

Test Tema 7: Herencia y Polimorfismo.

1. Dadas las siguientes clases :

```
public interface Saltarin {}  
  
public class Rana implements Saltarin { }  
  
public class RanaPortuguesa extends Rana {}  
  
public class RanaBrasilenha extends Rana {}
```

Cuál o cuáles de las siguientes sentencias pueden ser insertadas en la línea para que el código compile:

Rana rana = new RanaPortuguesa();

- A. Object
- B. Saltarin
- C. RanaPortuguesa
- D. Rana
- E. RanaBrasilenha



2. Indica cuál de las siguientes afirmaciones sobre polimorfismo son ciertas:

- a. Una referencia de un objeto puede ser convertida (cast) a una sub clase sin realizar un cast explícito.
- b. Si un método recibe como parámetro una super clase de 3 objetos , entonces cualquier de esas 3 clases puede ser pasada como argumento.
- c. Todas las excepciones de tipo cast pueden ser detectadas en tiempo de compilación.
- d. Un método que recibe una referencia de tipo Object podrá recibir cualquier referencia.



3. Cuáles de los siguientes modificadores son asumidos por todas las variables de una interfaz:

a. public

b. static

c. private

d. final

e. protected

f. abstract



4. Indica que mensaje se verá por pantalla:

```
public class Artropodo {

    public void imprime(){
        System.out.println("Soy un artrópodo");
    }
}

public class Aranha extends Artropodo{

    public void imprime(){

        System.out.println("Soy una araña");

    }
}

public class Principal {

    public static void main(String []args){

        Aranha a = new Aranha();
        Artropodo art = new Aranha();

        a.imprime();
        art.imprime();

    }
}
```

- a. Soy una araña , Soy una araña
- b. Soy una araña , Soy un artrópodo
- c. Soy un artrópodo , Soy un artrópodo
- d. Soy un artrópodo , Soy una araña.



5. Dadas las siguientes clases :

```
public class Vehiculo{

    public void acelerar(int x){

        System.out.println("Aumento Velocidad vehículo : " + x);
    }

}

public class Coche extends Vehiculo {

    public void acelerar(double x){
        System.out.println("Aumento Velocidad Coche: " + x);
    }

}

public class Moto extends Vehiculo{

    public void acelerar(double x){
        System.out.println("Aumento Velocidad Moto: " + x);
    }

}

public class Principal {
    public static void main(String []args){

        Vehiculo v = new Coche();
        Vehiculo v2 = new Moto();

        v.acelerar(10);
        v2.acelerar(10.4);

    } }

```

Indica cuál sería la salida :

- "Aumento Velocidad **vehículo** : 10" , "Aumento Velocidad **vehículo** : 10.4"
- "Aumento Velocidad **coche**: 10" , "Aumento Velocidad **moto** 10.4: "
- "Aumento Velocidad **vehículo** : 10 , "Aumento Velocidad **moto** 10.4: "
- Dará un error de compilación ya que no encontrará el método **acelerar(double x);**



6. Dadas las siguientes clases:

```
public abstract class AnimalMarino{

    public abstract void bucear();
}

public class Ballena extends AnimalMarino{

    public void bucear(double x){
        System.out.println("Ballena buceando a: " + x + " metros");
    }
}

public class Principal {

    public static void main(String []args) {

        Ballena b = new Ballena();
        b.bucear(200);

    }
}
```

¿Cuál sería la salida del programa:?

- Ballena buceando a 200 metros
- Dará un error de compilación ya que el método bucear abstracto no está siendo implementado
- Dará un error en tiempo de ejecución ya que no encuentra el método bucear(int x)



- d. Dará un error en tiempo de compilación porque una referencia Ballena no puede encontrar el método bucear(int x)

7. Dadas las siguientes clases:

```
public class Vehiculo{ }

public class Coche extends Vehiculo { }

public class Moto { }

public class VehiculoHandler {

    private Vehiculo;

    public void setVehiculo (Vehiculo v){

        this.vehiculo = v;

    }

}

public class Principal {
    public static void main(String []args){

        new VehiculoHandler().setVehiculo(_____)

    }

}
```

Cuáles de las siguientes opciones puede escribirse en la línea para que compile correctamente :

- a. Object
- b. new Coche(); ✓
- c. new Moto(); ✓
- d. new String();
- e. null; ✓
- f. new Vehiculo(); ✓

✗