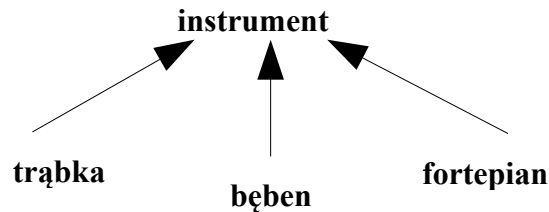


## VIII Programowanie obiektowe

### 3. Metody wirtualne. Przykład ilustrujący ich użyteczność

Zacznijmy od przykładu, spójrzmy na poniższą hierarchię klas:



a oto realizacja tych klas, zwróć uwagę na *virtual* przy deklaracji metody `wydaj_dzwiek()`, będzie to nasza metoda wirtualna.

```
// plik instrume.h
#include <iostream>
class instrument
{
    int cena;
public:
    virtual void wydaj_dzwiek()
    {
        cout << "nieokreslony brzdek!\n";
    }
    //....
}

class trąbka: public instrument
{
public:
    void wydaj_dzwiek()
    {
        cout<<" Tra-ta-ta!\n";
    }
};

class beben: public instrument
{
public:
    void wydaj_dzwiek()
    {
        cout<<"bum-bum-bum!\n";
    }
};

class fortepian: public instrument
```

```
{
    public:
    void wydaj_dzwiek()
    {
        cout<<"Plim-plim-plim!\n";
    }
};
```

a teraz przystąpmy do realizacji naszej głównej funkcji programu, czyli main()

```
//plik main.cpp
#include "instrume.h"
int main()
{
    instrument jakis_instrument;
    trabka zlota_trabka;
    fortepian drewniany_fortepian;
    beben moj_werbel;

    cout<<"Wywolywanie funkcji składowej wydaj_dzwiek()\n";

    jakis_instrument.wydaj_dzwiek();
    zlota_trabka.wydaj_dzwiek();
    drewniany_fortepian.wydaj_dzwiek();
    moj_werbel.wydaj_dzwiek();

    cout<<"Wywolywanie funkcji na rzecz obiektu pokazywanego
        wskaźnikiem instrumentu\n";

    instrument *wskinstr;

    wskinstr=&jakis_instrument;
    wskinstr->wydaj_dzwiek();

    wskinstr=&zlota_trabka;
    wskinstr->wydaj_dzwiek();

    wskinstr=&drewniany_fortepian;
    wskinstr->wydaj_dzwiek();

    wskinstr=&moj_werbel;
    wskinstr->wydaj_dzwiek();
}
```

a teraz najważniejsze, czyli co pokaże nam program na wyjściu:

```
Wywolywanie funkcji składowej wydaj_dzwiek()
nieokreslony brzdek!
Tra-ta-ta!
```

```
Plim-plim-plim!  
Bum-bum-bum!  
Wywolywanie funkcji na rzecz obiektu pokazywanego wskaźnikiem  
instrumentu  
nieokreslony brzdek!  
Tra-ta-ta!  
Plim-plim-plim!  
Bum-bum-bum!
```

Zwrócić uwagę przede wszystkim trzeba na cztery ostatnie linijki(pogrubione), które wyrzucił nam program na wyjściu. To wszystko zasługa tego, że metoda `wydaj_dzwiek()`, jest metodą wirtualną(*virtual*). Gdybyśmy usuneli słowo *virtual* z deklaracji funkcji, otrzymalibyśmy następujące wyjście:

```
Wywolywanie funkcji składowej wydaj_dzwiek()  
nieokreslony brzdek!  
Tra-ta-ta!  
Plim-plim-plim!  
Bum-bum-bum!  
Wywolywanie funkcji na rzecz obiektu pokazywanego wskaźnikiem  
instrumentu  
nieokreslony brzdek!  
nieokreslony brzdek!  
nieokreslony brzdek!  
nieokreslony brzdek!
```

Na jakiej podstawie kompilator określa, którą funkcję składową wykonać?

Mamy dwie możliwości:

1. Kompilator widzi, że ma do czynienia ze wskaźnikiem do pokazywania na *instrument*. Obok tego wskaźnika stoi wywołanie funkcji składowej `wydaj_dzwiek()`. Sięga więc "na ślepo" do klasy *instrument* i uruchamia tę funkcję. Kompilator uruchomił więc funkcję właściwą typowi wskaźnika.
2. Kompilator widzi, że ma do czynienia ze wskaźnikiem do pokazywania na *instrument*. Nie sięga na ślepo do klasy *instrument*, tylko sprawdza na co dokładnie wskazuje wskaźnik. Orientuje się, że wskaźnik pokazuje na obiekt klasy np. *trabka* zatem: orientując się według typu obiektu wykonuje funkcję właściwą pokazywanemu obiektowi. (A nie typowi wskaźnika)

Nasz program wykonał wariant oznaczony numerem 2, sprawiła to właśnie deklaracja funkcji jako wirtualnej, czyli użyliśmy słowa *virtual*. W powyższym przykładzie użyliśmy wskaźników, jednak **sytuacja wygląda tak samo w przypadku referencji**.

Destruktor także może być metodą wirtualną.