

Programowanie w języku C++ (EAR1S03006)

Politechnika Białostocka - Wydział Elektryczny
Automatyka i Robotyka, semestr III, studia stacjonarne I stopnia
Rok akademicki 2021/2022

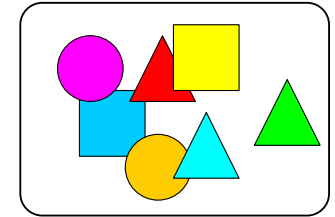
Zajęcia nr 7 (17.11.2021)

dr inż. Jarosław Forenc

Funkcje wirtualne (polimorfizm)

Przykład

- program ma wyświetlać na ekranie różne figury (kwadrat, trójkąt, koło)
- do wyświetlenia każdej figury stosowana jest inna funkcja, figury powinny być wyświetlane na ekranie w określonej kolejności



Problem: Jak zorganizować przechowywanie informacji o figurach?
Jak zorganizować wyświetlanie figur?

Rozwiązanie: Klasy + dziedziczenie + funkcje wirtualne

- definiujemy klasę podstawową (figura) oraz trzy klasy pochodne (kwadrat, trojkąt, koło)
- w klasie podstawowej umieszczamy funkcję `void rysuj()` poprzedzoną słowem `virtual` (funkcja ta nic nie robi)
- w klasach pochodnych umieszczamy funkcje o takich samych nazwach jak w klasie podstawowej - `void rysuj()` wyświetlające poszczególne figury

Przykład: funkcje wirtualne

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

class figura
{
public:
    virtual void rysuj() { };
};

class kwadrat : public figura
{
public:
    void rysuj()
    {
        cout << "Rysuj: kwadrat" << endl;
    }
};
```

klasa podstawowa figura

funkcja wirtualna rysuj()

klasa pochodna kwadrat

Przykład: funkcje wirtualne

```
class trojkat : public figura
{
public:
    void rysuj()
    {
        cout << "Rysuj: trojkat" << endl;
    }
};

class kolo : public figura
{
public:
    void rysuj()
    {
        cout << "Rysuj: kolo" << endl;
    }
};
```

klasa pochodna trojkat

klasa pochodna kolo

Funkcje wirtualne (polimorfizm)

- jeśli wskaźnikowi do klasy podstawowej (**figura**) przypiszemy adres obiektu klasy pochodnej (**kwadrat**, **trojkat**, **kolo**), to wywołując poprzez wskaźnik funkcję **rysuj()**, wywołamy funkcję odpowiadającą danemu obiektowi, np.

figura *ptr; - deklaracja wskaźnika do obiektu klasy **figura**
kwadrat kw1; - deklaracja obiektu klasy **kwadrat**
trojkat tr1; - deklaracja obiektu klasy **trojkat**
kolo kol1; - deklaracja obiektu klasy **kolo**
ptr = &kw1;
ptr->rysuj(); - wywołana zostanie funkcja **rysuj()** z klasy **kwadrat**
ptr = &tr1;
ptr->rysuj(); - wywołana zostanie funkcja **rysuj()** z klasy **trojkat**
ptr = &kol1;
ptr->rysuj(); - wywołana zostanie funkcja **rysuj()** z klasy **kolo**

- mówimy, że w powyższym przykładzie wystąpił **polimorfizm** (wielopostaciowość)

Przykład: funkcje wirtualne

```
int main(void)
{
    kwadrat kwadrat1, kwadrat2;
    trojkat trojkat1, trojkat2;
    kolo kolo1, kolo2;
    figura *lista[6];

    lista[0] = &trojkat1;
    lista[1] = &kwadrat1;
    lista[2] = &kolo1;
    lista[3] = &kwadrat2;
    lista[4] = &kolo2;
    lista[5] = &trojkat2;

    for (int i=0; i<6; i++)
        lista[i]->rysuj();
}
```

Rysuj: trojkat
Rysuj: kwadrat
Rysuj: kolo
Rysuj: kwadrat
Rysuj: kolo
Rysuj: trojkat