

Ćwiczenie 3.

Celem ćwiczenia jest przybliżenie technik obiektowego programowania na przykładzie wybranego problemu. Rozwiązanie problemu powinno mieć postać aplikacji:

- działającej w trybie tekstowym,
- pobierającej argumenty z linii wywołania lub z klawiatury (np. dzięki wykorzystaniu klasy EasyReader),
- wyświetlającej wyniki na ekran.

Etap projektowania aplikacji powinien uwzględniać budowę statycznego diagramu klas w standardzie UML i automatyczne generowanie kodu Javy. Należy rozwiązać przynajmniej jeden z poniższych problemów.

Problem 1

Należy zliczyć częstość wystąpień słów występujących w podanych plikach.

Jeśli w plik1 słowo „Ala” występuje 1 raz, w plik2 – 1 raz, w plik3 – 4 razy, w plik4 – 3razy, to częstość wystąpień tego słowa da się przedstawić jako wiersz w tabeli jak niżej:

słowo	częstość wystąpień			
	plik1	plik2	plik3	plik4
Ala	1	1	4	3
Ela	3	2	2	2
Ola	1	4	0	1

Tabela taka powinna być wynikiem działania programu. Ilość wierszy w tabeli zależy od ilości znalezionych słów. Liczba kolumn w tabeli zależy od liczby analizowanych plików. Nazwy plików powinny być przekazane do programu w linii poleceń (albo wprowadzone z klawiatury).

Implementacja algorytmu zliczania słów powinna opierać się o strukturę danych będącą listą liter powiązaną z listą słów. Powiązanie list powinno odbywać się na zasadzie: do każdego elementu listy liter dołączona jest lista słów, zaczynających się od literą przechowywanej w elementach listy liter. Elementów na liście liter może być, co najwyżej, 26 (zakładamy brak polskich znaków). Elementów na liście słów może być wiele - każdy element powinien pamiętać krotność swoich wystąpień w kolejnych plikach. Lista słów powinna być posortowana alfabetycznie.

Do implementacji list można wykorzystać klasy standardowych kolekcji Javy lub też można utworzyć własne klasy,

Problem 2

Napisz program, w którym zamodelowane zostanie funkcjonowanie magazynu sklepowego. Magazyn powinien przechowywać informacje o posiadanych zapasach artykułów, oraz:

- umożliwiać modyfikację stanu magazynu metodami `dodajArtykul`, `pobierzArtykul`
- umożliwiać przeglądanie stanu magazynu metodą `znajdzArtykul`, której parametrem może być nazwa artykułu lub ciąg znaków zawierający znaki „?” (dowolna litera) i „*” (dowolny ciąg liter)
- umożliwiać dokonywanie wyceny zamówienia zawierającego listę artykułów metodą `kosztZamowienia` (przy czym lista zawiera nazwy artykułów z ich ilością)

Artykuł powinien zawierać kod artykułu, nazwę artykułu, cenę artykułu.

Interfejs użytkownika programu powinien symulować działanie punktu przyjmowania zamówień hurtowni. Tj. użytkownik powinien mieć możliwość: zapytania, czy jakiś artykuł jest na stanie magazynu; zestawienia swojego zamówienia, dokonania wyceny zamówienia.