Laboratorio de PRO2. Ejercicio Factor PSI

version oct-2013

Generado por Doxygen 1.8.2

Miércoles, 18 de Marzo de 2015 18:29:42

Índice general

Capítulo 1

Ejemplo de diseño modular: Factor Psi.

En este ejemplo se construye un programa modular que, dado un texto marcado, obtiene la frecuencia de la palabra más frecuente del mismo. Se introducen las clases *Palabra* y *ListaPalabras*.

Capítulo 2

Índice de clases

2.1. Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

Capítulo 3

Indice de archivos

3.1. Lista de archivos

Lista de todos los archivos con descripciones breves:

Capítulo 4

Documentación de las clases

4.1. Referencia de la Clase ListaPalabras

Representa una colección de palabras distintas, cada una con un entero mayor que 0 asociado.

Métodos públicos

- ListaPalabras ()

  Creadora por defecto.
- void anadir_palabra (const Palabra &p)

  Añade una palabra a la lista.
- int longitud () const

  Consultora de la longitud.
- int max_frec () const

  Consultora de la frecuencia de la palabra más frecuente.
- void escribir () const

  Operación de escritura.

Métodos públicos estáticos

- static int longitud_maxima ()

  Consultora de la longitud máxima.

4.1.1. Descripción detallada

Representa una colección de palabras distintas, cada una con un entero mayor que 0 asociado.

El entero representa el número de veces que la palabra se ha añadido a la lista, lo consideramos la *frecuencia* de dicha palabra

Definición en la línea 18 del archivo ListaPalabras.hh.

4.1.2. Documentación del constructor y destructor

4.1.2.1. ListaPalabras::ListaPalabras ( )

Creadora por defecto.

Se ejecuta automáticamente al declarar una lista.

Precondición

cierto

Postcondición

El resultado es una lista vacía

4.1.3. Documentación de las funciones miembro

4.1.3.1. void ListaPalabras::anadir_palabra (const Palabra & *p*)

Añade una palabra a la lista.

Precondición

La longitud del parámetro implícito es menor que la longitud máxima o *p* ya está en él

Postcondición

Si *p* está en el p.i. original, su frecuencia queda incrementada en 1; si no, p pasa a estar en el p.i., con frecuencia 1

4.1.3.2. static int ListaPalabras::longitud_maxima ( ) [static]

Consultora de la longitud máxima.

Precondición

cierto

Postcondición

El resultado es la longitud máxima de una lista permitida por la implementación

4.1.3.3. int ListaPalabras::longitud ( ) const

Consultora de la longitud.

Precondición

cierto

Postcondición

El resultado es la longitud del parámetro implícito

4.1.3.4. int ListaPalabras::max_frec ( ) const

Consultora de la frecuencia de la palabra más frecuente.

Precondición

cierto

Postcondición

El resultado es la frecuencia de la palabra más frecuente del parámetro implícito

4.1.3.5. void ListaPalabras::escribir ( ) const

Operación de escritura.

Precondición

cierto

Postcondición

Por el canal estándar de salida se ha escrito cada palabra del parámetro implícito y su frecuencia (separadas por un espacio en blanco, un par palabra-frecuencia en cada línea), en el orden en que cada palabra ha sido añadida por primera vez al p. i.

La documentación para esta clase fue generada a partir del siguiente fichero:

- ListaPalabras.hh

4.2. Referencia de la Clase Palabra

Representa una lista indexada y acotada de caracteres alfanuméricos.

Métodos públicos

- Palabra ()

  Creadora por defecto.
- void anadir_letra (char c)

  Añade un carácter nuevo al final de una palabra.
- int long_pal () const

  Consultora de la longitud.
- char consultar_letra (int i) const

  Consultora del carácter i-ésimo.
- bool iguales (const Palabra &p) const

  Igualdad de palabras.
- void leer (char x)

  Operación de lectura.
- void escribir () const

  Operación de escritura.

Métodos públicos estáticos

- static int longitud_maxima ()

  Consultora de la longitud máxima.
- static bool es_separador (char c)

  Comprobación de caracter separador.

4.2.1. Descripción detallada

Representa una lista indexada y acotada de caracteres alfanuméricos.

Los caracteres válidos son 'a'..'z', 'A'..'Z' y '0'..'9'. El resto son considerados separadores de cara a la lectura por el canal standard.

Las operaciones de lectura requieren un parámetro que se usará para distinguir un separador especial, que sirva por ejemplo para marcar el final de un texto. Dicho separador especial puede ser cualquier caracter ASCII del rango 0-127, "visible" y distinto de "a".."z", "A".."Z" y "0".."9", como "$", "." o "+" (pero no el blanco, el salto de línea o el tabulador, por no ser visibles, o "ç", "ñ", etc., por no ser del rango 0-127).

Definición en la línea 25 del archivo Palabra.hh.

4.2.2. Documentación del constructor y destructor

4.2.2.1. Palabra::Palabra ( )

Creadora por defecto.

Se ejecuta automáticamente al declarar una palabra.

Precondición

cierto

Postcondición

El resultado es una palabra sin caracteres y longitud 0

4.2.3. Documentación de las funciones miembro

4.2.3.1. void Palabra::anadir_letra (char *c*)

Añade un carácter nuevo al final de una palabra.

Precondición

c no es un separador; la longitud del parametro implícito es menor que la máxima permitida

Postcondición

el parametro implícito queda como el original pero con *c* añadido al final

4.2.3.2. static int Palabra::longitud_maxima ( ) [static]

Consultora de la longitud máxima.

Precondición

cierto

Postcondición

El resultado es la longitud máxima de una palabra permitida por la implementación

4.2.3.3. int Palabra::long_pal ( ) const

Consultora de la longitud.

Precondición

cierto

Postcondición

El resultado es la longitud del parámetro implícito

4.2.3.4. static bool Palabra::es_separador (char *c*) [static]

Comprobación de caracter separador.

Precondición

cierto

Postcondición

El resultado indica si c es un separador para la clase (c no puede formar parte de una palabra)

4.2.3.5. char Palabra::consultar_letra (int *i*) const

Consultora del carácter i-ésimo.

Precondición

1<=*i*<=longitud del parámetro implícito

Postcondición

El resultado es el carácter *i*-ésimo del parámetro implícito

4.2.3.6. bool Palabra::iguales (const Palabra & *p*) const

Igualdad de palabras.

Precondición

cierto

Postcondición

El resultado indica si *p* es igual al parámetro implícito

4.2.3.7. void Palabra::leer (char *x*)

Operación de lectura.

Precondición

En el canal standard de entrada hay uno o más caracteres 'a'..'z', 'A'..'Z', '0'..'9' o x; x es un separador visible del rango (0-127)

Postcondición

El parámetro implícito contiene el primer grupo de caracteres válidos leídos del canal standard de entrada, hasta el primer separador posterior a éstos; si antes del primer carácter válido aparece "x", se obtiene una palabra de longitud cero.

4.2.3.8. void Palabra::escribir ( ) const

Operación de escritura.

Precondición

cierto

Postcondición

Se han escrito los caracteres del parámetro implícito en el canal standard de salida. Si está vacío no se escribe nada.

La documentación para esta clase fue generada a partir del siguiente fichero:

- Palabra.hh

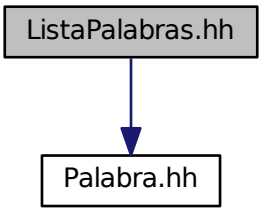
Capítulo 5

Documentación de archivos

5.1. Referencia del Archivo ListaPalabras.hh

Especificación de la clase ListaPalabras.

Dependencia gráfica adjunta para ListaPalabras.hh:



Clases

- class ListaPalabras

  Representa una colección de palabras distintas, cada una con un entero mayor que 0 asociado.

5.1.1. Descripción detallada

Especificación de la clase ListaPalabras.

Definición en el archivo ListaPalabras.hh.

5.2. Referencia del Archivo Palabra.hh

Especificación de la clase Palabra.

Clases

- class Palabra

  Representa una lista indexada y acotada de caracteres alfanuméricos.

5.2.1. Descripción detallada

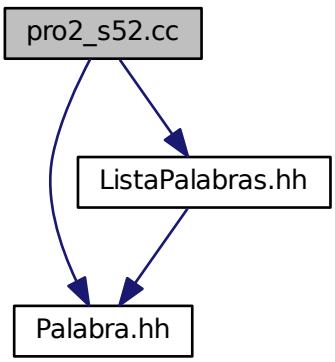
Especificación de la clase Palabra.

Definición en el archivo Palabra.hh.

5.3. Referencia del Archivo pro2_s52.cc

Programa principal para el ejercicio *Factor Psi*.

Dependencia gráfica adjunta para pro2_s52.cc:



Funciones

- int main ()

  Programa principal para el ejercicio Factor Psi.

5.3.1. Descripción detallada

Programa principal para el ejercicio *Factor Psi*.

Definición en el archivo pro2_s52.cc.

5.3.2. Documentación de las funciones

5.3.2.1. int main ( )

Programa principal para el ejercicio *Factor Psi*.

Definición en la línea 19 del archivo pro2_s52.cc.

```
{

}
```

Índice alfabético