

Wzorce Projektowe, lab. 6

Temat: Wykorzystanie wzorców **Abstract Factory** oraz **Factory Method** do przeprowadzenia działań wojennych.

Prolog: Nie mam dobrych wieści. Wybuchła III Wojna Światowa, a nasze państwo zostało najechane przez wroga (nie mogę powiedzieć jakiego konkretnie, bo RODO). Zostałeś powołany przez rząd do napisania programu, który pozwoli łatwiej produkować zasoby wojenne i ich używać...

1. Chcemy używać trzech rodzajów zasobów wojennych: ludzi, broni oraz pojazdów. Każdy z tych zasobów może być innego typu – każdy człowiek, broń i pojazd może być ofensywny, defensywny lub przystosowany stricte do działań dywersyjnych. Należy więc na początku stworzyć typ enumeracyjny **TypZasobu**, który określa trzy wartości: **BRON**, **CZLOWIEK**, **POJAZD**.
2. Kolejny krok to opracowanie interfejsów dla naszych zasobów. Tworzymy trzy interfejsy: **ICzlowiek** z publiczną metodą **void walcz()**; **IBron** z publiczną metodą **void uzyj()**; oraz **IPojazd** z publiczną metodą **void ruszaj()**.
3. **ICzlowiek** musi być teraz zaimplementowany w trzech konkretnych klasach: **KomandosGROM** do działań ofensywnych, **ZolnierzWOT** (Wojsk Obrony Terytorialnej – przyp. red.) do defensywy oraz **Komunista** do przeprowadzenia rewolucji na terenie wroga w ramach dywersji i całkowitego zniszczenia jego gospodarki i ustroju państwowego.
4. **IBron** również musi być zaimplementowana w trzech konkretnych klasach:
 - ofensywa: **AK47**
 - defensywa: **TarczaAntyrakietowa**
 - dywersja: **ImpulsElektromagnetyczny**
5. Oczywiście ten sam los czeka **IPojazd**:
 - ofensywa: **MysliwiecF16**
 - defensywa: **Jelcz44232**
 - dywersja: **DronBezzalogowy**
6. Każda z klas z punktów 3,4 i 5 będzie musiała nadpisywać odpowiednie metody z interfejsów, które dziedziczą. Nadpisanie to ma polegać na wypisaniu w konsoli odpowiedniej wiadomości. Np. **Komunista** po wywołaniu metody **walcz()** może wypisać nam „Juz jutro u wroga zapanuje rownosc, a burzuzazja zawisnie na drzewach...”. Każda informacja powinna w sposób jednoznaczny określać kto jest jej autorem i na czym polega jego działanie. Dowództwo musi wiedzieć, co się dzieje!
7. Mamy już pełny wachlarz zasobów wojennych. Czas przejść do fabryk, które je wyprodukują. Stwórz najpierw interfejs **IFabryka** z metodą **public Object produkuj(TypZasobu typZasobu)**. Kolejny krok to 3 klasy implementujące ten interfejs: **FabrykaOfensywna**, **FabrykaDefensywna** oraz **FabrykaDywersyjna**. Metoda **produkuj()** powinna w zależności od dostarczonego typu zasobu zwrócić nową instancję konkretnej klasy, np. **FabrykaDywersyjna** dla typu **CZLOWIEK** powinna zwrócić obiekt klasy **Komunista**. Podsumowując, mamy teraz 3 typy fabryk, 3 typy zasobów wojennych i każda fabryka powinna dla każdego typu zasobu „produkować” obiekt innej klasy.
8. Dowództwo narzuciło też warunek, że każda fabryka może zostać utworzona tylko raz. Jako dobry programista domyślasz się na pewno, że musi ona zatem być **Singletonem**. Dowództwo jest przekonane, że wiesz, jak to zrobić, więc bez zagłębiania się w szczegóły: ma to być **Singleton** z późną inicjalizacją.

9. Mamy już gotową strukturę klas. Teraz przechodzimy do funkcji **main()**. Mamy w niej zadeklarować obiekt klasy **IFabryka** i zainicjalizować klasą **FabrykaDefensywna**, ponieważ zostaliśmy zaatakowani i najpierw musimy się bronić. Dowództwo nie ma zbyt dużej styczności z informatyką i dla pewności każe po zainicjalizowaniu obiektu zrobić to ponownie, jakby coś nie zadziałało... Nie musisz jednak o nic się martwić, bo **Singleton** nie pozwoli na drugie utworzenie tej samej fabryki. Następnie, mając **ArrayList<Object>** zaczynamy dodawać do niego konkretne produkty, które będzie tworzyła nasza fabryka. Dodajmy tam jakiegoś człowieka, pojazd i broń.
10. Po wypełnieniu listy przejdź po niej pętlą i w zależności od tego, czy to człowiek, pojazd czy broń, wywołaj odpowiednią dla niego metodę (pomocne może być tutaj **instanceof**).
11. Pod naszą zmienną z fabryką podstaw teraz obiekt klasy **FabrykaDywersyjna** (dwukrotnie) i znowu wyprodukuj kilka zasobów. Ponownie przejdź po niej pętlą i wywołaj odpowiednie metody.
12. Udało nam się odeprzeć atak wroga i podjąć działania dywersyjne. Teraz przechodzimy do ofensywy! Przesławiamy naszą fabrykę na działania ofensywne podstawiając pod zadeklarowaną wcześniej zmienną klasy **IFabryka** obiekt klasy **FabrykaOfensywna** zgodnie z wcześniejszym schematem: dwukrotne podstawienie, dodanie zasobów wojennych różnych typów i wywołanie dla nich metod.

Epilog: Działania ofensywne zakończyły się sukcesem. Skutecznie obroniliśmy się przed atakiem nieprzyjaciela, sprawnie przestawiliśmy nasze fabryki na produkcję zasobów dywersyjnych, a potem ofensywnych i przeprowadziliśmy świetnie zorganizowany kontratak. Polska stała się teraz światowym imperium... A potem się obudziliśmy...

UWAGI:

- Każda klasa, interfejs i typ enumeryczny **musi** być w oddzielnym pliku.
- Wynik działania funkcji **main()** ma dawać konkretne wyniki w postaci komunikatów w konsoli. Brak pewności co do tego, kto jest autorem komunikatu będzie prowadził do irytacji dowództwa i odesłania programu do poprawek, a czas na wojnie jest ograniczony...
- Pamiętaj, aby po każdym podstawieniu nowej fabryki czyścić listę! Mają się na niej znajdować tylko obiekty wyprodukowane przez aktualnie interesującą nas fabrykę.
- Ponieważ w funkcji **main()** tak naprawdę mamy zrobić to samo dla 3 typów fabryk, można pomyśleć nad utworzeniem jakiejś funkcji, żeby nie przekopiowywać tego samego kodu.