

Estudio.pdf

1. Especificación, definición y conjunto de operaciones básicas.

T.D.A Frase:

Especificación: se trata de una agrupación de traducciones a un idioma "destino" de una frase en un idioma "origen". Es decir, dada una frase en un idioma (PEj: inglés), el T.D.A le asocia una o más frases en otro idioma (PEje: español).

Operaciones básicas:

- Constructor por defecto (frase vacía)
- Consulta: permite acceder a las traducciones de la frase origen y a la frase origen.
- insercción de traducciones.
- salida de la frase por un flujo de salida.
- operación de asignación

T.D.A Conjunto de Frases:

Especificación: se trata de una agrupación (del tipo conjunto) de objetos del T.D.A Frase. Una frase en un idioma fuente no aparece más de una vez (ni siquiera con diferentes traducciones).

Operaciones básicas:

- Consultar tamaño del conjunto (número de frases disponibles)
- Consulta: permite acceder a la información contenida en cada Frase del conjunto.
- Búsqueda: comprueba si una frase (en el idioma de origen) está en el conjunto y en qué posición se encuentra.
- Insercción de frases en el conjunto.
- Escritura y Lectura de Conjuntos de Frases a través de un flujo de entrada/salida, por ejemplo para leer un Conjunto de Frases desde un archivo (o escribir uno en un archivo).

nota: respecto al constructor, no tiene sentido pues la clase VECTOR-DINÁMICO tiene su propio constructor por defecto y, por lo general, el uso adecuado del T.D.A conjunto de frases sería el de leer un conjunto de frases desde un fichero de entrada.

2 y 3. Estructura de datos para tipo rep.

T.D.A Frase:

Posibles:

```
# Tipo rep 1: Con un vector dinámico (que habría que implementar)
class Frase{
private:
    String origen ;
    VD<String> traducciones
} // VD == VECTOR DINÁMICO

# Tipo rep 2: Con un vector estático.
class Frase{
private:
    String origen ;
    String traducciones[20]
}

# Tipo rep 3: Con un conjunto de celdas enlazadas.
class Frase{
struct Celda{
    String traduccion ;
    Celda *siguiente ;
}

    Celda *primera ;
    String origen ;
}
```

El segundo tipo rep no es adecuado ya que a priori no sabemos cuantas traducciones puede tener una frase (algunas tendrán más y otras menos). Las otras dos opciones son prácticamente igual de buenas. Seleccionaré el uso de un vector dinámico pues este proporciona mayor facilidad para acceder a una posición cualquiera del mismo y dado que el orden de las traducciones no es importante, no necesitaremos añadir datos en posiciones intermedias (lo cual

sería una operación lenta con VD y no con celdas enlazadas). (Además, las celdas enlazadas son más complejas de utilizar y no aportan casi ninguna ventaja sobre el vector dinámico para este problema).

T.D.A Conjunto de Frases:

Tipo rep:

```
class ConjuntoFrases{
private:
    VD<Frase> frases ;
} // VD == VECTOR DINÁMICO

class ConjuntoFrases{
private:
    Frase frases[20] ;
}

class ConjuntoFrases{
private:
    struct Celda{
        String origen ;
        Celda *siguiente ;
    }

    Celda *primera;
}
```

Al igual que en el caso anterior, no es viable realizar el tipo rep. con un vector de tamaño fijo porque el número de frases a traducir puede variar mucho de un uso a otro, como utilizamos un vector dinámico para el tipo rep Frase y este también se adapta correctamente a este tipo, lo utilizaremos aprovechando las clases plantilla.

4. Invariante de representación y función de abstracción.

T.D.A Frase:

Función de abstracción:

$f(r) = \{r.\text{origen}; r.\text{traducciones}[0]; r.\text{traducciones}[1]; \dots; r.\text{traducciones}[\text{traducciones.size()}]\}$

Invariante de representación:

$r.\text{traducciones.size()} > 0$

T.D.A Conjunto de Frases:

Función de abstracción:

$f(r) = \{ r.\text{frases}[0].\text{origen} ; r.\text{frases}.\text{traducciones}[0] ; \dots ; r.\text{frases}.\text{traducciones}[\text{traducciones.size()}]$

.....

.....

$r.\text{frases}[\text{frases.size()}].\text{origen} ; r.\text{frases}[\text{frases.size()}].\text{traducciones}[0] ; \dots r.\text{frases}[\text{frases.size()}].\text{traducciones}[\text{traducciones[size()]}] \}$

Invariante de representación:

para todo i, j $\text{frases}[i].\text{origen} \neq \text{frases}[j].\text{origen}$

