

Wzorce Projektowe, lab. 10

Temat: Praktyczne zastosowanie wzorca **Adapter**.

Otrzymujesz bazę danych, z której możesz pobierać informacje za pomocą odgórnie narzuconego interfejsu. Nie jest on jednak zgodny z interfejsem i sposobem reprezentacji danych, który chcesz dostarczyć klientowi. Musisz stworzyć klasę adaptującą bazę danych do Twoich potrzeb.

1. Na początku utwórz klasę **Database** z następującymi metodami:

`public static float GetCriticalPoint(String compound, String point)`: przyjmuje na wejściu nazwę związku chemicznego oraz symbol punktu krytycznego: „M” – melting, „B” – boiling, „F” – flash. Następnie w zależności od punktu i nazwy związku chemicznego zwraca odpowiednią wartość. Dla wszystkich przypadków, których nie ma w bazie danych ma zwracać `Float.MIN_VALUE`.

`public static String GetMolecularStructure(String compound)`: przyjmuje na wejściu nazwę związku chemicznego i zwraca jego wzór chemiczny. Dla nieznanego związku zwraca „Unknown”.

`public static double GetMolecularWeight(String compound)`: przyjmuje na wejściu nazwę związku chemicznego i zwraca jego masę molową. Dla nieznanego związku zwraca „Unknown”.

2. Uzupełnij utworzoną wyżej bazę danych o 4 wybrane związki chemiczne (typu: woda, benzyna). Odpowiednie informacje na ich temat znajdź w internecie.
3. Stwórz interfejs **IAdapter** z metodami `public String stopienLatwopalnosci()` i `public void wyswietl()`.
4. Utwórz klasę adaptującą o nazwie **ZwiazekChemiczny** dziedziczącą z **Database** i implementującą **IAdapter**. Ma ona mieć prywatne pola odpowiadające za nazwę związku, jego temperaturę topnienia, temperaturę wrzenia, temperaturę zapłonu, masę molową i wzór chemiczny. Konstruktor ma przyjmować w parametrze nazwę związku, a reszta informacji ma być zaczerpnięta poprzez metody z **Database**. Pamiętaj o obsłudze przypadków, gdy ktoś dostarczył nazwę pisaną dużymi literami, a w Twojej bazie danych będzie to samo napisane małymi literami itp.
5. Implementacja metody `stopienLatwopalnosci()` z interfejsu **IAdapter** w klasie **ZwiazekChemiczny** ma polegać na zwracaniu odpowiedniej informacji:
 - a) Produkt nie jest palny
 - b) Produkt jest palny: „Substancją palną nazywamy substancję, która łatwo wchodzi w reakcję spalania a jej temperatura zapłonu jest niższa niż 90°C.” (źródło: safetruck.net-mex.hu)
 - c) Produkt jest łatwopalny: „Substancje i preparaty (...) o temperaturze zapłonu od 21 °C do 55 °C” (źródło: Wikipedia)
 - d) Produkt jest wysoce łatwopalny: „Substancje i preparaty (...) o temperaturze zapłonu poniżej 21 °C, które nie są skrajnie łatwopalne.” (źródło: Wikipedia)
 - e) Produkt jest skrajnie łatwopalny: „Substancje i preparaty (...) o temperaturze zapłonu poniżej 0 °C oraz temperaturze wrzenia (...) niższej lub równej 35 °C” (źródło: Wikipedia)

Pamiętaj, że stosujemy tu pewne uproszczenia: nie uwzględniamy stanu skupienia związku i kilku innych czynników. Dla potrzeb naszego zadania nie musimy jednak aż tak zagłębiać się w szczegóły.

6. Metoda `wyswietl()` w klasie **ZwiazekChemiczny** ma drukować pełną, przechowywaną w klasie informację o związku chemicznym wraz z jego stopniem łatwopalności (wykorzystaj metodę z pkt. 5).
7. W metodzie `main()` utwórz pętlę, która będzie pobierała od użytkownika w konsoli nazwę związku chemicznego, tworzyła na jej podstawie obiekt klasy **ZwiazekChemiczny** i wywoływała dla niego metodę `wyswietl()`. Wyjście z pętli ma następować po wciśnięciu odpowiedniego przycisku.

UWAGI:

- Wszystkie klasy i interfejsy muszą być w oddzielnych plikach.
- Związki chemiczne w Twojej bazie danych mają być unikalne w skali grupy laboratoryjnej. Powtarzać się może jedynie woda.