Factory method

Nazwa: Factory method

Drugie nazwy: Virtual Constructor

Klasyfikacja: według celu – kreacyjny, według stosowania – do klas

Jak często się używa: bardzo często

Przeznaczenie: być fundamentem wszystkich wzorców kreacyjnych

Opis – bardzo prosty wzorzec który jest głównym, fundamentalnym składnikiem pozostałych wzorców kreacyjnych. Wzorzec zapewnia abstraktowy interfejs, który możemy użyć do tworzenia obiektów-produktów. Użytkownicy mają możliwość wybierać konkretny produkt który trzeba ztworzyć.

\*Strona architekta:

Architekt tworzy abstraktowy szkielet mechanizmu tworzenia produktu.

Architekt deleguje realizatoru odpowiedzialność za wybór konkretnego produktu do stworzenia.

\*Strona programisty-realizatora:

Programista realizuje konkretne klasy produktów.

Programista przyjmuje odpowiedzialność i decyduje jaki konkretny produkt trzeba stworzyć.

Kiedy użyć:

- Jeżeli jest wątpliwość jaki wzorzec kreacyjny użyć do tworzenia obiektu, najlepszym rozwiązaniem jest użycie Factory Method o ile mamy możliwość łatwego transformowania do dowolnego z pozostałych kreacyjnych wzorców.

C# osobliwości:

Wykorzystanie Generics – pozwala nie tworzyć kylka ConcreteCreator klasów, a przy tworzeniu podać typ tworzonego produktu:

ICreator creator = new StandardCreator();

IProduct productA = creator.CreateProduct<ProductA>();

IProduct productB = creator.CreateProduct<ProductB>();

IProduct productC = creator.CreateProduct<ProductC>();

Nazewnictwo:

Warto nadawać takie nazwy, żeby było łatwo zrozumieć że jest używany kreacyjny wzorzec. Przykładem jest CreateApplication jako fabryczna metoda tworząca aplikację.

Wykorzystanie w .NET:

Jest fundamentem wszystkich kreacyjnych wzorców, czyli jest wykorzystywany wszędzie gdzie można spotkać wzorce kreacyjne.

UML

C#