

Introducción a Ciencias de la Computación (I)

Pérez Servin Darshan Israel

Tarea 6: Herencia

Fecha de entrega: Miércoles 22 de Noviembre del 2023

1. **Explica qué significa que un atributo o método tenga visibilidad `protected`.**

Un atributo o método que sea `protected` puede ser visible solo dentro de la clase y en sus clases heredadas, a comparación del `private` que solo es visible dentro de la clase padre.

Por ejemplo si se crea un método o atributo en una clase y de esta se hereda, este mismo puede ser usado en la clase heredada sin necesidad de alterar lo que se encuentra en la clase “padre”.

2. **Indica la diferencia entre la sobrecarga de un método y la redefinición de un método.**

Sobrecarga de Método:

Una sobrecarga de método Es tener varios métodos con el mismo nombre en una sola clase, aunque sean diferentes en lo que hagan.

Redefinición de Método

Una redefinición es cuando se crea un método en una clase padre y se hereda en otra clase bajo el mismo nombre implementando nuevas funciones sobre el anterior.

3. **Indica cuáles atributos y métodos de una clase pueden ser heredados a otra.**

Los atributos y métodos que se pueden heredar de una clase a otra son aquellos que se encuentran en `protected` y en `public`. No se pueden heredar aquellos que se encuentren como `final` o en `static`.

4. **Menciona de qué manera se relaciona la redefinición de métodos con el polimorfismo.**

La redefinición de métodos, ocurre cuando una clase hija proporciona una implementación nueva y específica para un método que ya está definido en su clase padre. Usualmente para evitar errores en el compilador se opta por usar la sintaxis “`@Override`” antes del método. Esto se lleva de la

mano con el polimorfismo, ya que este se refiere cuando un método que se redefine con una firma idéntica o compatible a su método de la clase base, de manera de que se puede invocar de dos formas diferentes. Como por ejemplo estos dos objetos serían distintos y por ende si se invoca con el mismo método, actuaría según que hace el método de la clase:

CÓDIGO EJEMPLO:

```
ClasePadre objeto1 = new ClasePadre();
ClasePadre objeto1 = new ClaseHija();

objeto1.metodo();
//Hace lo que se encuentre en la clase padre
objeto2.metodo();
//Hace lo que se encuentre en la clase hija
```

5. Menciona una diferencia y una similitud entre clases abstractas e interfaces.

Diferencias:

En una clase abstracta se puede almacenar métodos ya sean abstractos o concretos.

En una interfaz SOLO se pueden almacenar métodos abstractos y variables finales.

Similitud:

Ambas se usan para de alguna manera "heredar" métodos que no están definidos aún, una buena guía para saber que métodos se pueden usar al heredarse.