

Pamięciowa baza danych – projekt

Cel projektu

Aplikacja desktopowa w języku Java do przechowywania w pamięci informacji na temat pewnego rodzaju obiektów (wg własnego pomysłu) z interfejsem do wyboru:

- tekstowym (konsolowa) – do zaliczenia przedmiotu wymagane również wykonanie wszystkich zestawów ćwiczeń
- graficznym (aplikacja okienkowa z użyciem biblioteki JavaFx) – wybór tej opcji zwalnia z obowiązku wykonania zestawów ćwiczeń.

Wymagania szczegółowe

- Dziedzina przedmiotowa:

1. Hierarchia klas dla wybranej przez siebie dziedziny składająca się z minimum 7 klas (wliczając korzeń i węzły pośrednie).
2. Klasa reprezentująca najbardziej ogólny rodzaj obiektów powinna zawierać co najmniej 3 pola. Jednym z pól powinien być numeryczny *identyfikator*, a drugim tekstowa *nazwa*.
3. Każda podklasa w hierarchii powinna rozszerzać nadklasę o co najmniej jedno pole.
4. Wszystkie pola we wszystkich klasach powinny być prywatne. Należy do ich obsługi zdefiniować publiczne właściwości.
5. Każda klasa powinna posiadać jeden publiczny konstruktor ustawiający wszystkie właściwości tworzonego obiektu zgodnie z przekazanym kompletem parametrów. Klasy nie powinny posiadać konstruktora bezargumentowego.
6. Hierarchia klas powinna zawierać klasy abstrakcyjne. Przynajmniej jedna z klas abstrakcyjnych powinna zawierać przynajmniej jedną abstrakcyjną metodę.
7. Interfejs pozwalający na wykonanie pewnej specyficznej operacji, implementowany przez niektóre klasy z hierarchii.
8. Przynajmniej jedna metoda statyczna i jedno statyczne pole.
9. Typ enum wykorzystany jako typ pola w klasie (lub klasach).
10. Klasy, enum i interfejs powinny znajdować się w nazwanym pakiecie (a nie w domyślnym).

- Model danych:

Hierarchia klas (wraz z interfejsem) powinna zostać przedstawiona jako diagram klas UML

- Przechowywanie danych:

Dane (instancje klas z zaprojektowanej przez siebie hierarchii) powinny być przechowywane w pamięci programu w strukturze listy. (Jedna wspólna lista dla wszystkich podtypów.)

- Funkcjonalność aplikacji:

1. Dodawanie nowego obiektu (zestaw pól do wypełnienia zależny od wybranego podtypu).
2. Usuwanie obiektu o podanym identyfikatorze.
3. Wyświetlanie wszystkich zapamiętanych obiektów (komplet informacji o każdym z nich)
4. Wyświetlanie obiektów danego podtypu.
5. Wyświetlanie obiektów, których nazwa zawiera podany ciąg znaków.
6. Wykonanie operacji zawartej w interfejsie na wszystkich obiektach klas implementujących ten interfejs.
7. Walidacja wprowadzanych danych liczbowych poprzez obsługę wyjątku przy konwersji.

- Zasady oceniania: -0,5 oceny za każde 2 usterki / braki, -1,0 za brak diagramu UML