

Tema 1. Tipos Abstractos de Datos (TAD)

Estructura de Datos y Algoritmos (EDA)



Objetivos

- ▶ Entender el concepto de **abstracción**
- ▶ Entender los conceptos de **Tipo Abstracto de Datos (TAD)** and **estructura de datos** y explicar la diferencia entre ambos
- ▶ Aprender **cómo especificar formalmente un TAD**
- ▶ Escribir **implementaciones correctas** para TAD simples

Contenidos

- ▶ Abstracción
- ▶ Tipo Abstracto de Datos (TAD)

Abstracción

- ▶ Centrarse en los detalles relevantes para un problema y descartar detalles irrelevantes



Estudiante
...
NotaAcceso
ListadeAsignaturas

Paciente
...
Altura
ListadeAlergías
ListadeOperaciones

Permitir definir software (qué debe hacer sin decir cómo debe hacerse)

Tipo Abstracto de Datos (TAD)

- ▶ Un modelo matemático con una colección de operaciones definidas sobre un conjunto de datos
 - ▶ Ejemplo: número complejo con las operaciones de suma, multiplicación, resta y módulo
- ▶ Una estructura de datos es una representación (**implementación**) de un TAD en un lenguaje de programación. Las estructuras de datos a menudo son colecciones de variables de diferentes tipos de datos
- ▶ Un TAD puede tener varias implementaciones (estructuras de datos)

Especificación

- ▶ Un TAD puede describirse usando dos maneras diferentes:
 - ▶ **Especificación No-Formal** (usando lenguaje natural)
 - ▶ **Especificación Formal** (usando pseudo-código o incluso algún lenguaje de programación)
 - ▶ En JAVA, las interfaces se utilizan para definir los TAD

TAD complejo (Especificación informal)

Un número complejo se puede representar como un par de números: parte real y parte imaginaria. Este TAD implica las siguientes operaciones:

▶ *sum(Complex)*:

$$(a+ib) + (c+id) = (a+c) + (b+d)i$$

▶ *times(Complex)*:

$$(a+ib) * (c+id) = (ac-bd) + (ad+bc)i$$

▶ *module()*: devuelve el módulo del número complejo.

$$|a+ib| = \sqrt{a^2+b^2}$$

Especificación informal

Operaciones

- ▶ ***getReal ()***: devuelve su parte real
(a+ib->a)
- ▶ ***getImag()***: devuelve su parte imaginaria
(a+ib->b)
- ▶ ***setReal(x)***: toma un número *x* como entrada y lo asigna a su parte real
- ▶ ***setImag(x)***: toma un número *x* como entrada y lo asigna a su parte imaginaria

Especificación formal de un TAD Complejo

```
public interface iComplex {  
  
    public float getReal();  
    public float getImag();  
  
    public void setReal(float x);  
    public void setImag(float x);  
  
    public float module();  
  
    public iComplex sum(iComplex obj);  
    public iComplex times(iComplex obj);  
  
    public boolean isequals(iComplex obj);  
}
```

Estructuras de datos: implementaciones de TAD

- ▶ En Java, las estructuras de datos se implementan como clases
- ▶ Una estructura de datos (class) debe implementar todas las operaciones definidas en su TAD (interfaz)
- ▶ Una class describe cómo se realizan las operaciones de TAD

```
public class Complex implements iComplex {
```

```
    private float real;  
    private float imag;  
    public Complex(float r, float i) {  
        real=r;  
        imag=i;  
    }  
    public float getReal() {  
        return real;  
    }  
    public float getImag() {  
        return imag;  
    }  
    public void setReal(float x) {  
        real=x;  
    }  
    public void setImag(float x) {  
        imag=x;  
    }  
}
```

La clase debe implementar todos los métodos definidos en la interfaz

```
public iComplex sum(iComplex obj) {
    iComplex result=new Complex(real+obj.getReal(),imag+obj.getImag());
    return result;
}
```

```
public iComplex times(iComplex obj) {
    iComplex result=new Complex(real*obj.getReal()-imag*obj.getImag(),
        real*obj.getImag()+imag*obj.getReal());
    return result;
}
```

```
public boolean isequals(iComplex obj) {
    return (real==obj.getReal() && imag==obj.getImag());
}
```

Resumen

- ▶ Un TAD define **qué operaciones**, pero no **cómo hacerlas** (implementar)
- ▶ Un TAD puede tener diferentes implementaciones
- ▶ En Java, las **interfaces** permiten definir formalmente TAD
- ▶ En Java, las **clases (classes)** permiten implementar TAD
- ▶ Una clase (**class**) que implementa una interfaz debe implementar todos los métodos de la interfaz

uc3m | Universidad **Carlos III** de Madrid

