacja pamięci

Tworzenie pojedynczego obiektu

```
double *wsk = new double;
```

```
int *n;  
n = new int;
```

Kasowanie obiektów

```
delete wskaznik;  
delete [] wskaznik;
```

Struktury

```
struct moja_struktura
{
    typ zmienna1;
    typ zmienna2;
    ...
};
main( )
{
    moja_struktura obiekt;
    ...
}
```

Struktury

```
struct osoba
{
    string imie;
    int wiek;
};
main( )
{
    osoba student;
    student.imie = "Paweł";
    student.wiek = 20;
}
```

Struktury

```
struct osoba
{ ... };
main( )
{
    osoba student;
    osoba *wsk = &student;
    wsk->imie = "Paweł";
    wsk->wiek = 20;
}
```

Struktury

```
struct osoba
{ ... };
main( )
{
    osoba *wsk = new osoba;

    wsk->imie = "Paweł";
    wsk->wiek = 20;
}
```