

TAREA 7

Nombre: _____

1.- Complete las siguientes oraciones con las palabras que se encuentran en el recuadro:

Concretas	<i>abstract</i>	Polimorfismo	Abstracta
-----------	-----------------	--------------	-----------

- a) Si una clase contiene al menos un método abstracto, es una clase _____.
- b) Las clases a partir de las cuales pueden instanciarse objetos se llaman clases _____.
- c) El _____ implica el uso de una variable de referencia del tipo de la superclase para invocar métodos en objetos de superclase y subclase, lo cual nos permite "Programar en general".
- d) Los métodos que no son métodos de interfaz y que no proporcionan implementaciones deben declararse utilizando la palabra clave _____.

2.- Conteste verdadero o Falso a cada una de las siguientes proposiciones; en caso de ser falso, explique por qué.

- a) Todos los métodos en una clase *abstract* deben declararse como métodos *abstract*.

- b) No está permitido invocar a un método que solo pertenece a una subclase, a través de una variable de referencia de la subclase.

- c) Si una superclase declara a un método como *abstract*, toda subclase debe implementar a ese método.

3.- Dadas las siguientes declaraciones:

```
class A {}  
abstract class B {}  
class C extends B {}  
class D extends A {}  
class E extends C {}  
class F extends E {}
```

1. A no es una clase abstracta.
2. C hereda de la clase A.
3. No se pueden crear instancias de la clase B.
4. F hereda de la clase B.
5. F hereda de la clase A.
6. D hereda de B.

¿Cuáles afirmaciones son ciertas?

- a) 1, 3 y 4
- b) 1,2,3, y 5
- c) 1 y 5
- d) Todas

4.- ¿Identifica el error en el siguiente código?

```
1 | abstract class AbstractClass  
2 | {  
3 |     private abstract int abstractMethod();  
4 | }
```

5.- ¿Qué clase es instanciable? ¿Clase A o Clase B?

```
1 | abstract class A  
2 | {  
3 |  
4 | }  
5 |  
6 | class B extends A  
7 | {  
8 |  
9 | }
```

6.- ¿El fragmento de código siguiente muestra un error de compilación? ¿Puede sugerir las correcciones?

```
1  abstract class A
2  {
3      abstract int add(int a, int b);
4  }
5
6  class B extends A
7  {
8
9  }
```

7.- ¿Es correcto declarar métodos protegidos en una interfaz?

8.- ¿Podemos escribir constructores explícitos en una clase abstracta?

9.- ¿Cuál será la salida del siguiente programa?

```
abstract class A
{
    abstract void firstMethod();

    void secondMethod()
    {
        System.out.println("SECOND");

        firstMethod();
    }
}

abstract class B extends A
{
    @Override
    void firstMethod()
    {
        System.out.println("FIRST");

        thirdMethod();
    }

    abstract void thirdMethod();
}

class C extends B
{
    @Override
    void thirdMethod()
    {
        System.out.println("THIRD");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        C c = new C();

        c.firstMethod();

        c.secondMethod();

        c.thirdMethod();
    }
}
```