



UNIWERSYTET  
WSB **MERITO**  
CHORZÓW KATOWICE

WZORCE PROJEKTOWE

SPRAWOZDANIE

Ćwiczenie 4 - Adapter

Sebastian Marcinkowski  
Informatyka, Programowanie  
31\_Inf\_P\_NW\_n1\_6

Rok akademicki 2024/25

# 1. Cel

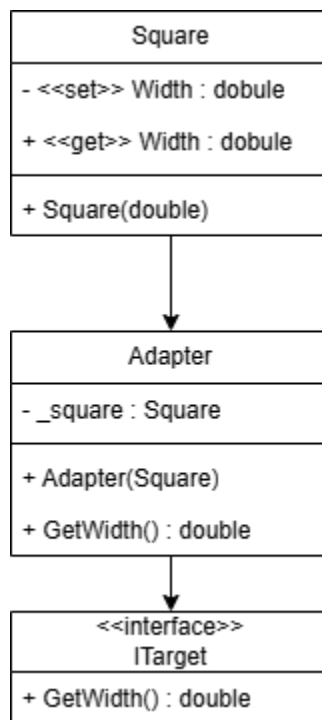
Celem ćwiczenia jest zaprezentowanie rozwiązania problemu integracji istniejącej już klasy Square z funkcją oczekującej obiektu z metodą GetArea przy pomocy wzorca projektowego Adapter.

## 2. Opis rozwiązania

Do implementacji zostanie wykorzystany język C#, więc na zarówno na diagramie UML jak i pseudokodzie mogą być wykorzystywane elementy specyficzne dla tego języka, nie powinno to jednak stanowić problemu przy przenoszeniu rozwiązania do innej technologii.

W rozwiązaniu odpowiednio:

- Klasa Square reprezentuje klasę z istniejącą już właściwością Width, implementacji tej klasy już nie można zmienić.
- Klasę Adapter która zawiera pole typu Square oraz implementującą interfejs ITarget który wymusza implementację metody GetArea.
- Interfejs ITarget wymusza implementację metody GetArea.
- Klasa Program prezentuje działanie całości.



Rys. 1. Diagram UML.

## Pseudokod

```
class Square
{
    public double Width { get, private set }

    public Square(double width)
    {
        Width = width
    }
}

interface ITarget
{
    double GetWidth()
}

public class Adapter : ITarget
{
    private Square _square

    public Adapter(Square square)
    {
        _square = square
    }

    public double GetWidth()
    {
        return _square.Width * _square.Width
    }
}
```

### 3. Implementacja

#### **Square.cs**

```
namespace s_marcinkowski_135898_cw4;

public class Square
{
    public double Width { get; private set; }

    public Square(double width)
    {
        Width = width;
    }
}
```

#### **ITarget.cs**

```
namespace s_marcinkowski_135898_cw4;

public interface ITarget
{
    double GetWidth();
}
```

## **Adapter.cs**

```
namespace s_marcinkowski_135898_cw4;
```

```
public class Adapter : ITarget
```

```
{
```

```
    private Square _square;
```

```
    public Adapter(Square square)
```

```
    {
```

```
        _square = square;
```

```
    }
```

```
    public double GetWidth()
```

```
        => _square.Width * _square.Width;
```

```
}
```

## **Program.cs**

```
using s_marcinkowski_135898_cw4;
```

```
var width = 5;
```

```
Square square = new Square(width);
```

```
ITarget adapter = new Adapter(square);
```

```
var results = adapter.GetWidth();
```

```
Console.WriteLine($"Pole kwadratu o boku {width} wynosi {results}.");
```

## 4. Podsumowanie

Do implementacji wykorzystano wzorzec Adapter ponieważ wymagano klasy zawierającej metodę `GetWidth` a klasa `Square` nie oferowała takiej funkcjonalności. Dzięki zastosowaniu wzorca Adapter można teraz wywołać metodę `GetWidth` dostarczaną przez odpowiedni interfejs dzięki czemu brak metody `GetWidth` w klasie `Square` nie stanowi problemu.

## Lista załączników