#include "Error.h"

#include <string.h>

#include <stdio.h>

#include <fstream>

#define error1 "El jugador no pertenece a un club existente"

#define error2 "Valor total supera al maximo valor por equipo"

#define error3 "El valor de Arqueros supera el valor permitido"

#define error4 "El valor de Defensores supera el valor permitido"

#define error5 "El valor de Mediocampistas supera el valor permitido"

#define error6 "El valor de Delanteros supera el valor permitido"

#define error7 ""

/\*\*

PRE: La instancia del TDA(error) no debe haber sido creada ni destruida con anterioridad.

POST:Deja al TDA(error) listo para ser utilizado, con las variables por defecto:

CodigoError=1

Descripcion=" "

Atrivutos:errorInstancia a la cual se aplica la primitiva.

RETORNO:No aplica.

\*/

void crear(Error &error)

{

error.CodigoError=1;

error.Descripcion=" ";

}

/\*\*

PRE: La instancia del TDA(error) debe haber sido creada y no destruida con anterioridad.

POST:Destruye la instancia del TDA(error) y no podra ser utilizada.

Atrivutos:error:Instancia a la cual se aplica la primitiva.

RETORNO:No aplica.

\*/

void destruir(Error &error)

{

}

/\*\*

PRE: La instancia del TDA(error) debe haber sido creada y no destruida con anterioridad.

El codigo de error debe ser entre 1 y 7;

POST:Le asigna el nuevo CodigoError al error.

Atrivutos:error:Instancia a la cual se aplica la primitiva.

CodigoError: Valor del codigo de error

RETORNO:No aplica.

\*/

void setCodigoError(Error &error,int CodigoError)

{

error.CodigoError=CodigoError;

SetDescripcion(error,CodigoError);

}

/\*\*

PRE: La instancia del TDA(error) debe haber sido creada y no destruida con anterioridad.

El codigo de error debe ser entre 1 y 7;

POST:Le asigna la nueva descripción al error.

Atrivutos:error:Instancia a la cual se aplica la primitiva.

Descripcion: Valor de la descripcion de error

RETORNO:No aplica.

\*/

void SetDescripcion(Error &error,int CodigoError)

{

switch(CodigoError)

{

case 1:error.Descripcion=error1;break;

case 2:error.Descripcion=error2;break;

case 3:error.Descripcion=error3;break;

case 4:error.Descripcion=error4;break;

case 5:error.Descripcion=error5;break;

case 6:error.Descripcion=error6;break;

case 7:error.Descripcion=error7;break;

}

}

/\*\*

PRE: La instancia del TDA(error) debe haber sido creada y no destruida con anterioridad.

POST:Devuelve el codigo de error de la instancia del TDA(error),

Atrivutos:error:Instancia a la cual se aplica la primitiva.

RETORNO:Devuelve el codigo de error.

\*/

int getCodigoError(Error &error)

{

return(error.CodigoError);

}

/\*\*

PRE: La instancia del TDA(error) debe haber sido creada y no destruida con anterioridad.

POST:Devuelve la descripción de la instancia del TDA(error),

Atrivutos:error:Instancia a la cual se aplica la primitiva.

RETORNO:Devuelve la descripción de error.

\*/

std::string getDescripcion(Error &error)

{

return error.Descripcion;

}