

Przewodnik 9 – operacje wyjścia wejścia dla plików

dr inż. Łukasz Sosnowski
Akademia WIT
pod auspicjami Polskiej Akademii Nauk

1 Import pliku

Zaimportuj udostępniony plik tutorial9.txt do katalogu src/test/resources

2 Dodanie klasy

- a. W pakiecie pl.wit.lab4 w katalogu src/main/java utwórz klasę DocHistogram.
- b. Zdefiniuj w klasie zmienną składową prywatną typu String przechowującą ścieżkę do pliku podawaną w konstruktorze. Utwórz stosowny konstruktor publiczny. W konstruktorze przypisz przekazaną wartość ścieżki do zmiennej składowej klasy.
- c. Dodaj prywatną zmienną składową klasy Map<String,Integer> mapHistogram oraz zainicjuj ją w konstruktorze.
- d. Dodaj metodę prywatną nie zwracającą wartości o nazwie parseFile();
Zadaniem tej metody jest odczytanie zawartości pliku przekazanego w konstruktorze (ścieżki) z użyciem strumienia znakowego sprzężonego ze strumieniem buforowanym (patrz wykład!). Zakładając, że tekst nie zawiera przenoszenia wyrazów, zaimplementuj dla każdej odczytanej linii tekstu:
 - podział linii tekstu na wyrazy „oczyszczone” ze znaków typu przecinek i kropka oraz znaków końca linii. Można użyć metody split() z klasy String.
 - nową metodę prywatną liczącą wystąpienia wyrazów i dodającą do już zapisanego wcześniej stanu zliczania (w zmiennej mapHistogram). Przy okazji liczenia zamieniaj wszystkie wyrazy na pisane małymi literami.

Przykład:

Jeśli plik zawierałby następujące przykładowe 3 linie tekstu:

Ala ma kota
ale Ala nie ma psa
za to psa ma Ela

To wtedy przetwarzanie pierwszej linii powinno zasilić mapę następującymi elementami:

Ala,1

ma,1

kota,1

przetwarzanie drugiej linii powinno pozostawić w zmiennej mapy następujące elementy:

Ala,2

ma,2

kota,1

ale,1

nie,1

psa,1

po zakończeniu przetwarzania całościowo mapa powinna zawierać:

Ala,2

ma,3

kota,1

ale,1

nie,1

psa,2

za,1

to,1

Ela,1

Kolejność elementów nie ma znaczenia!!!

Po implementacji metody, dodaj getter dla zmiennej mapHistogram; Do odczytania zawartości pliku użyć try-catch dla zasobów (patrz wykład!).

Metodę parseFile() wywołaj w konstruktorze.

e. Dodaj metodę publiczną String printHistogram() i zaimplementuj ją z użyciem StringBuildera oraz klasy Collections w taki sposób aby zwracała w postaci String'a zawartość histogramu posortowanego po wyrazach alfabetycznie. Czyli dla wyniku z przykładu powyżej powinno być zwrócone:

Ala,2

ale,1

Ela,1

kota,1

ma,3

nie,1

psa,2

to,1

za,1

f. Dodaj metodę publiczną `saveHistogramToFile()` przyjmującą w parametrze ścieżkę do pliku oraz zaimplementuj ją w taki sposób aby zawartość zwróconą przez metodę `printHistogram` metoda zapisała do pliku z użyciem buforowanego strumienia znakowego.

3 Dodanie testu jednostkowego `DocHistogramTest`

Zdefiniuj test jednostkowy sprawdzający poprawność działania poszczególnych metod, z użyciem pliku przekazanego do tego zadania.

Który wyraz jest najczęściej występującym w tym tekście?