Technologia programowania - notatki na laboratoria

...czyli empiryczny dowód na to, jak długo trzeba nie spać, by wpaść na pomysł pisania czegoś takiego w LaTeXu

Mikołaj Pietrek

Lista 3

Zadanie 1.

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

- ReportBuilder

Rozwiązanie. Najprostsza metoda to zastosowanie leniwego wartościowania. W momencie wywołania następuje sprawdzenie, czy pole jest równe null, jeśli tak, to tworzy nową instancję. Kiedy jednak dwa wątki będą wywoływać równocześnie, drugi z nich może trafić na moment, gdy pierwszy rozpoczął już tworzenie, ale nie przypisał wartości. Nastąpi wtedy konflikt. Lepsza wersja wykorzystuje słowo kluczowe `volatile`, które zagwarantuje, że wszystkie odczyty i zapisy będą odbywać się w kolejności globalnej, przez co każdy wątek odczyta prawidłową wartość zmiennej zamiast podręcznej. Podwójne sprawdzenie zagwarantuje z kolei, że przed próbą utworzenia zmienna zostanie źablokowana".

Rozwiązanie (enum).

```
public enum mojSingleton {
  INSTANCE;
}
mojSingleton singleton = mojSingleton.INSTANCE;
```

Zadanie 2.

Definicja. Adapter - strukturalny wzorzec projektowy, którego celem jest umożliwienie współpracy dwóm klasom o niekompatybilnych interfejsach. Adapter przekształca interfejs jednej z klas na interfejs drugiej klasy. Innym zadaniem omawianego wzorca jest opakowanie istniejącego interfejsu w nowy.

- ThirdPartyDoorAdapter - wariant klasowy, dziedziczy po klasie adaptowanej, implementuje interfejs klienta.
- ThirdPartyObjectDoorAdapter - wariant obiektowy, dziedziczy interfejs klienta, zawiera klasę adaptowaną.
- ThirdPartyDoor - element adaptowany
- Door - element docelowy

Rozwiązanie. Oba adaptery utworzone na podstawie Door, w oparciu o SimpleDoor

Zadanie 3.

Definicja. Builder – kreacyjny wzorzec projektowy, rozdziela sposób 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. Budowniczy różni się od pozostałych wzorców kreacyjnych tym, że skupia się na sposobie tworzenia obiektów reprezentujących produkty. Klient otrzymuje produkt po zakończeniu jego pracy, a nie tak jak w przypadku Fabryki Abstrakcyjnej bezzwłocznie.

- ReportAssembler - nadzorca
- ****ReportBody - budowniczy
- ****ReportBodyBuilder - konkretny budowniczy

Rozwiązanie. Metody przeniesione do klas ****ReportBodyBuilder, proces tworzenia zgodnie z powyższym opisem, testy zmodyfikowane do zlecenia pracy właściwemu budowniczemu.

Zadanie 4.

Definicja. Dekorator – wzorzec projektowy należący do grupy wzorców strukturalnych. Pozwala na dodanie nowej funkcji do istniejących klas dynamicznie podczas działania programu. Polega na opakowaniu oryginalnej klasy w nową klasę "dekorującą". Zwykle przekazuje się oryginalny obiekt jako parametr konstruktora dekoratora, metody dekoratora wywołują metody oryginalnego obiektu i dodatkowo implementują nową funkcję.

- TheSocialChannel - Component
- LinkedinChannel, FacebookChannel, itd. - Concrete Components
- SocialChannelDecorator - Decorator
- MessageTruncator, itd. - Concrete Decorators

Rozwiązanie. Dodać WordCensor i odpowiednie testy, dodać testy łańcuchowe, sprawdzić pokrycie.

Zadanie 5.

Definicja. Composite - strukturalny wzorzec projektowy, którego celem jest składanie obiektów w taki sposób, aby klient widział wiele z nich jako pojedynczy obiekt.

- Shape - Component (klasa abstrakcyjna)
- LeafShape - Leaf (typ prosty, nie posiada potomków)
- CompositeShape - Composite (zawiera Leaf)

Rozwiązanie. Zgodnie z poleceniem, zostały zaimplementowane metody TODO, tak aby przechodziły testy.

Zadanie 6.

Definicja. State – czynnościowy wzorzec projektowy, który umożliwia zmianę zachowania obiektu poprzez zmianę jego stanu wewnętrznego. Innymi słowy uzależnia sposób działania obiektu od stanu w jakim się aktualnie znajduje.

- ***State - ConcreteStates (poszczególne stany)
- ApplianceStateBehavior - State
- Appliance - context

Rozwiązanie. Podobnie jak wyżej, polecenie dotyczyło zakomentowania i odkomentowania odpowiednich fragmentów kodu oraz zaimplementowania wskazanych metod.

Zadanie 7.

Definicja. Factory Method – kreacyjny wzorzec projektowy, którego celem jest dostarczenie interfejsu do tworzenia obiektów nieokreślonych jako powiązanych typów. Tworzeniem egzemplarzy zajmują się podklasy. Dwie ogólne klasy bądź interfejsy definiują pewien typ zasobów (Product) oraz sposób ich tworzenia (Creator, metoda factoryMethod). Od nich wyprowadza się konkretne klasy zasobów (ConcreteProduct) wraz z tworzącymi je klasami wytwórczymi (ConcreteCreator), które dostarczają odpowiednią implementację metody factoryMethod.

- RaportGenerator - Creator
- Report - Product
- ***ReportGenerators - Concrete Creators
- ***Reports - Concrete Products

Rozwiązanie. Utworzyć ***ReportGenerators oraz zmodyfikować ReportGenerator, aby sposób tworzenia raportów był zgodny ze wzorcem, następnie skorygować testy.

Zadanie 8.

Definicja. Abstract Factory – kreacyjny wzorzec projektowy, którego celem jest dostarczenie interfejsu do tworzenia różnych obiektów jednego typu (tej samej rodziny) bez specyfikowania ich konkretnych klas. Klasa Fabryka abstrakcyjna deklaruje abstrakcyjny interfejs umożliwiający tworzenie produktów. Interfejs ten jest implementowany w Fabrykach konkretnych, które odpowiedzialne są za tworzenie konkretnych produktów.

- AbstractReportElementsFactory - Abstract Factory
- ****ReportElementFactory - Factory
- Report - Client
- Report***** - Abstract Products
- ****Report**** - Products

Rozwiązanie. Dodać fabrykę abstrakcyjną i fabryki, zmodyfikować klienta, poprawić testy.

Zadanie 9.

Definicja. Fasada – wzorzec projektowy należący do grupy wzorców strukturalnych. Służy do ujednolicenia dostępu do złożonego systemu poprzez wystawienie uproszczonego, uporządkowanego interfejsu programistycznego, który ułatwia jego użycie.

- DefaultBookstoreFacade - Facade (ma referencje do elementów z metodami wykonującymi zadania)
- interfejsy w pakiecie 'service' - system złożony

Rozwiązanie. Wykonać polecenia w klasie testowej, utworzyć fasadę i referencje do metod, przetestować.

Zadanie 10.

Definicja. Flyweight – strukturalny wzorzec projektowy, którego celem jest zmniejszenie wykorzystania pamięci poprzez poprawę efektywności obsługi dużych obiektów zbudowanych z wielu mniejszych elementów poprzez współdzielenie wspólnych małych elementów.

- WeatherStationController - Flyweight
- WeatherStationControllerFactory - Flyweight Factory

Rozwiązanie. Grupowanie wspólnych wartości przez Flyweight Factory zostało już wykonane przez autora zadania, dlatego zgodnie z poleceniem wystarczy odkomentować i zmierzyć czas. W IntelliJ dodatek JMeter wymaga jedynie wskazania lokalizacji katalogu, a jconsole jest dostępne domyślnie w jdk. Testy wykazały około 1/3 mniejsze zużycie pamięci. Wynik jest niestety mało dokładny, ponieważ komputer (z CPU Intel i5 2x1.7Ghz) nie odświeżał danych wystarczająco szybko nawet przy zalecanej przez dr-a Lemiesza 10x mniejszej ilości wątków.