

Programación II

Práctica 02b: Tipos Abstractos de Datos (TAD)

Avanzados

Versión del 20/03/2022

Nota importante: Recordar que en todos los casos se debe realizar primero la especificación, donde **nos focalizamos en qué debe hacer el TAD y no cómo**, en esa etapa no pensamos en código, luego en la implementación, resolvemos como se realizan las distintas operaciones planteadas para el TAD.

En cada ejercicio se pide escribir la interfaz de los TADs pedidos en cada ejercicio, y de los TADs que necesite cada uno para cumplir con lo solicitado de una manera correcta.
Implementar en Java solamente cuando se indique. Escribir el IREP en cada caso que se implemente.

Ejercicio 1: Fecha

Modelar el TAD fecha, una fecha está formada por 3 números enteros, cuyos valores deben comportarse de la siguiente manera:

- Día, puede variar entre 1 y 30
- Mes, puede variar entre 1 y 12
- Año debe ser mayor a 1900

Se desea que el TAD cumpla con las siguientes especificaciones:

- Dada una fecha se pide saber cuántos días faltan para una fecha dada,
- Dada una fecha y un período en días se desea saber qué fecha será.
- Según el año de la fecha, decir si el año es bisiesto.

Ejercicio 2: Agenda

Implementar una Agenda de personas basada en el TAD diccionario

El acceso a la agenda se hace por el DNI de la persona (será la Clave del Diccionario) y un significado (o Valor) con los datos que interesan de esa persona:

Su Nombre que es una cadena (String), su teléfono (Integer) y su dirección (String).

Ayuda: Considerar el utilizar otro TAD como abstracción del significado de cada entrada del diccionario. Se puede modificar el TAD Diccionario, de manera que la firma de la operación de agregar una nueva entrada sea:

```
Void agregar (Integer dni, Significado s);
```

Ejercicio 3: Robot

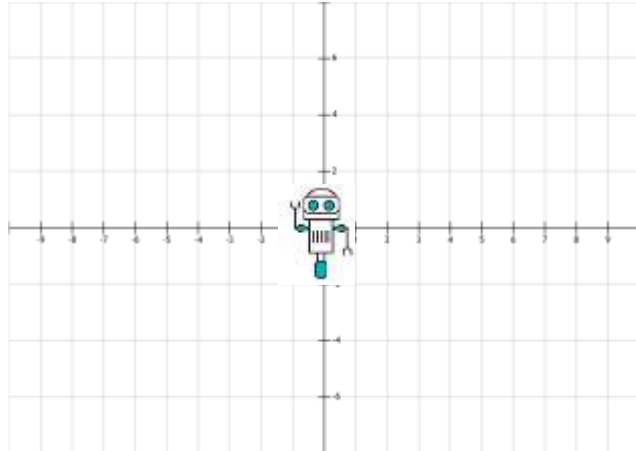
Crear un TAD robot, un robot está dado por su posición en el plano, X e Y, ambas son números



enteros.

Puede moverse sobre el plano siguiendo esta especificaciones:

- Puede avanzar en línea recta una cantidad de posiciones determinada, si no se indica cuántas posiciones debe avanzar, deberá avanzar una sola.
- No puede retroceder.
- Puede girar a izquierda o derecha.
- Se debe poder informar en todo momento la posición del robot.



Suponer que el plano es infinito (o sea, no se pide controlar que se salga del plano). Inicialmente está en la posición 0,0.

- a) Especificar y definir la interfaz
- b) Implementar

Ejercicio 4: El ahorcado (Diseño de la interfaz)

En el juego del ahorcado a partir de una palabra secreta, el jugador debe adivinar esta palabra, arriesgando las letras que la componen. Si luego de una cantidad de intentos no lo hizo entonces pierde.

Modelar el juego, para lo cual se dan las siguientes indicaciones:

- Una partida se crea con una palabra inicial.
- El jugador arriesga una letra y el juego le debe informar si acertó o no, en caso de que lo haga, muestra cómo se va completando la palabra, si no acierta se contabiliza como error y luego de 6 errores pierde.
- Si la letra no era correcta también se debe informar al jugador, cuantos intentos le quedan. En cualquier momento el jugador puede arriesgar la palabra, si no acierta pierde un punto. La partida finaliza porque se le acabaron los intentos o porque adivinó la palabra.

- 1) Realizar la especificación y la interfaz.
- 2) Implementarlo creando la clase Ahorcado en Java.

Ejercicio 5: Juego del ahorcado (Diseño e Implementación).

A partir de la interfaz del TAD Ahorcado diseñado en el punto anterior se pide crear el juego. Para ello se pide **modelar el TAD Juego** (especificar y definir interfaz) **y luego implementarlo**, utilizando el TAD Ahorcado del ejercicio anterior.

El Juego debe respetar las siguientes indicaciones:

- Se deberá contar inicialmente con 10 palabras para crear una partida.
- Se seleccionará una palabra al azar al momento de crearla.
- Las palabras que se van utilizando, no deben volver a salir.
- Cuando se hayan usado todas las palabras, todas vuelven a estar disponibles.

Ejercicio 5: Matriz infinita de booleanos

El departamento de matemáticas de la UNGS nos pidió ayuda para implementar el TAD “Matriz infinita de booleanos”

La implementación(trivial) actual es la siguiente:

```
public class MIB {  
  
    private int i;  
    private int j;  
    private Boolean [][] mat;  
  
    public MIB(int i, int j){  
        this.i=i;  
        this.j=j;  
        mat = new Boolean[i][j];  
    }  
  
    Boolean leerValor(int i, int j){  
        return mat[i][j];  
    }  
    public void setearValor(int i, int j, Boolean x){  
        mat[i][j]=x;  
    }  
}
```

El problema de esta implementación, es que ocupa demasiada memoria. Por

ejemplo “MIB x = new MIB(100000,200000)” arroja:

java.lang.OutOfMemoryError: Java heap space

Lo que se solicita es:

a) Implementar la función test() que implemente el siguiente testeo:

```
MIB x = new MIB(100000,200000);  
x.setearvalor(5000,3000,true);  
System.out.println(x.leervalor(5000,3000)); // debe devolver true  
System.out.println(x.leervalor(5000,3001)); // debe devolver false
```

b) Hacer el diseño de un TAD que represente la matriz de la manera más eficiente.

Ayuda: Se puede asumir que la mayoría de los valores van a ser *False*.

c) Calcular el orden de complejidad para los métodos *leerValor* y *setearValor* de la implementación trivial.

d) Calcular el orden de complejidad para los métodos *leerValor* y *setearValor* de la implementación mejorada.

6)

- a) Necesitamos crear el TAD para comprobar si un ciudadano tiene el esquema de vacunación COVID completa.

El esquema de vacunación está completo cuando tiene dos dosis aplicadas. Se registrará el número de documento del ciudadano y la cantidad de dosis aplicadas.

Considerar que las dosis se aplican en distinto momento.

Las situaciones posibles de un ciudadano son:

No inició la vacunación (No posee vacunas)

Esquema de vacunación incompleto (1 dosis)

Esquema de vacunación completo (2 dosis)

Primer refuerzo (3 dosis)

Segundo refuerzo (4 dosis)

Se debe informar la situación actual cuando se consulta por un ciudadano en particular.

- b) Cómo se modifica el TAD anterior si además se desea registrar la fecha en que se aplicó cada dosis y cuál fue la vacuna aplicada.