

Wzorce Projektowe, lab. 2

1. Zaimplementować klasę **Sklep**, który jest singletonem, zgodnie z kodem ze slajdu 2 pliku singleton.pdf (wczesna inicjalizacja, już podczas deklaracji). Klasa **Sklep** publiczna w pliku oddzielnym od publicznej klasy programu. **toString()** **Sklepu** powinien zwracać nazwę **Sklepu**.

W **main** wywołać statyczną metodę **Sklep.getInstance()**, żeby uzyskać dostęp do unikalnego obiektu **Sklepu**, przechowywać go dla wygody w lokalnej referencji na **Sklep** o nazwie **sk**. Ustawić wartość zmiennej **nazwa** dla **sk**. Wypisać **Sklep** przy pomocy **System.out.println()**.

Sklep
- Sklep instance + String nazwa
- Sklep() + getInstance: Sklep + toString(): String

2. Zmodyfikować poprzedni projekt tak, żeby inicjalizacja była późna (slajd 4, w kodzie błąd w drugiej linijce, instance nie powinna być teraz final, bo trzeba do niej pisać). Dopisać w konstruktorze bezparametrowym komunikat diagnostyczny, np. „Utworzono sklep.”. Z **maina** wywołać **getInstance()** dwa razy, żeby zaobserwować komunikat z konstruktora za pierwszym razem, brak komunikatu za drugim.

3. Do projektu dopisujemy implementację wzorca **State**, zgodnie ze schematem ze slajdu 2 plik state.pdf.

Do pliku Sklep.java dopisujemy interfejs **StanSklepu** z metodą **kupuj()**. Interfejs ten implementujemy w trzech klasach **StanOtwarty**, **StanZamknięty**, **StanRemanent**. Treści metody **kupuj()** to odpowiedni komunikat diagnostyczny typu np.: „Nie można kupić, bo trwa remanent.”. Do klasy **Sklep** dopisujemy zmienną klasową **stan** typu **StanSklepu**, inicjujemy na nowoutworzony obiekt klasy **StanOtwarty**. Dopisujemy do **Sklepu** metodę **kupuj()**, która wywołuje metodę **kupuj()** dla referencji **stan** (przekazuje żądanie do swojego stanu). Z **main** wywołujemy metodę **kupuj()** dla **Sklepu**, żeby zaobserwować komunikat z aktualnego stanu.

Sklep
- Sklep instance + String nazwa - StanSklepu stan
- Sklep() + getInstance: Sklep + toString(): String + kupuj()

4. Do **Sklepu** dopisujemy publiczne metody zmieniające **stan**: **zamknij()**, **otwórz()**, **remanentuj()**. W ich treści ustawiamy **stan** na nowoutworzony obiekt odpowiedniego stanu sklepu. W **main** wywołujemy po kolei te metody, po każdej wywołując **kupuj()** dla sklepu **sk**, żeby zaobserwować komunikaty dla różnych stanów.

Sklep
- Sklep instance + String nazwa - StanSklepu stan
- Sklep() + getInstance: Sklep + toString(): String + kupuj() + zamknij() + otwórz() + remanentuj()

UWAGI:

- publiczna klasa musi być zadeklarowana w osobnym pliku o tej samej nazwie co klasa, z rozszerzeniem .java. Projekty Netbeansa nie lubią ręcznych zmian nazwy klasy publicznej.
- interfejs tylko deklaruje metody (nie definiuje ich)
- implementacja interfejsu przez klasę wymusza implementację metod w nim zdefiniowanych
- nie dopisujemy żadnych getterów/seterów/konstruktorów poza tym, co jest jawnie zapisane w treści
- literówki w nazwach kompilator traktuje bardzo poważnie