Zaawansowane programowanie obiektowe - wykład 3

dr Piotr Jastrzębski

Wzorce projektowe

Wzorce projektowe

Wzorzec projektowy (ang. design pattern) – uniwersalne, sprawdzone w praktyce rozwiązanie często pojawiających się, powtarzalnych problemów projektowych. Pokazuje powiązania i zależności pomiędzy klasami oraz obiektami i ułatwia tworzenie, modyfikację oraz pielęgnację kodu źródłowego. Jest opisem rozwiązania, a nie jego implementacją. Wzorce projektowe stosowane są w projektach wykorzystujących programowanie obiektowe.

Elementy wzorca projektowego

Wzorzec projektowy składa się z czterech podstawowych elementów:

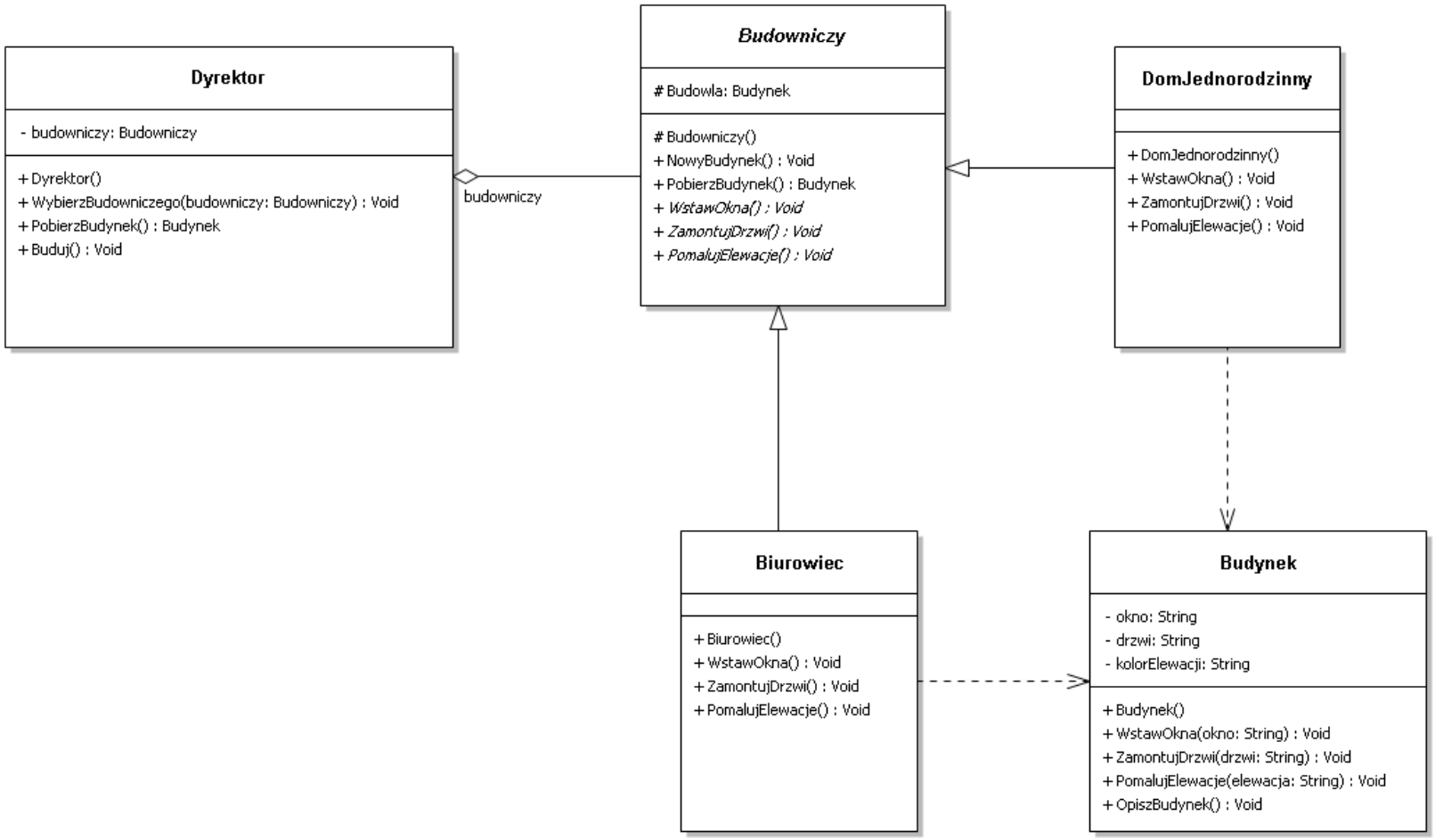
- nazwy wzorca;
- problemu – opisuje sposoby rozpoznawania sytuacji, w których możemy zastosować dany wzorzec oraz warunki jakie muszą zostać spełnione, by jego zastosowanie miało sens;
- rozwiązania – opisuje elementy rozwiązania: ich relacje, powiązania oraz obowiązki, zawiera także wskazówki implementacyjne dla różnych technologii;
- konsekwencji – zestawienie wad i zalet stosowania wzorca, uwzględniające informacje o jego brakach oraz kosztach rozwoju i utrzymania systemu wykorzystującego dany wzorzec.

Wzorce kreacyjne

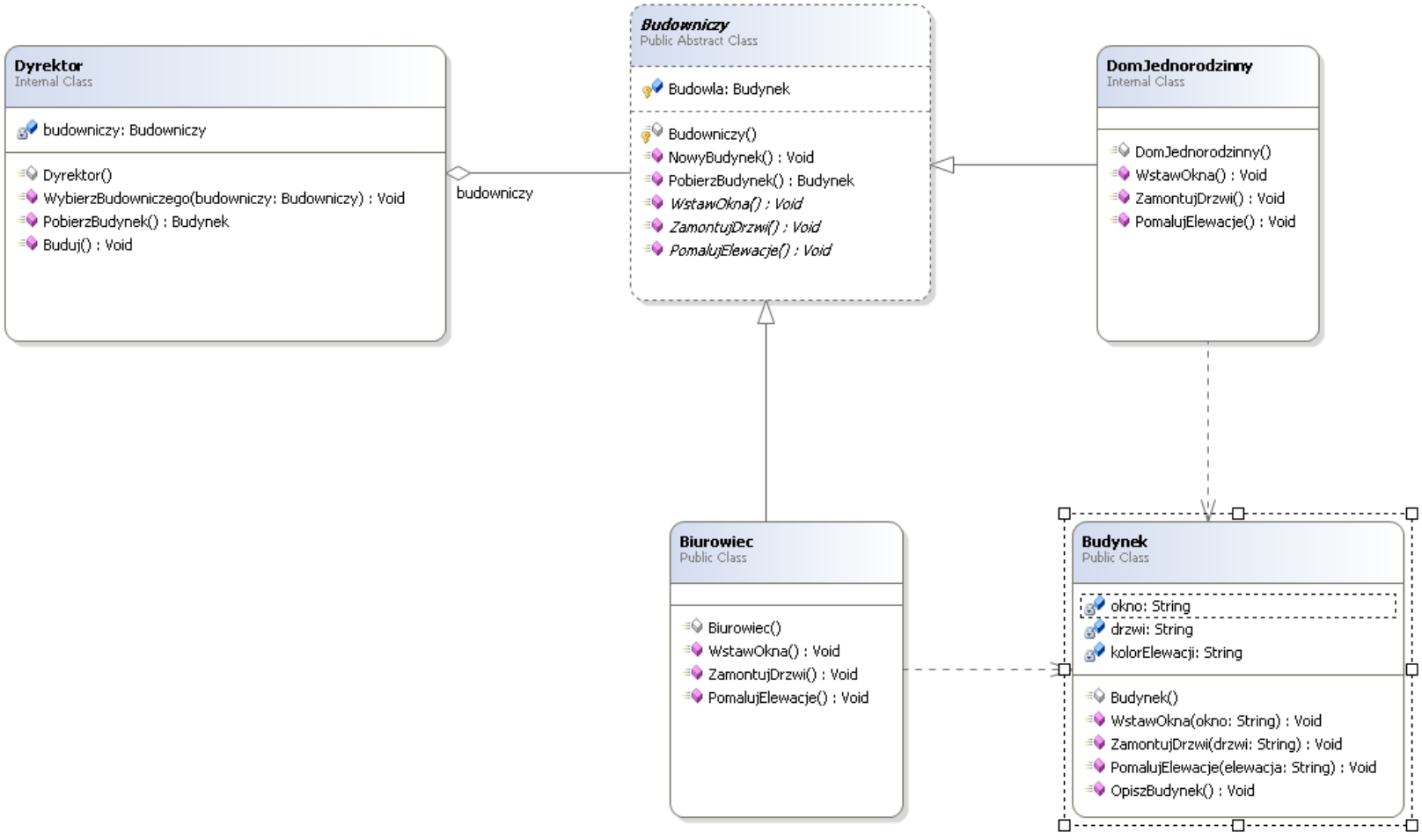
Wzorce konstrukcyjne (kreacyjne) pozwalają w sposób abstrakcyjny tworzyć i konfigurować obiekty w celu ich wielokrotnego użycia i zachowania niezależności systemu od sposobu ich tworzenia.

Budowniczy

- cel: rozdzielenie sposobu tworzenia obiektów od ich reprezentacji
- proces tworzenia obiektu podzielony jest na kilka mniejszych etapów a każdy z tych etapów może być implementowany na wiele sposobów
- zastosowanie:
  - węzły XMLa
  - konwertowanie tekstu, zdjęć, filmów z jednego formatu na drugi.



Rysunek 1: Diagram UML Budowniczego



Rysunek 2: Diagram UML Budowniczego

Kod do analizy: https://github.com/pjastr/ZPO_programy/tree/master/Budowniczy/Budowniczy

Przykład praktyczny `StringBuilder`

https://docs.microsoft.com/pl-pl/dotnet/api/system.text.stringbuilder?view=netframework-4.8

Fabryka abstrakcyjna

Przeanalizujmy przykład: http://tomaszjarzynski.pl/fabryka-abstrakcyjna-wzorzec-projektowy-abstract-factory/

Zastosowania:

- komunikacja z różnymi api
- wyświetlanie tekstu z różnych formatów plików
- obsługa różnych typów baz danych

Przykład

https://stackoverflow.com/questions/2280170/why-do-we-need-abstract-factory-design-pattern?rq=1

Metoda wytwórcza

Przykład do przeanalizowania: http://tomaszjarzynski.pl/metoda-wytworcza-wzorzec-projektowy-factory-method/

Zastosowania:

- ukrycie przed klientem implementacji
- sterowniki

Prototyp

Prototyp – kreacyjny wzorzec projektowy, którego celem jest umożliwienie tworzenia obiektów danej klasy bądź klas z wykorzystaniem już istniejącego obiektu, zwanego prototypem.

Przykład do analizy: https://pl.wikibooks.org/wiki/Kody_%C5%BAr%C3%B3d%C5%82owe/Prototyp_(wzorzec_projektowy)#C#

Zastosowania

- duża liczba obiektów tego samego typu
- chcemy otrzymać prawidłową kopię obiektu

Singleton

Singleton - kreacyjny wzorzec projektowy, którego celem jest ograniczenie możliwości tworzenia obiektów danej klasy do jednej instancji oraz zapewnienie globalnego dostępu do stworzonego obiektu

```
public sealed class Singleton
{
    private static Singleton instance = null;

    private Singleton()  {  }

    public static Singleton Instance()
    {
        if (instance == null)
        {
            instance = new Singleton();
        }
        return instance;
    }

}
```