

Semana 5: Tipos de Datos Abstractos

Semestre 1 - 2024

Análisis de Algoritmos y Estructura de Datos

Departamento de Ingeniería Informática

Contenido

- Especificación de TDA
- Implementación en C de TDA

Objetivos

- Implementar tipos de datos abstractos utilizando el lenguaje de programación C.

Actividad 1: TDA punto

- Sea P un punto en el plano cartesiano, especificar el dominio (estructura de datos) y operaciones para trabajar con la estructura.



Actividad 1: TDA punto dominio

- Estructura de datos:
Un punto $P = (x, y)$ se define por las coordenadas x e y , donde ambas coordenadas son números reales.
- Pseudocódigo:

```
REGISTER punto:  
  NUM x  
  NUM y
```

Actividad 1: TDA punto operaciones

- **crea_punto**(NUM x, NUM y): punto
 - Crea un punto en el plano asignando los valores de las coordenadas **x** e **y**.
 - Devuelve el punto creado.
- **calcula_distancia**(punto A, punto B): NUM
 - Calcula la distancia entre el punto A =(x_a,y_a) y el punto B=(x_b,y_b) con la fórmula: $d = ((y_b - y_a)^2 + (x_b - x_a)^2)^{1/2}$.
 - Devuelve la distancia obtenida.
- **distancia_origen**(punto A): NUM
 - Calcula la distancia entre el punto O =(0,0) y el punto A=(x, b) con la fórmula: $d = (y^2 + x^2)^{1/2}$.
 - Devuelve la distancia obtenida



Actividad 2: implementación TDA

- La implementación se divide en tres archivos:
 1. Una interfaz, que define la estructura de datos y declara prototipos de las funciones que se utilizarán para operar la estructura de datos (.h).
 2. Una implementación de las funciones declaradas en la interfaz (.c).
 3. Un programa cliente que utiliza las funciones declaradas en la interfaz para trabajar en un nivel superior de abstracción (.c).

Actividad 2: implementación TDA

- Implementar en lenguaje C el TDA punto en el plano cartesiano



Actividad 2: interfaz .h

- Estructura de datos

```
typedef struct{  
    float x;  
    float y;} punto;
```

- Operaciones

```
// prototipos  
punto crea_punto(float x, float y);  
  
float calcula_distancia(punto a, punto b);  
  
float distancia_origen(punto a);
```

Actividad 2: implementación .c

- **crea_punto**(NUM x, NUM y): punto
- Implementación:

```
#include <math.h>
#include "TDApunto.h"
```

```
punto crea_punto(float x, float y){
    punto nuevo_punto;
    nuevo_punto.x = x;
    nuevo_punto.y = y;
    return nuevo_punto;
}
```

Actividad 2: implementación .c

- **calcula_distancia(Punto A, Punto B): NUM**
- Implementación

```
float calcula_distancia(punto a, punto b){  
    float dx = a.x - b.x;  
    float dy = a.y - b.y;  
    return sqrt(dx * dx + dy * dy);  
}
```

- El archivo de cabecera math.h contiene la función sqrt() #include <math.h>

Actividad 2: implementación .c

- **distancia_origen(Punto A): NUM**
- Implementación

```
float distancia_origen(punto a){  
    punto origen = crea_punto(0.0, 0.0);  
    return calcula_distancia(origen, a);  
}
```

Actividad 3: programa cliente .c

- Implementar en lenguaje C un programa que solicite dos puntos y que muestre por pantalla la distancia entre ellos.



Actividad 3: programa cliente .c

- Implementar en lenguaje C un programa que solicite dos puntos y que muestre por pantalla la distancia entre ellos.

```
#include <stdio.h>
#include "TDApunto.h"

int main(void) {

    return 0;
}
```

Actividad 4: TDA conjunto

- Compilar y ejecutar lab05-conjunto.c
- Experimentar con las funciones de TDAconjunto, haciendo llamadas desde la función main() en lab05-conjunto.c para realizar lo siguiente:
 1. Crear un conjunto
 2. Mostrar los elementos del conjunto
 3. Agregar el elemento 2
 4. Agregar el elemento 4
 5. Mostrar los elementos del conjunto
 6. Eliminar el elemento 4
 7. Mostrar los elementos del conjunto

Actividad 5: TDA conjunto

- Implementar las siguientes operaciones del TDA conjunto:
- **union_conjuntos**(conjunto a, conjunto b): conjunto
 - Crea un conjunto que contiene todos los elementos, que pertenecen por lo menos a uno de los conjuntos a o b.
 - Retorna el conjunto creado.
- **interseccion_conjuntos**(conjunto a, conjunto b): conjunto
 - Crea un conjunto cuyos elementos, necesariamente, pertenecen a ambos conjuntos a y b.
 - Retorna el conjunto creado.
- **diferencia_conjuntos**(conjunto a, conjunto b): conjunto
 - Crea un conjunto que contiene los elementos de a que no están en b.
 - Retorna el conjunto creado.

Actividad 5: TDA conjunto

- Verificar el funcionamiento de las operaciones implementadas con los conjuntos:
 - $A=\{1,2,3,4,5\}$ y $B=\{4,5,6,7,8,9\}$
 - $A \cup B = \{1,2,3,4,5,6,7,8,9\}$
 - $A \cap B = \{4,5\}$
 - $A - B = \{1,2,3\}$

Entrega de actividad de laboratorio

- Entrega obligatoria
- Subir SOLO actividad nro. 5 de esta sesión en buzón de Campus Virtual en un único archivo **s5_apellido_nombre.zip**
- Se espera TDAconjunto.h, TDAconjunto_implmentacion.c y lab05-conjunto.c comprimidos en archivo .zip
- Plazo: hoy dentro del horario de laboratorio de cada coordinación

Actividad de cierre

¿Completó las actividades síncronas?

¿Qué dificultades encontró?

