



**UNIVERSIDAD AUTONOMA DE ENCARNACION**

**LENGUAJE DE PROGRAMACION 3**

**RESPONSABLE:**

- JUAN ALBERTO GODOY
- PAMELA GONZÁLEZ MEDINA

**DOCENTE: OSVALDO MICNIUK**

**CURSO: 3°**

**SEMESTRE: 5°**

**AÑO:2020**

## TAREA 3

### Objetivo/s

- Repasar conceptos de Objetos y sus partes.
- Fijar conceptos básicos de herencia.
- Fijar el conocimiento adquirido.
- Desarrollar la capacidad de debate.
- Fijar el conocimiento adquirido

### Descripción de sus actividades

En el marco de la programación orientada a objetos.

#### 1) Busca los conceptos de

**Clases:** Una clase es una descripción de todos los objetos que define, como tal, es una abstracción; un concepto a partir de cualquier instancia en particular. Cada objeto es una instancia de su clase, así pues para propósitos prácticos objeto e instancia son sinónimos. Una clase puede tener cualquier o ningún número de objetos asociados con ella.

**Objetos:** Un objeto es un conjunto de datos relacionados que identifican el estado actual del objeto más un conjunto de comportamientos

**Atributos:** Son datos de miembro o variables de instancias, porción de información que un objeto posee o conoce de sí mismo.

**Métodos:** Los métodos, como saben, pueden tener una serie de parámetros en su declaración, que se denominan parámetros formales. Por ejemplo, un método que realice el cuadrado de un número (elevar a 2 este número), como se muestra a continuación:

```
public static double cuadrado(double n){  
  
    return n*n;  
  
}
```

En esta declaración del método, «n» es el parámetro formal

Cuando el método se invoca, se le pasan entre paréntesis los valores de los parámetros y a éstos se les llama parámetros actuales, por ejemplo: `double resultado = cuadrado(4);`

**Paso de parámetros:** El paso de parámetros en Java solo se realiza por valor y no por referencia, aunque pueda parecer que sí. Generalmente, el paso de parámetros, o comunicación de datos del algoritmo invocante al método invocado, puede hacerse de dos formas:

**Por valor:** los parámetros pasados al método no se modifican para fuera de este método.

**Por referencia:** los parámetros pasados al método sí se modifican para fuera de este método (Disquis).

**2) Ya conocidos los conceptos listados más arriba, debate en grupo describe con tus palabras:**

**a) ¿Cuál es la relación que existe entre los objetos y las clases?**

Un objeto es una cosa tangible, algo a que se puede aprehender intelectualmente o algo hacia lo que se puede dirigir una acción o pensamiento. Un objeto representa un ítem individual e identificable, o una entidad real o abstracta, con un papel definido en el dominio del problema. Un objeto tiene: Estado, Comportamiento, Identidad.

Una clase es un conjunto de objetos que comparten una estructura y comportamiento comunes.

**La declaración de una clase sigue la siguiente sintaxis:**

```
[modificadores] class IdentificadorClase {  
  
    // Declaraciones de atributos y metodos  
  
    ...  
  
}
```

**b) ¿Qué entiendes por Atributos de objetos y como identificarías los atributos de un objeto dado? Describe un ejemplo.**

modifVisibilidad indica desde que parte del código se puede acceder a la variable:

- **public:** indica que es un atributo accesible a través de una instancia del objeto.
- **private:** indica que a través de una instancia no es accesible el atributo. Al heredar el atributo se convierte en inaccesible.
- **protected:** indica que a través de una instancia no es accesible el atributo. Al heredar si se puede usar desde la clase derivada.
- Sin especificar: indica visibilidad de paquete, se puede acceder a través de una instancia, pero sólo desde clases que se encuentren en el mismo paquete.

modifAtributos son características específicas del atributo, son:

- **static:** El atributo pertenece a la clase, no a los objetos creados a partir de ella.

- **final:** El atributo es una constante, en ese caso debe de tener valor inicial obligatoriamente. Por convenio en java las constantes se escriben en mayúsculas.
- **transient:** Marca al atributo como transitorio, para no ser serializado. Lo emplearemos en java beans.
- **volatile:** es un atributo accedido de forma asíncrona mediante hilos, con este atributo se lo notificamos a java (Albasanz).

**c) ¿Qué entiendes por Métodos de objetos y como identificarías los atributos de un objeto dado. Describe un ejemplo.**

Un método es una subrutina cuyo código es definido en una clase y puede pertenecer tanto a una clase, como es el caso de los métodos de clase o estáticos, como a un objeto, como es el caso de los métodos de instancia.

El comportamiento de los objetos de una clase se implementa mediante funciones miembro o métodos. Un método es un conjunto de instrucciones que realizan una determinada tarea y son similares a las funciones de los lenguajes estructurados.

Del mismo modo que hay variables de instancia y de clase, también hay métodos de instancia y de clase. En el primer caso, un objeto llama a un método para realizar una determinada tarea, en el segundo, el método se llama desde la propia clase.

Los atributos, también llamados datos o variables miembro son porciones de información que un objeto posee o conoce de sí mismo. Una clase puede tener cualquier número de atributos o no tener ninguno. Se declaran con un identificador y el tipo de dato correspondiente. Además los atributos y tienen asociado un modificador que define su visibilidad según se muestra en la siguiente tabla (Eguiluz, 2014).

| Modificador | Visibilidad                   |
|-------------|-------------------------------|
| public      | Pública (+)                   |
| protected   | Protegida / en la herencia(#) |
| private     | Privada(-)                    |
| package     | De paquete (~)                |

**d) Describe y ejemplifica que entiendes por paso de parámetros**

El paso de parámetros en Java es siempre por valor pero al pasar objetos se envía la dirección de memoria del mismo, por lo que, en la práctica, funciona como si fuera por referencia.

**Por ejemplo**, en el siguiente programa el método cambiar utiliza un parámetro de tipo primitivo (a, un valor numérico entero) y un parámetro de tipo referencia (b, un array de valores enteros):

```
/**  
  
 * Ejemplo de uso de parametros de distintos tipos  
  
 */  
  
public class Parametros {  
  
    public static void main (String [] args ) {  
  
        int n;  
  
        int [] v = new int[2];  
  
        n=10;  
  
        v[0]=20;  
  
        v[1]=30;  
  
        System.out.println("Antes:  " + n + " " + v[0] + " "+ v[1]);  
  
        cambiar(n, v);  
  
        System.out.println("Después: " + n + " " + v[0] + " "+ v[1]);  
  
    }  
  
    public static void cambiar (int a, int [] b) {  
  
        a = 50;  
  
        b[0] = 60;  
  
        b[1] = 70;  
  
        System.out.println("Dentro: " + a + " " + b[0] + " "+ b[1]);  
  
    }  
  
}
```

**La ejecución del código anterior origina la siguiente salida por pantalla:**

\$>java Parametros

Antes: 10 20 30

Dentro: 50 60 70

Despues: 10 60 70

### 3) Investiga sobre herencia contesta las siguientes preguntas

#### ¿De qué se trata la herencia en POO?

Es el mecanismo de implementación mediante el cual elementos más específicos incorporan la estructura y comportamiento de elementos más generales (Cristina Cachero)

#### ¿Qué son súper clases y sub clases en java?. ¿Cómo se relacionan?

La **superclase** sería la clase que tiene todos los atributos y métodos en común, y las **subclases** van a ser las que heredan de la primera, con ello establecemos un sistema de jerarquía de clases, donde a mayor especificación se van bajando los niveles y no hay límite, podemos tener un clase que herede de otra clase que a su vez ha heredado de otra más, con ello podemos ir traspasando atributos comunes a una clase inferior que será muy específica pero que puede compartir con otros objetos una serie de funcionalidades.

El **ejemplo clásico** que se utiliza para describir este comportamiento son las clases de figuras geométricas, este caso sirve de estudio debido a que todas las figuras comparte algunos atributos, por ejemplo, pueden tener un color, pueden estar llenas con este color y después al detalle cada una tiene sus características propias, como por ejemplo la forma en que se calcula el área, si tiene lados iguales o lados diferentes o radio en el caso de las circunferencias.

#### ¿Qué es una clase abstracta en java?

**Abstracción:** Capacidad de especificar las características comunes a un conjunto de clases ,, Definición parcial del estado y del comportamiento .Declaración del comportamiento (interfaz)

- **Clases abstractas:** Especificación de datos y comportamiento común a un conjunto de clases. Forzar a que las subclases proporcionen un comportamiento específico ,,
- **Interfaces:** Declaración de métodos a incorporar en las clases que implementen la interfaz. Definición de constantes.

#### Cita los beneficios del uso de la Herencia en POO.

- ❖ La herencia no permite cambios en tiempo de ejecución ,,
- ❖ La herencia rompe la encapsulación ,,
- ❖ La herencia impone al menos una parte de la representación física a las subclases ,, Cambios en la superclase pueden afectar a las subclases ,,
- ❖ Las implementaciones de superclase y subclases están ligadas ,,

- ❖ Si hacen falta cambios para reutilizar una clase en nuevos dominios de aplicación habrá que cambiarla ,,
- ❖ Limita la flexibilidad y al final la reutilización. Por ello es más práctico utilizar interfaces y clases abstractas. Ya que proporcionan menos o ninguna implementación (Mestras, 2004).