

Problema 2: Especificació i implementació de la classe **WordCounter**

Esteve Brugulat

Josep M. Ribó

26 de setembre de 2009

1 Objectius

- Usar els apuntadors i la memòria dinàmica
- Comprendre què és una classe i les passes de definició de la mateixa (especificació-representació-implementació)

Per fer aquest problema, cal que us llegiu:

- L'annex A dels apunts del primer capítol (Apuntadors i referències)
- Els apartat 1.6 dels apunts

A cada apartat del problema trobareu entre parèntesi i en negreta els apartats dels apunts on s'explica la teoria que necessiteu per resoldre'l.

2 Especificació de la classe **WordCounter**

Els objectes de la classe **WordCounter** emmagatzemaran una col·lecció de paraules i mantindran un comptador de freqüències de les mateixes.

Així doncs, si introduïm a un objecte **wc** de la classe **WordCounter** les paraules:

pepet, joanet, anneta, anneta, pepet, carles, pius, pius, joanet, anneta, pepet

Després voldrem:

- Saber quants cops hem introduït la paraula *anneta* a l'objecte **wc**? (3 cops)
- Saber quantes paraules diferents hi ha a emmagatzemades a **wc**? (5 paraules diferents)
- Obtenir totes les paraules emmagatzemades a **wc** i la seva freqüència (*pepet, 3; joanet, 2; anneta, 3; carles, 1; pius, 2*).

De tota manera, això darrer no resulta tan senzill. Per fer-ho una mica més simple us proposo que la classe ens ofereixi una operació (anomenem-la **getWord(...)**) que ens retorni la paraula i-èsima de les que estan emmagatzemades a **wc**.

Exemple: La paraula 0 podria ser *pepet*; la paraula 1, *joanet*; la 2 *anneta*; la 3, *carles*.

Quan haguem cridat **getWord(...)** per cada *i* des de 0 fins al nombre de paraules diferents emmagatzemades a **wc** haurem hagut d'obtenir totes les paraules diferents emmagatzemades a **wc**.

Recordeu que, a més a més de tot això, necessitarem una operació per inicialitzar convenientment l'objecte `wc` (l'equivalent a `createDate` de la classe `Date`) i una altra per anar-hi afegint paraules!!!

De moment, podem adoptar la simplificació que un objecte de la classe `WordCounter` no contindrà mai més de `MAXDIFWORDS` paraules diferents.

1. Quines operacions et sembla que haurà d'oferir la classe `WordCounter`? Amb quins paràmetres? Escriu-les a la part pública del fitxer `WordCounter.h`
Per respondre adequadament a aquesta pregunta, us en podeu fer una altra: *Què li demanaran a l'objecte `wc` de la classe `WordCounter` els seus clients?*
2. Especifica cadascuna d'aquestes operacions al fitxer `WordCounter.txt`. Tingues en compte les situacions d'error que es poden produir.

3 Representació de la classe `WordCounter`

1. Ara cal representar la classe `WordCounter`. Per això haureu d'escriure la part privada del fitxer `WordCounter.h`.

Aquest és un punt molt important del problema. La representació de `WordCounter` és més complicada que al de `Date`, així que us vindrà bé dibuixar aquesta representació en un paper per ajudar-vos a triar-la convenientment.

Els aspectes que has de considerar a l'hora de triar aquesta representació són:

- Què he d'emmagatzemar en un objecte de la classe `WordCounter` (una col·lecció de paraules).
- De quina manera es pot emmagatzemar la informació per tal que després es puguin implementar eficientment les operacions que he especificat a la secció anterior?

Us proposo que, a l'hora de triar aquesta representació useu arrays d'apuntadors.

4 Implementació de les operacions de la classe `WordCounter`

1. Ara, per acabar la definició de la classe `WordCounter` cal implementar les operacions de la classe de manera que:
 - Satisfacin l'especificació donada a la secció 2
 - La implementació usi els atributs de la representació de la classe donats a la secció 3.

5 Ús de la classe `WordCounter`

1. Ara es tracta de dissenyar un programa, que col·loquem al fitxer `WordCounterUser.cpp` que sigui *client* de la classe `WordCounter`. Voldrem que aquest programa faci el següent:
 - Crei un objecte `wc` de la classe `WordCounter`.
 - Afegixi a `wc` tota una sèrie de paraules que es poden llegir de l'entrada estàndar (teclat, usualment). La sèrie de paraules acabarà amb la paraula `fi`.
 - Indiqui per la sortida estàndar els cops que apareix a la sèrie de paraules que conté `wc` una paraula que l'usuari entrarà per l'entrada estàndar.
 - Finalment, tregui per la sortida estàndar la llista de paraules diferents emmagatzemades a `wc` juntament amb la freqüència de cadascuna.

6 Feines opcionals

Com a feines opcionals us proposo les següents:

1. Treure la restricció del nombre màxim de paraules diferents (**MAXDIFWORDS**). Ara, quan inicialitzem **wc**, podrem indicar el nombre màxim de paraules diferents que podrà contenir.
2. Substituir l'operació inicialitzadora d'un objecte de la classe per una operació **CONSTRUCTORA**. Quines ventatges té això???? (**VEGEU APUNTS APARTAT 1.7**)
3. I, ja que hi som, per què no hi afegeixes també una operació destructora? (**VEGEU APUNTS APARTAT 1.8**)

7 Projecte Netbeans

Aquest projecte us el deixem com a projecte Netbeans, que podeu obrir amb aquesta aplicació i completar convenientment.