

# Polimorfizm

## Przegląd zagadnień

- Funkcje wirtualne: wstęp
- Przykład: funkcje wirtualne
- Przykład: virtual i new
- Metody klasy object
- Wzorzec projektowy adapter

## Funkcje wirtualne - wstęp

- Polimorfizm: wielość form
- Decyzja, którą metodę wywołać podejmowana jest na podstawie typu obiektu, a nie typu zmiennej
- Słowo kluczowe virtual i override
- Funkcje wirtualne nie mogą być prywatne oraz static
- Tablica funkcji wirtualnych (ang. virtual-table lub v-table).

```
class Instrument {
    public void Graj1() {
        Console.WriteLine("Cisza");
    }
    public virtual void Graj2() {
        Console.WriteLine("Cisza 2!!!");
    }
}
class Traba : Instrument {
    public new void Graj1() {
        Console.WriteLine("Tra ta ta, tra ta ta");
    }
    public override void Graj2() {
        Console.WriteLine("Tra ta ta, tra ta ta");
    }
}
class Beben : Instrument {
    public new void Graj1() {
        Console.WriteLine("Bum bum bum");
    }
    public override void Graj2() {
        Console.WriteLine("Bum bum bum");
    }
}
```

Przykład: funkcje wirtualne (1)

### Przykład: funkcje wirtualne (2)

```
static void Main(string[] args)
{
    Instrument instr = new Instrument();
    Console.WriteLine("A - traba");
    Console.WriteLine("B - beben");
    char c = Console.ReadKey().KeyChar;
    switch (c) {
        case 'a':
            instr = new Traba();
            break;
        case 'b':
            instr = new Beben();
            break;
    }

    instr.Graj1(); //zawsze Cisza
    instr.Graj2(); // nie wiadomo, "użytkownik wybiera"
    Console.ReadKey();
}
```

### Przykład: virtual i new

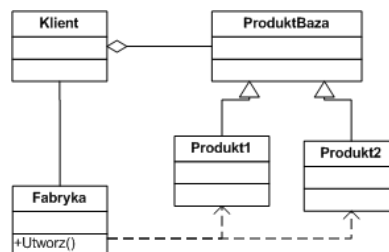
```
class A{
    public virtual void f(){
        Console.WriteLine("Tutaj A");
    }
}
class B : A {
    public override void f(){
        Console.WriteLine("Tutaj B");
    }
}
class C : B{
    public new virtual void f(){
        Console.WriteLine("Tutaj C");
    }
}
class D : C{
    public override void f(){
        Console.WriteLine("Tutaj D");
    }
}

static void Main(string[] args){
    A a = new A();
    a.f(); //Tutaj A
    a = new B();
    a.f(); //Tutaj B
    a = new C();
    a.f(); //Tutaj B
    a = new D();
    a.f(); //Tutaj B
    C c = new D();
    c.f(); //Tutaj D
    Console.ReadKey();
}
```

### Metody klasy object

- ToString
  - domyślna wersja tej metody zwraca w pełni kwalifikowaną nazwę klasy
- Equals
  - domyślnie porównuje referencje
- GetHashCode

### Wzorec projektowy Factory Method



Wzorzec projektowy adapter