



GRUPA .NET

● GRUPA

NET

develop. future is yours.



Wartość vs referencja

Value vs reference

<https://michalchecinski.pl>



MICHAŁ CHĘCIŃSKI

Programowanie | Produktywność | Technologie

[STRONA GŁÓWNA](#) [PRODUKTYWNOŚĆ](#) [PROGRAMOWANIE](#) [TECHNOLOGIE](#) [PRZEMYSŁENIA](#) [INFO](#)



Recenzja książki 10 kroków do maksymalnej produktywności





MICHAŁ CHĘCIŃSKI

Programowanie | Produktywność | Technologie

[STRONA GŁÓWNA](#) [PRODUKTYWNOŚĆ](#) [PROGRAMOWANIE](#) [TECHNOLOGIE](#) [PRZEMYSŁENIA](#) [INFO](#)

KATEGORIA: DAJ SIĘ POZNAĆ 2017



Daj Się Poznać 2017

Wpisy dotyczące konkursu [Daj Się Poznać 2017](#) podczas którego realizuję projekt Codziennik. Opisałem go w [tym wpisie](#). Kod źródłowy mojego projektu znajduje się [w tym repozytorium na Githubie](#).





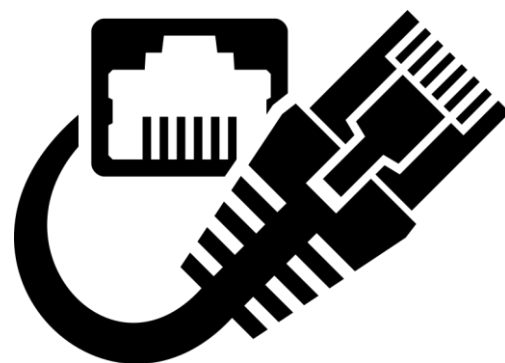
referencja



GRUPA

NET

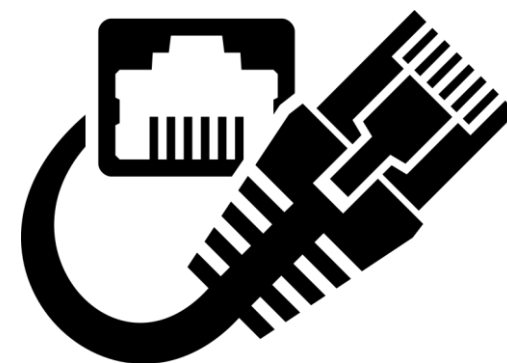
develop. future is yours.



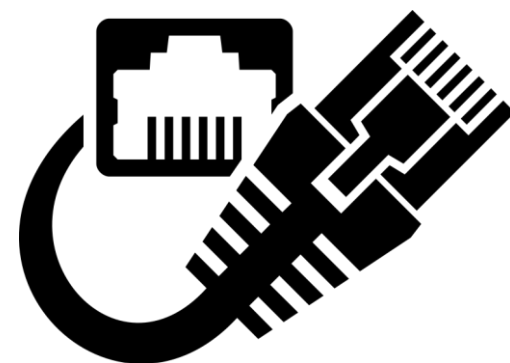
GRUPA

NET

develop. future is yours.



Wartość
„kopia oryginału”





MICHAŁ CHĘCIŃSKI

Programowanie | Produktywność | Technologie

[STRONA GŁÓWNA](#)

[PRODUKTYWNOŚĆ](#)

[PROGRAMOWANIE](#) ▾

[TECHNOLOGIE](#)

[PRZEMYSŁENIA](#)

[INFO](#) ▾



Recenzja książki 10 kroków do maksymalnej produktywności



Przez wartość

```
static int ByValue(int a)  
{  
    ...  
}
```

Typ prosty

Przez referencję

```
static int ByReference(Rectangle rectangle)
{
    ...
}
```

Typ „złożony” –
klasa/struktura

Wartość

```
static int AddByValue(int a)  
{  
    a = 2;  
    return a + a;  
}
```

Metody statyczne

Wywołanie metody
statycznej NIE wymaga
tworzenia obiektu

Metody statyczne

~~new~~

Używanie metod statycznych

```
Console.WriteLine();
```

Static

```
public class Class
{
    public static void StaticMethod()
    {
    }
}
```

Static

```
public class Operations
{
    public static int Add(int a, int b)
    {
        return a + b;
    }
}
```

Static

```
Operations operations = new Operations();  
operations.Add(5, 2);
```


Static

`Operations op`
`operations.Add(Operations());`

Static

```
Operations.Add(5, 2);
```

```
public class NonStaticClass
{
    public static void StaticMethod()
    {
    }

    public void NonStaticMethod()
    {
    }
}
```

Klasa statyczna

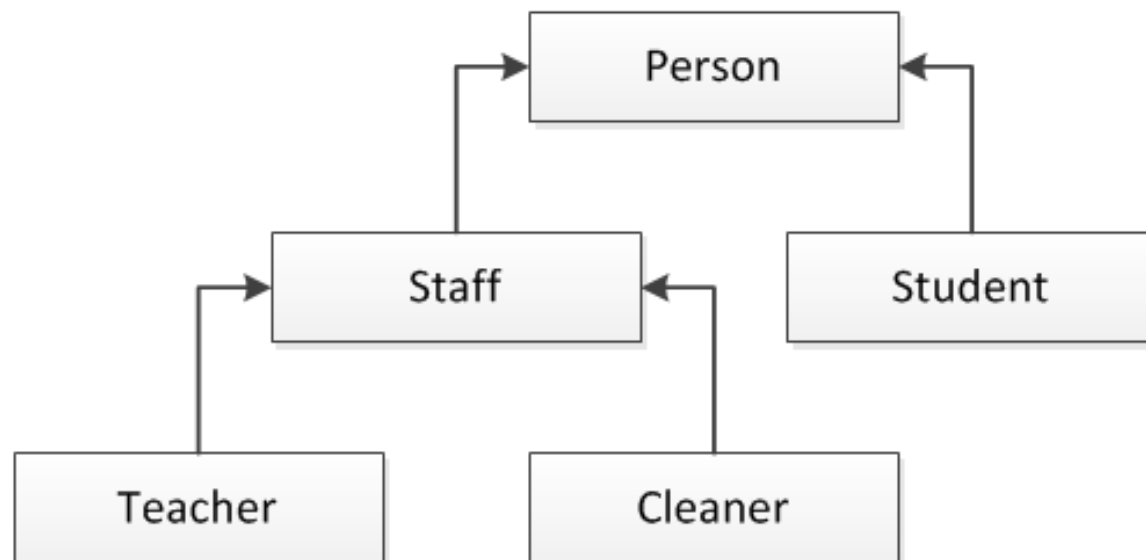
```
public static class StaticClass
{
    public static void StaticMethod()
    {
    }
}
```



```
public static class StaticClass
{
    //konstruktor klasy statycznej
    static StaticClass()
    {
        //ten kod wywoła się przy pierwszym wywołaniu
        //jednej z metod statycznych tej klasy statycznej
    }

    public static void StaticMethod()
    {
    }
}
```

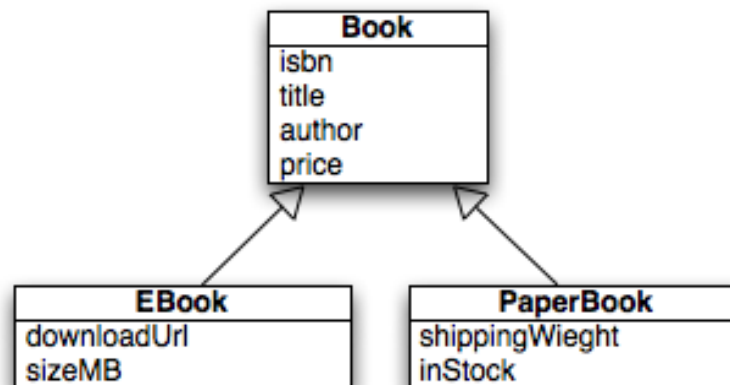
Dziedziczenie



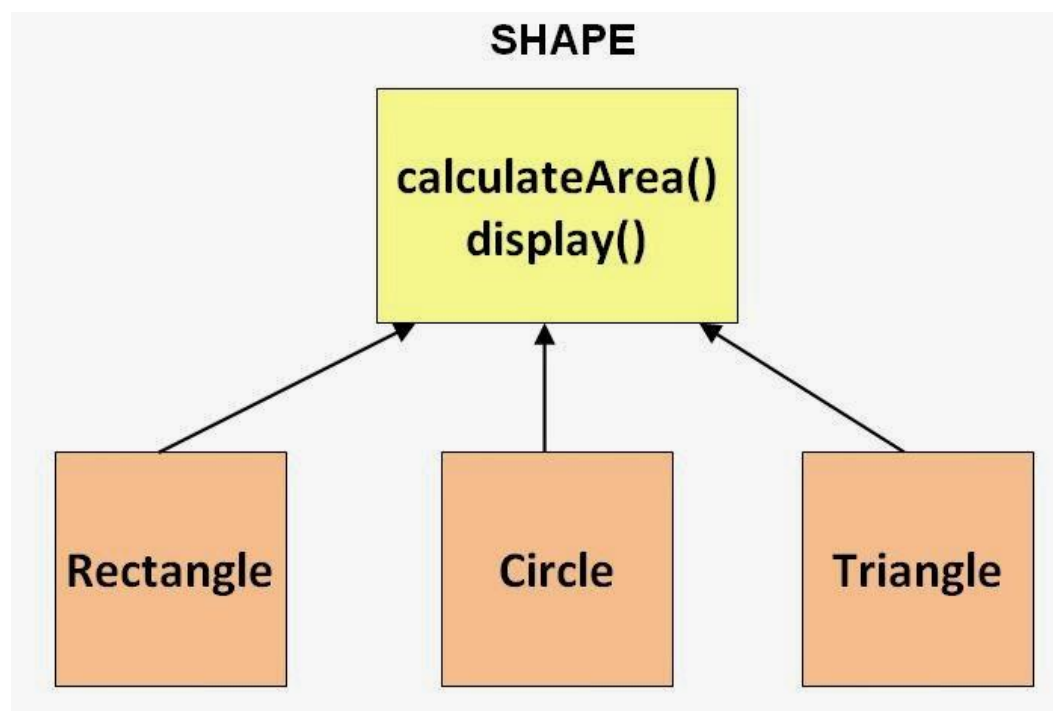
Dziedziczenie

- po co je wykorzystujemy

BOOK
ID (PK)
TYPE
ISBN
TITLE
AUTHOR
PRICE
DOWNLOAD_URL
SIZE_MB
SHIPPING_WEIGHT
IN_STOCK



Przykład



Protected



Sealed



Virtual, abstract and override

```
public abstract void FireYourself();  
public virtual void Work()  
{  
    Console.WriteLine("I am working");  
}
```

```
public override void FireYourself()  
{  
    Console.WriteLine("Idę do dziekana się zwolnić.");  
}
```


IS & AS

```
if (person is ITStudent)
{
    ... (person as ITStudent).HackGrades();
}
```


Witaj świecie!

```
Hello, World_
```

Pytania



Do zobaczenia !