

Trabajo Práctico N° 1: Herencia y Polimorfismo

1. Dadas la siguiente jerarquía de herencia:

```
class A {
    private int atributoA;
    void metodoA() { }
}
class B extends A {
    private int atributoB;
    void metodoB() { }
}
class C extends B {
    private int atributoC;
    void metodoC() { }
}
```

a) ¿qué mensajes se pueden enviar a un objeto de la clase A?
b) ¿qué mensajes se pueden enviar a un objeto de la clase B?
c) ¿qué mensajes se pueden enviar a un objeto de la clase C?
d) ¿qué atributos describen a un objeto de la clase A?
e) ¿qué atributos describen a un objeto de la clase B?
f) ¿qué atributos describen a un objeto de la clase C?

2. Dada la siguiente jerarquía de herencia:

```
class A {
    int atributoA=5;
    void metodoA() { System.out.println(" *** metodo A" ***); }
}
class B extends A {
    int atributoB=2;
    void metodoA() { System.out.println(" *** método A redefinido en B" ***); }
    void metodoB() { System.out.println(" *** método B" ***); }
}
class C extends B {
    int atributoC=0;
    int atributoB=7;
    void metodoC() { System.out.println(" *** método C" ***); }
}
```

- a) ¿qué ocurre cuando se le envía el mensaje metodoA a un objeto de la clase A?
b) ¿qué ocurre cuando se le envía el mensaje metodoA a un objeto de la clase B?
c) ¿qué ocurre cuando se le envía el mensaje metodoA a un objeto de la clase C?
d) ¿qué valor tiene la propiedad atributoB de un objeto de la clase B?
e) ¿qué valor tiene la propiedad atributoB de un objeto de la clase C?
f) ¿qué valor tiene la propiedad atributoB de un objeto de la clase A?
g) ¿qué valor tiene la propiedad atributoA de un objeto de la clase C?
h) ¿qué valor tiene la propiedad atributoA de un objeto de la clase B?

3. Dadas las siguientes jerarquías de herencia indique la salida que produce Prueba y sus conclusiones:

```
class Prueba {
    public static void main (String [] args) {
        Hija x;
        x = new Hija();
        System.out.println("Atributo j de X:" + x.getJ());
        System.out.println("Atributo i de X:" + x.getI());
    }
}
```

3.1.- // clase base y clase derivada tienen constructores

```
class Base {
    private int i;
    Base() { i=100; }
    int getI() { return i; }
}
class Hija extends Base {
    private int j;
    Hija() { j=120; }
    int getJ() { return j; }
}
```

3.2.- // clase derivada no tiene constructor

Programación Orientada a Objetos
Analista de Sistemas – Licenciatura en Sistemas

```
class Base {
    private int i;
    Base() { i=100; }
    int getI() { return i; }
}
```

```
class Hija extends Base {
    private int j;
    int getJ() { return j; }
}
```

3.3.- // clase base no tiene constructor sin argumentos

```
class Base {
    private int i;
    Base(int n) { i =n; }
    int getI() { return i; }
}
```

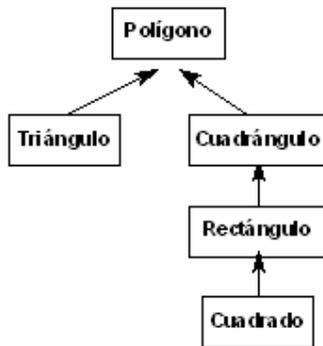
```
class Hija extends Base {
    private int j;
    Hija() { j =120;}
    int getJ() { return j; }
}
```

3.4.- // clase base tiene constructor con argumentos

```
class Base {
    private int i;
    Base(int n) { i =n; }
    int getI() { return i; }
}
```

```
class Hija extends Base {
    private int j;
    Hija() {
        super (110);
        j =120;}
    int getJ() { return j; }
}
```

4. Teniendo en cuenta la siguiente jerarquía de clases:



Y las declaraciones:
 Triangulo t;
 Polígono p;
 Cuadrángulo c;
 Rectángulo r;
 Cuadrado cd;

Indique cuales de las siguientes sentencias no son validas y por qué:

<pre>p = c; p = r; cd = c; c = t;</pre>	<pre>p = r; r = p; p.diagonal(); // diagonal es una propiedad de los cuadrángulos t.perimetro(); // perimetro es una propiedad de los polígonos</pre>
---	---

5. Explique claramente cual es el error en la siguiente jerarquía de herencia implementada en Java:

<pre>public class Uno { public int f() { return k();} public int k() { return 50;} }</pre>	<pre>public class Dos extends Uno { public int f() { return super.f() + h();} public int h() { return 20;} }</pre>	<pre>public class Tres extends Dos { public int f() { return 80;} public int k() { return super.f() + 10;} }</pre>
---	---	---

6. Dada la siguiente definición de clases:

<pre>public class AA { public int g() { return 5;} }</pre>	<pre>public class BB extends AA { public int g() { return 3;} public int h() { return g();} }</pre>	<pre>public class CC extends BB { public int g() { return super.g();} public int h() { return super.h() + 1;} }</pre>
--	--	--

Y las declaraciones:

AA p; BB q; CC r; int M;

Haga la **traza** para cada segmento de código. Si encuentra una sentencia errónea indique si el error es en tiempo de ejecución o de compilación. De haber error indique, además el valor de M hasta la última sentencia válida.

(a)	(b)	(c)
<pre>p = q; q = r; M = p.g(); M = M + p.h();</pre>	<pre>p = q; M = r.g(); q = r; M = M + q.h();</pre>	<pre>p = q; M = q.g() + p.g() + r.g(); q = r; r = null; M = q.h();</pre>

7. Dada la siguiente definición de clases:

<pre>public class Uno { public int f() { return g()+10;} public int g() { return 2;} }</pre>	<pre>public class Dos extends Uno { public int g() { return 50 + k();} public int k() { return 100;} }</pre>
---	---

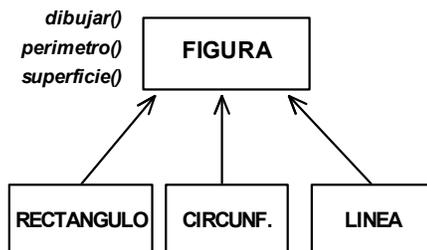
Y las declaraciones:

Uno a, b; Dos c, d; int n,m;

Indique para cada segmento de código si existen sentencias incorrectas y si el error es en tiempo de ejecución o de compilación, justificando su respuesta. Siempre que sea posible, señale además los valores finales de n y m.

(a)	(b)	(c)
<pre>a = new Uno(); c = new Dos(); b = c; m = b.k();</pre>	<pre>a = new Uno(); b = a; d = new Dos(); n = d.f(); a = d; m = a.g();</pre>	<pre>a = new Uno(); c = new Dos(); b = c; d = c; m = d.g()+c.k()+a.f(); d = (Dos)a; n = m - d.k();</pre>

8. Implementar el siguiente diagrama de clases. Codificar la entidad Figura como Interface.

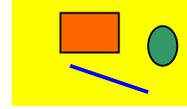


Tener en cuenta las siguientes propiedades:

- Cada tipo de figura tiene un procedimiento para su dibujo.
- a una figura se le puede calcular el perímetro
- a una figura se le puede calcular la superficie
- una figura tiene un color de fondo
- una figura tiene un color de borde

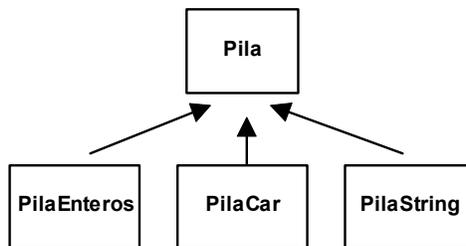
Una pizarra es un conjunto de figuras que se pueden manipular:

- agregar una Figura a la pizarra
- eliminar una Figura de la pizarra
- borrar la Pizarra
- calcula la superficie de las figuras de la pizarra
- calcular el perímetro de las Figuras de la pizarra
- cambiar el color de una Figura de la Pizarra (fondo, borde)
- cambiar el color de fondo de la pizarra.



Codificar la clase Pizarra y un menú que permita manipular la pizarra. Considerar la implementación más conveniente para Figura (clase, clase abstracta o interface).

9. Las pilas son muy utilizadas en informática. Algunos programas trabajan con pilas de enteros, otros con pilas de char, otros con pilas de String, etc. Al codificar la clase pila para cada tipo de dato, según los elementos que contiene, solo se reusa el concepto, pero no el código. En OO esta situación puede mejorarse con herencia:



a) Implemente la solución propuesta en el diagrama. Identifique que atributos y métodos pueden realmente reusarse mediante herencia.

b) Otra solución es usar polimorfismo. Esta propuesta implica que en todas las pilas los elementos son Object. Implemente este esquema

c) Compare ambas implementaciones. Cual es más reusable????

10. Una empresa de multimedios audiovisuales maneja diferentes programas en los medios de comunicación más conocidos: la radio y la televisión. Todos los programas tienen un título y una persona responsable de la dirección de la publicación. Tienen además un código numérico que los identifica unívocamente. Tanto los programas de radio como los de la televisión poseen una duración determinada en minutos junto con una hora de inicio, así como el nombre de la emisora correspondiente. Además, existe un espacio disponible para los comerciales.

En los programas de radio, este tiempo se mide en minutos al aire (por ejemplo, "LA PARADA" tiene 23 minutos de propagandas). Para la televisión, se mide en cantidad de comerciales por tanda (por ejemplo, "SHOWMATCH" tiene siete propagandas en cada tanda). Cada programa televisivo puede tener además un número diferente de tandas comerciales. Los programas de radio necesitan por otra parte un responsable de la musicalización.

Definir todas las clases necesarias para modelar los distintos productos financiados por esta empresa de multimedios.

11. Una empresa de medios impresos y gráficos financia publicaciones. Todas las publicaciones tienen un título y una persona responsable de la dirección de la publicación. Tienen además un código numérico que las identifica unívocamente. Las publicaciones pueden ser de dos tipos: diarios y revistas. Las publicaciones escritas tienen un editor responsable, una cantidad de páginas, un precio y un año de publicación. Los diarios que se imprimen son semanales o quincenales, se cobra una pequeña cantidad de recargo por envíos al interior, y se registra la semana y mes de publicación. Las revistas poseen el mes de publicación, el número y el título de la nota de tapa.

Definir todas las clases necesarias para modelar los distintos productos financiados por esta empresa de medios gráficos e impresos..

12. En un puerto se alquilan amarres para barcos de distinto tipo. Para cada ALQUILER se guarda el nombre y DNI del cliente, las fechas inicial y final de alquiler, la posición del amarre y el barco que lo ocupará. Un BARCO se caracteriza por su matrícula, su eslora en metros y año de fabricación.

Un alquiler se calcula multiplicando el número de días de ocupación (incluyendo los días inicial y final) por un valor módulo de cada barco (obtenido simplemente multiplicando por 10 los metros de eslora) y por un valor fijo (2 es en la actualidad).

Sin embargo ahora se pretende diferenciar la información de algunos tipos de barcos:

- Número de mástiles para veleros
- Potencia en cv para embarcaciones deportivas a motor
- Potencia en cv y número de camarotes para yates de lujo

El módulo de los barcos de un tipo especial se obtiene como el módulo normal más:

- El número de mástiles para veleros
- La potencia en cv para embarcaciones deportivas a motor
- La potencia en cv mas el número de camarotes para yates de lujo

Utilizando la herencia de forma apropiada, diseñe el diagrama de clases y sus relaciones, con detalle de atributos y métodos necesarios. Programe en Java los métodos que permitan calcular el alquiler de cualquier tipo de barco.

13. En un sistema Java que gestiona los usuarios de un servicio telemático se utiliza una clase Usuario:

```
class Usuario {
    private String nombre, dni;
    private int cuenta;

    Usuario (String d, String n)
    { nombre = n;
      dni = d;
      cuenta = 0;}

    void conexion (int s) // Contabiliza 's' segundos en la cuenta
    { cuenta = cuenta + s;}

    double calculaFacturación() //Calcula el importe facturable
    { return cuenta * 0,32; }

    void reset() // Pone a cero la cuenta
    { cuenta = 0;}
}
```

El departamento de marketing ha diseñado un conjunto de ofertas no acumulables para los diferentes tipos de usuario con las siguientes condiciones:

Oferta1 (para usuarios residenciales): No se contabilizan los tres primeros minutos de cada conexión del cliente.

Oferta2 (para usuarios comerciales): Se aplica un tanto por ciento de descuento sobre el importe facturable total. Este tanto por ciento se negocia por separado con cada cliente, pero una vez fijado no es modificable.

Oferta3 (para grandes usuarios): No se contabiliza la conexión más larga de cada periodo de facturación.

Elaborar las clases necesarias para resolver las nuevas necesidades del sistema considerando que no podemos modificar la clase USUARIO.

14. En una biblioteca universitaria multimedia se está construyendo un sistema de control de los fondos disponibles que incluyen documentos de distinto tipo. La forma de acceso es doble: Consultas en sala y préstamos temporales. La consulta en sala requiere registrar el número del documento, la fecha y el DNI del alumno. En el procedimiento de préstamo se manejan los mismos datos que en el caso de la consulta.

Los tipos de documentos que se contemplan son:

Los libros clásicos, en papel. Los datos que interesa conocer son: Título, autor o autores, editorial, año de publicación. Estos libros se pueden prestar a los alumnos, salvo excepciones (diccionarios, normas ISO, etc., que sólo se pueden consultar en la sala).

Las revistas en papel que tienen las mismas características que los libros, más algunas peculiaridades: volumen, número y mes de salida. Se pueden consultar y prestar a los alumnos.

Documentos en formato CD (libros, software). Se pueden prestar, al igual que los libros. En este caso interesa mantener algún dato más (formato del CD, tipo de licencia).

Revistas de investigación microfilmadas que tienen las mismas características que las revistas en papel pero no se prestan y sólo se pueden consultar en la sala mediante terminales. Como dato adicional hay que mantener el código de microfilm.

Diseñe las clases y relaciones que representen una solución para este problema. Se pide en concreto, la estructura de herencia implicada con el detalle de características atribuidas a cada clase y sus posibles redefiniciones. Escriba en Java la clase origen de la jerarquía de documento y la interfaz de la clase que incluya los procedimientos de préstamo y consulta en sala.

15. Una compañía de promociones turísticas desea mantener información sobre la infraestructura de alojamiento para turistas, de forma tal que los clientes puedan planear sus vacaciones de acuerdo a sus posibilidades. Los alojamientos se identifican por un nombre, una dirección, una localidad y un gerente encargado del lugar. La compañía trabaja con dos tipos de alojamientos: Hoteles y Alojamientos Extrahoteleros. Los Hoteles pueden ser de tres, cuatro o cinco estrellas. Las características de las distintas categorías son las siguientes:

Hotel *** Cantidad de Habitaciones, Número de Camas, Cantidad de Pisos, Precio de las Habitaciones.

Hotel **** Cantidad de Habitaciones, Número de camas, Cantidad de Pisos, Gimnasio, Nombre del Restaurante, Capacidad del Restaurante, Precio de las Habitaciones.

Hotel ***** Cantidad de Habitaciones, Número de camas, Cantidad de Pisos, Gimnasio, Nombre del Restaurante, Capacidad del Restaurante, Cantidad Salones de Conferencia, Cantidad de Suites, Cantidad de Limosinas, Precio de las Habitaciones.

Los gimnasios pueden ser clasificados por la empresa como de tipo "A" o de tipo "B", de acuerdo a las prestaciones observadas. Las limosinas están disponibles para cualquier cliente, pero sujeto a disponibilidad, por lo que cuanto más limosinas tenga el hotel, más caro será. El precio de una habitación debe calcularse de acuerdo a la siguiente fórmula:

$$\text{PrecioHabitación} = \$50 + (\$1 \times \text{capacidad del hotel}) + (\text{valor agregado por restaurante}) +$$

$$+ (\text{valor agregado por gimnasio}) + (\text{valor agregado por limosinas}).$$

en donde:

Valor agregado por el restaurante:

- 10\$ si la capacidad del restaurante es de menos de 30 personas.
- 30\$ si está entre 30 y 50 personas.
- 50\$ si es mayor de 50.

Valor agregado por el gimnasio:

- 50\$ si el tipo del gimnasio es A.
- 30\$ si el tipo del gimnasio es B.

Valor agregado por las limosinas:

- 15\$ por la cantidad de limosinas del hotel.

En contraste, los Alojamientos Extrahoteleros proveen servicios diferentes a los de los hoteles, estando más orientados a la vida al aire libre y al turista de bajos recursos. Por cada Alojamiento Extrahotelero se indica si es privado o no, así como la cantidad de metros cuadrados que ocupa. Existen dos tipos de alojamientos extrahoteleros: los Camping y las Residencias. Para los Camping se indica la capacidad máxima de carpas, la cantidad de baños disponibles y si posee o no un restaurante dentro de las instalaciones. Para las residencias se indica la cantidad de habitaciones, si se hacen o no descuentos a los gremios y si posee o no campo deportivo.

Realizar un diagrama de clases en el que se representen todas las relaciones descriptas.

Implementar el diagrama y un sistema de consulta que le permite al usuario consultar por diferentes criterios:

- todos los alojamientos
- todos los hoteles de una determinada localidad
- todos los campings de una determinada localidad