

Práctica 4: STL e Iteradores

SESIÓN I

Gustavo Rivas Gervilla



UNIVERSIDAD
DE GRANADA



DECSAI

Con esta práctica

- ▶ seguiremos practicando la abstracción y la documentación de clases.
- ▶ trabajaremos con la STL.
- ▶ tenéis unas transparencias que resumen la STL.

Contenido

El T.D.A Diccionario

La práctica

Un Diccionario es un conjunto de entradas del siguiente tipo:

```
1 | template <class T,class U>
2 | struct data{
3 |     T clave;
4 |     list<U> info_asoci;
5 | };
```

Como vemos en la Línea 4, para almacenar dichas entradas usamos la estructura de datos `STL::list`.

Este T.D.A cuenta con los siguientes métodos:

```
1 void Copiar(const Diccionario<T,U>& D);
2 void Borrar();
3 Diccionario():datos(list<data<T,U> >());
4 Diccionario(const Diccionario &D);
5 ~Diccionario();
6 Diccionario<T,U> & operator=(const Diccionario<T,U> &D);
7 bool Esta_Clave(const T &p, typename list<data<T,U> >::iterator &it_out)
   ;
8 void Insertar(const T& clave,const list<U> &info);
9 void AddSignificado_Palabra(const U & s ,const T &p);
10 list<U> getInfo_Asoc(const T & p);
11 int size() const;
12
13 typename list<data<T,U> >::iterator & begin();
14 typename list<data<T,U> >::iterator & end();
15 typename list<data<T,U> >::const_iterator & begin() const;
16 typename list<data<T,U> >::const_iterator & end() const;
```

Los operadores >> y << están implementados en el fichero
usodiccionario.cpp

Contenido

El T.D.A Diccionario

La práctica

- ▶ Añadir al menos 3 métodos nuevos al `T.D.A Diccionario` que aumenten su funcionalidad (cuanto más la completéis pues mejor).
- ▶ Crear la misma estructura de ficheros que hemos venido manejando en prácticas anteriores. Por tanto los métodos que vosotros añadáis se declaran en el `.h` y se definen en el `.cpp`.

- ▶ Esta es la última práctica puntuable. Se entrega el 9/12/2018.
- ▶ La podéis hacer por parejas.
- ▶ Intentad añadir el @authors en cada fichero de código.
- ▶ La semana que viene trabajaremos con el T.D.A Guia_Tlf.

¡Buena semana!