Nowoczesne języki programowania obiektowego

Fabryka

Prosta fabryka (Simple Factory)

Cel

- Stworzenie wygodnego interfejsu do tworzenia obiektów.
- Wsparcie dla wyboru klasy i konstruktora (przy tworzeniu instancji obiektu)
- Uniezależnienie od operatora new

Zastosowania:

- Rozszerzenie hierarchii obiektów bez ingerencji w wielu miejscach istniejącego kodu.

Sytuacja bez wykorzystania wzorca

```
Chart wykres = null;

if (typ == "bar")
    wykres = new BarChart();
else if (typ == "circle")
    wykres = new CircleChart();
else if (typ == "histogram")
    wykres = new HistogramChart();
else
    wykres = new PieChart();
```

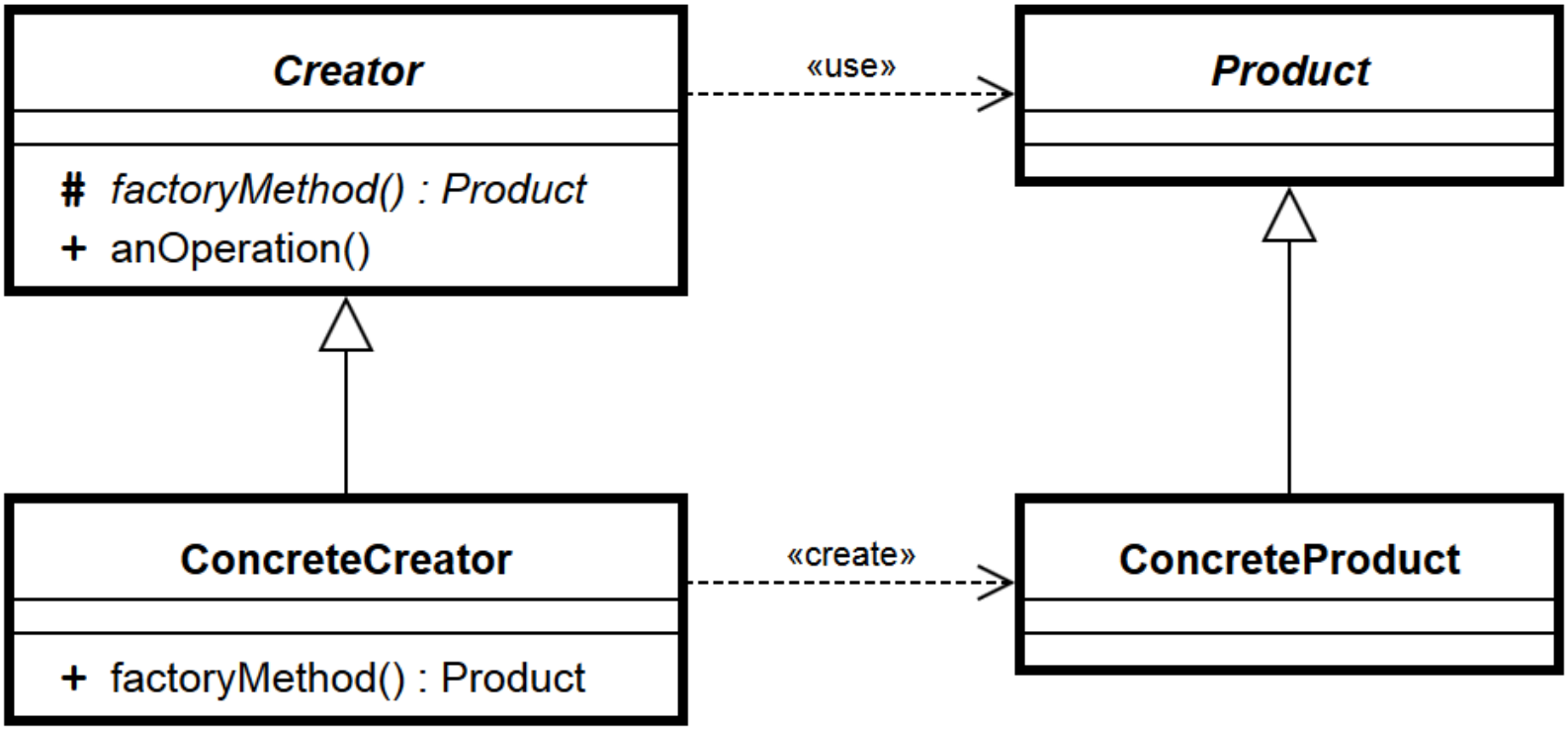
Command- struktura

Product – definiuje interfejs obiektów tworzony przez wzorzec Simple Factory

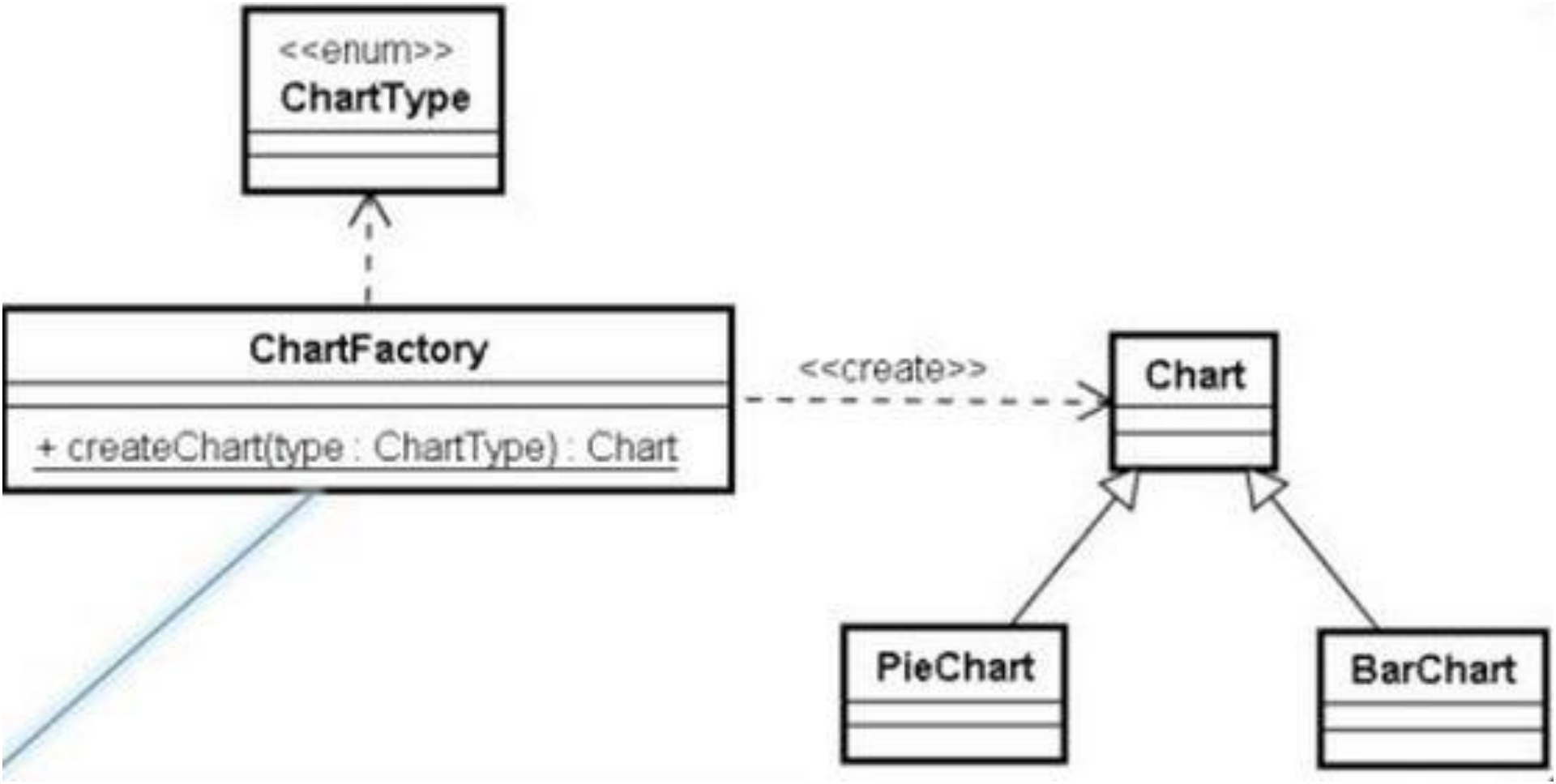
ConcreteProduct – konkretna implementacja produktu

Creator – definiuje interfejs do tworzenia obiektów typu *Product*

ConcreteCreator – tworzy obiekt *ConcreteProduct*



Przykładowy diagram klas



Przykładowa implementacja prostej fabryki

```
public class ChartFactory {

  public static Chart createChart( ChartType type ) {

    if ( ChartType.BAR.equals( type ) ) {
      return new BarChart();
    } else if ( ChartType.PIE.equals( type ) ) {
      return new PieChart();
        //...
```

```
public class ChartManager {

  public void drawChart() {

    Chart chart = ChartFactory.createChart( ChartType.BAR );
    //...
```

Ćwiczenia

- Korzystając ze wzorca SimpleFactory napisz program generujący klasy, które generują losowe strony HTMLowe.

- Strony to:
  - Galeria zdjęć (kod wrzucający obrazki <img>)
  - Strona informacyjna (dużo tekstu – może być losowego)
  - Strona kontaktowa (wygeneruj losowe dane teleadresowe)
  - Aktualności (podobna do informacyjnej, ale oddzielone sekcje z wiadomościami)