

Actividades Polimorfismo

Completar el código

Objetivo:

- Que los alumnos apliquen y demuestren su comprensión del concepto de polimorfismo en programación orientada a objetos, desarrollando la habilidad de implementar este principio de manera efectiva al completar y adaptar código proporcionado para que funcione correctamente.

Instrucciones:

- Coloca en los espacios vacíos la palabra clave para que el código funcione correctamente:

```
1  // Clase base (superclase)
2  class Animal {
3      void hacerSonido() {
4          System.out.println("El animal hace un sonido");
5      }
6  }
7
8  // Subclase 1
9  class Perro  Animal {
10     @Override
11     void hacerSonido() {
12         System.out.println("El perro ladra");
13     }
14 }
15
16 // Subclase 2
17 class Gato  Animal {
18     @Override
19     void hacerSonido() {
20         System.out.println("El gato maulla");
21     }
22 }
23
24 // Subclase 3
25 class Vaca  Animal {
26     @Override
27     void hacerSonido() {
28         System.out.println("La vaca muge");
29     }
30 }
```

La respuesta correcta sería: “Extends”, ya que gracias a esa palabra sabemos que está heredando a la superclase “Animal”, y la función “hacerSonido” podrá utilizarse en las subclases. Esto dando a entender que el Perro es un Animal y emite un sonido, así como el gato, pero aunque son sonidos, son distintos, uno ladra mientras el otro maulla.

El código se vería de la siguiente manera:

```
1  // Clase base (superclase)
2  class Animal {
3      void hacerSonido() {
4          System.out.println("El animal hace un sonido");
5      }
6  }
7
8  // Subclase 1
9  class Perro extends Animal {
10     @Override
11     void hacerSonido() {
12         System.out.println("El perro ladra");
13     }
14 }
15
16 // Subclase 2
17 class Gato extends Animal {
18     @Override
19     void hacerSonido() {
20         System.out.println("El gato maulla");
21     }
22 }
23
24 // Subclase 3
25 class Vaca extends Animal {
26     @Override
27     void hacerSonido() {
28         System.out.println("La vaca muge");
29     }
30 }
```

Identificar código.

Objetivo

- Que los alumnos apliquen y demuestren su comprensión del concepto de polimorfismo en programación orientada a objetos.

Instrucciones:

Siguiendo el código anterior la variable 'miAnimal' se declara un objeto de tipo Animal, pero en tiempo de ejecución, se le asignan instancias de las subclases Perro, Gato y Vaca. A través del polimorfismo, la llamada al método hacerSonido() se resuelve dinámicamente en función del tipo de objeto real al que se refiere la variable en ese momento, lo que permite que cada objeto haga su propio sonido. Esto es un ejemplo de polimorfismo.

```
public class Main {  
    public static void main(String[] args) {  
        Animal miAnimal;  
  
        miAnimal = new Perro();  
        miAnimal.hacerSonido();  
  
        miAnimal = new Gato();  
        miAnimal.hacerSonido();  
  
        miAnimal = new Vaca();  
        miAnimal.hacerSonido();  
    }  
}
```

Teniendo esto en cuenta, ¿Cuál sería la salida de este código?

Respuestas:

- El Perro Ladra.
- El Gato Maulla.
- La Vaca Muge.

Cuestionario

1. ¿Qué es el polimorfismo en programación orientada a objetos?
 - a) Una técnica para ocultar datos.
 - b) Una técnica para organizar el código.
 - c) Una técnica para tratar objetos de diferentes clases de manera uniforme
2. ¿Cuál de las siguientes palabras clave se usa en Java para permitir el polimorfismo?
 - a) polymorph
 - b) override
 - c) extends
3. ¿Cuál es el beneficio principal del polimorfismo en la programación?
 - a) Aumenta la complejidad del código.
 - b) Facilita la reutilización del código.
 - c) Hace que los objetos sean menos flexibles.