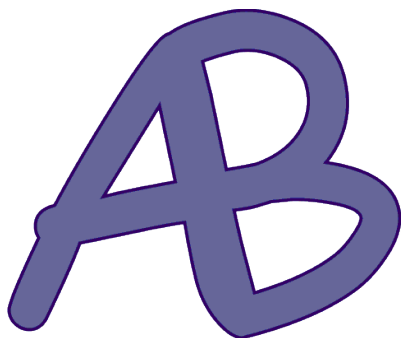


WYKŁAD: Programowanie obiektowe



dr Artur Bartoszewski
Katedra Informatyki
UTH Radom

Funkcje zaprzyjaźnione

Funkcje zaprzyjaźnione

Funkcja zaprzyjaźniona – to funkcja która ma prawo dostępu do prywatnych składników klasy.

Funkcja zaprzyjaźniona z klasą nie jest metodą tej klasy.

- ✓ Wewnątrz definicji klasy wystarczy umieścić deklarację tej funkcji poprzedzoną słowem **friend**.
- ✓ Uwaga: to nie funkcja ma twierdzić, że jest zaprzyjaźniona. To klasa ma zadeklarować, że przyjaźni się z tą funkcją i nadaje jej prawo dostępu do składników prywatnych. Zatem słowo **friend** pojawia się tylko wewnątrz definicji klasy.
- ✓ Funkcja może być zaprzyjaźniona z wieloma klasami.

Funkcje zaprzyjaźnione

```
7  class A
8  {
9      friend void funkcja(A &temp);
10 }
11
12 ▶ void funkcja(A &temp)
13 {
14
15 }
```

Funkcja „**f1**” jest zaprzyjaźniona z klasą „**A**”

Funkcje zaprzyjaźnione

```
6  class B; //nagłówek klasy B
7
8  class A
9  {
10     friend void f1(A &temp, B & temp2);
11 };
12
13 class B
14 {
15     friend void f1(A &temp, B & temp2);
16 };
17
18 void f1(A &temp, B & temp2)
19 {
20
21 }
```

Funkcja „**f1**” jest zaprzyjaźniona z klasami „**A**” i „**B**”

Funkcje zaprzyjaźnione

```
6  class RGB
7  {
8  private:
9      int R, G, B;
10 public:
11     RGB(int r=0, int g=0, int b=0): R(r), G(g), B(b) {}
12     friend void wypisz(RGB &k1);
13 };
14
15 void wypisz(RGB &k1)
16 {
17     cout<< "R="<<k1.R<<endl
18     << "G="<<k1.G<<endl
19     << "B="<<k1.B<<endl;
20 }
21 int main()
22 {
23     RGB kolor1(100,200,300);
24     wypisz(kolor1);
25     return 0;
26 }
```

Funkcje zaprzyjaźnione

```
#include <iostream>
#include <cstring>

using namespace std;

class RGB
{
private:
    int R;
    int G;
    int B;
public:
    RGB(int r=0, int g=0, int b=0): R(r), G(g), B(b) {}
    ~RGB() {cout<<"Obiekt usuniety\n";}
    void przedstawSie();
    friend int jasnoc(RGB &k1);
    friend RGB * suma(RGB &k1, RGB &k2);
};
```

Przykład

Funkcje zaprzyjaźnione

```
void RGB::przedstawSie()
{
    cout<< "R="<<R<<endl << "G="<<G<<endl << "B="<<B<<endl;
}

int jasnoc(RGB &k1)
{
    return (k1.R+k1.G+k1.B)/3;
}

RGB * suma(RGB &k1, RGB &k2)
{
    int r =(k1.R+k2.R)/2;
    int g =(k1.G+k2.G)/2;
    int b =(k1.B+k2.B)/2;
    RGB * wynik = new RGB(r,g,b);
    return wynik;
}

int main()
{
    RGB kolor1(100,200,300), kolor2(10,20,30);
    RGB *kolorWynikowy = suma(kolor1,kolor2);
    kolorWynikowy->przedstawSie();
    delete kolorWynikowy;
    return 0;
}
```

Literatura:

W prezentacji wykorzystano przykłady i fragmenty:

- Grębosz J. : ***Symfonia C++, Programowanie w języku C++ orientowane obiektowo***, Wydawnictwo Edition 2000.
- Jakubczyk K.: *Turbo Pascal i Borland C++ Przykłady*, Helion.

Warto zajrzeć także do:

- Sokół R. : ***Microsoft Visual Studio 2012 Programowanie w Ci C++***, Helion.
- Kernighan B. W., Ritchie D. M.: ***język ANSI C***, Wydawnictwo Naukowo Techniczne.

Dla bardziej zaawansowanych:

- Grębosz J. : ***Pasja C++***, Wydawnictwo Edition 2000.
- Meyers S.: ***język C++ bardziej efektywnie***, Wydawnictwo Naukowo Techniczne