# DESIGN PATTERNS VIA C#

Praca licencjat – Wzorce projektowe w języku C#

CZĘSC 0 – Struktura

- Definicja oraz motywacja

Wzorzec projektowy – abstrakcyjny przykład poprawnego wykorzystania ograniczonej ilości kombinacji technik OOP. Wzorzec pokazuje jak poprawnie organizować relację między klasą a obiektem.

- Klasyfikacja

**Wzorce kreacyjne:**

1. Abstract Factory (obiektowy związek) Używany: często

2. Builder (obiektowy związek)

3. Factory Method (obiektowy związek)

4. Prototype (obiektowy związek)

5. Singleton (obiektowy związek)

**Wzorce strukturalne:**

**Wzorce czynnościowe:**

Czym róźnią się wzorce projektowe o podobnej klasyfikacji?

Przykład (rowery): jeżeli ktoś próbował jeźdić na szosejnym rowerze, napewno żalował o tym, że musi siedzieć w zwiniętej pozycji ale po jaździe trudno nie zauważyć, że dystancja przejechana za jednostkę czasu jest dość spora. Natomiast wybierając mountain bike można dość komfortowo poruszać się off road przy czym szybkość takiego egzemplaru nie jest taka wysoka jak na poprzednim rodzaju rowera. Też są rowery przystosowane wyłącznie do miasta. Takie rowery mają miękie siódełka oraz opony. Jechać w parku na podobnym rowerze – to jedno zadowolenie. Dokładnie tak wygląda sytuacja ze wzorcami projektowymi. Może się wydawać, że oni spełniają jedną funkcję. Ale z innej strony robią to w róźny sposób. Dla tego, żeby najlepiej dopasować wzorzec projektowy do sytuacji trzeba rozważyć kryterie (gdzie oraz jak długo ja chcę jechać), żeby wybrać sobie najwygodniejszy rower.

// Tabele

// Lista Wzorzec - Cel

|  |  |  |  |
| --- | --- | --- | --- |
| Wzorzec | Rodzaj / Zakres | Przeznaczenie | Użycie |
|  | Kreacyjne |  |  |
| Abstract Factory | Obiektowy | 103 |  |
| Builder | Obiektowy | 214 | 2 |
| Factory Method | Obiektowy | 120 | 5 |
| Prototype | Obiektowy | 121 | 3 |
| Singleton | Obiektowy | 210 | 4 |
|  | Strukturalne |  |  |
| Adapter | Obiektowy / Klasowy | 20 | 4 |
| Bridge | Obiektowy | 53 | 3 |
| Composite | Obiektowy | 11 | 4 |
| Decorator | Obiektowy | 4 | 3 |
| Facade  Flyweight  Proxy  Chain of Responsibility  Command  Interpreter  Iterator  Mediator  Memento  Observer  State  Strategy  Template Method  Visitor | Obiektowy  Obiektowy  Obiektowy  *Czynnościowe*  Obiektowy  Obiektowy  Obiektowy  Obiektowy  Obiektowy  Obiektowy  Obiektowy  Obiektowy  Obiektowy  Obiektowy  Obiektowy | 52 | 5 |

CZĘSC 1 – Wzorce kreacyjne

Wzorce kreacyjne – wszystkie wzorce kreacyjne opisane poniżej opisują proces tworzenia obiektów oraz sposoby na abstrahowanie tego procesu. Każdy wzorzec zawiera w sobie abstraktne fabryczne metody, pozwalające na abstraktne opisanie procesów tworzenia egzemplarów typu.

Cel: Tworzenie obiektowego systemu w taki sposób, żeby ten system nie zależał od sposobu tworzenia w nim obiektów, kompozycji obiektów oraz wewnętrznego stanu obiektów.

\*Obiektowy system – program złożony ze złączonych obiektów które są we wzajemnym stosunku.

# Abstract factory (Kit)

## Cel:

Przedstawić klientowi interfejs dla tworzenia związanych między sobą obiektów-produktów, hermetyzując informację o konkretnych (nie abstrakcyjnych) klasach tworzonych obiektów.

## Cel (Krótko):

Tworzenie rodzin współdziałających obiektów.

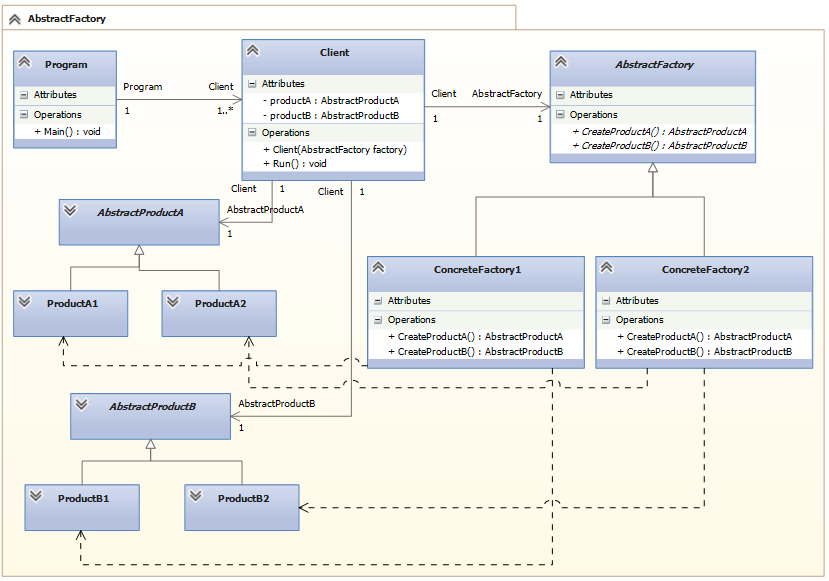
## Rekomendacje użycia:

Tworzenie złożonych kontrolek UI.

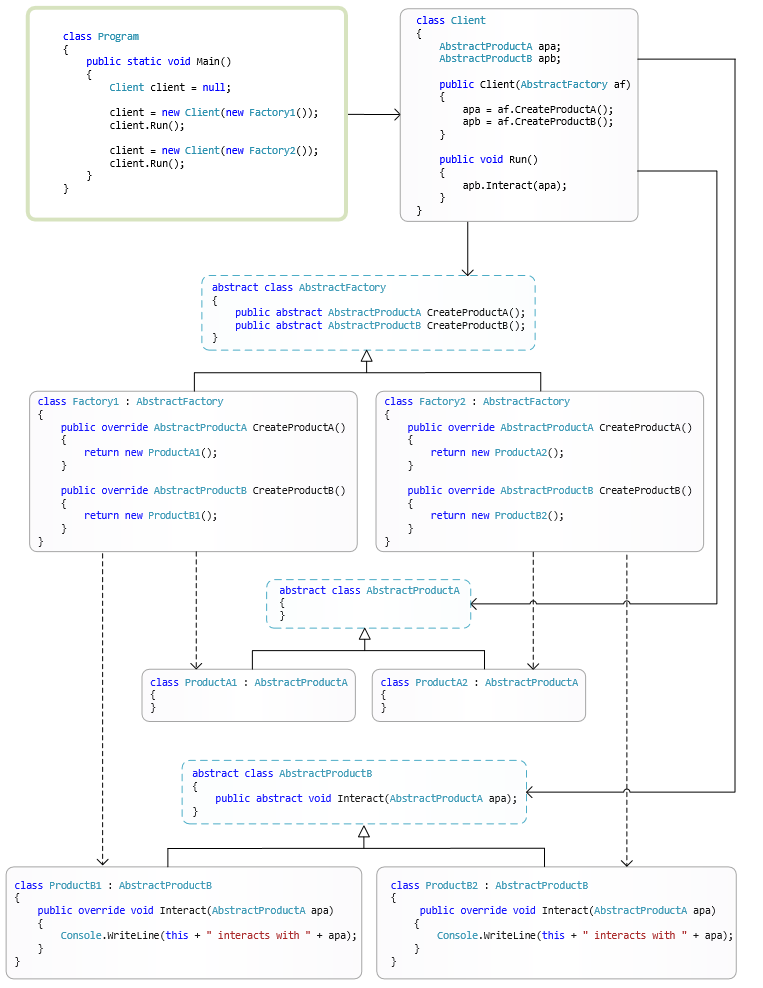
## Krótki opis-algorytm:

1. Tworzymy klasy abstrakcyjne dla typów produktów oraz fabryk.
2. Opisujemy interfejsy działań z każdym typem produktu oraz fabryki.
3. Tworzymy konkretną klasę Client w której abstrakcyjnie opisujemy procesy tworzenia egzemplarzy typów produktów oraz możliwości wykorzystania tych typów produktów przez ich abstraktne interfejsy. Client realizuje też hermetyzację wariacji.

## Diagram UML:



## Struktura w języku C#:



## Użycia w .NET:

Microsoft.Build.Tasks.CodeTaskFactory

http://msdn.microsoft.com/library/microsoft.build.tasks.codetaskfactory.aspx

Microsoft.Build.Tasks.XamlTaskFactory

http://msdn.microsoft.com/library/microsoft.build.tasks.xamltaskfactory.aspx

Microsoft.IE.SecureFactory

http://msdn.microsoft.com/library/microsoft.ie.securefactory(v=vs.90).aspx

System.Activities.Presentation.Model.ModelFactory http://msdn.microsoft.com/library/system.activities.presentation.model.modelfactory.aspx

System.Data.Common.DbProviderFactory

http://msdn.microsoft.com/library/system.data.common.dbproviderfactory.aspx

System.Data.EntityClient.EntityProviderFactory http://msdn.microsoft.com/library/system.data.entityclient.entityproviderfactory.aspx

System.Data.Odbc.OdbcFactory

http://msdn.microsoft.com/library/system.data.odbc.odbcfactory.aspx

System.Data.OleDb.OleDbFactory

http://msdn.microsoft.com/library/system.data.oledb.oledbfactory.aspx

System.Data.OracleClient.OracleClientFactory http://msdn.microsoft.com/library/system.data.oracleclient.oracleclientfactory.aspx

System.Data.Services.DataServiceHostFactory

http://msdn.microsoft.com/library/system.data.services.dataservicehostfactory.aspx

System.Data.SqlClient.SqlClientFactory

http://msdn.microsoft.com/library/system.data.sqlclient.sqlclientfactory.aspx

System.ServiceModel.ChannelFactory

http://msdn.microsoft.com/library/system.servicemodel.channelfactory.aspx

System.Threading.Tasks.TaskFactory http://msdn.microsoft.com/library/system.threading.tasks.taskfactory(v=vs.110).aspx

System.Web.Compilation.ResourceProviderFactory http://msdn.microsoft.com/library/system.web.compilation.resourceproviderfactory.aspx

System.Web.Hosting.AppDomainFactory http://msdn.microsoft.com/library/system.web.hosting.appdomainfactory(v=vs.90).aspx

System.Xml.Serialization.XmlSerializerFactory <http://msdn.microsoft.com/library/system.xml.serialization.xmlserializerfactory(v=vs.90).aspx>

Oraz inne

CZESC 4 – Praktyka

Wstep do aplikacji

Poszczegolne przykłady praktycznego wykorzystywania