

06/06/2025

Trabajo Práctico 6– Polimorfismo

Material de lectura

Java. Cómo programar. Deitel

<https://drive.google.com/file/d/1IRDQWeozOz6eJ0PngO8ue2SnfwpEPMHq/view?usp=sharing>

Complete las siguientes oraciones:

- 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 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 _____.
- e) Al proceso de convertir una referencia almacenada en una variable de una superclase a un tipo de una subclase se le conoce como _____.

Conteste con *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 abstracta deben declararse como métodos abstractos.
- b) No está permitido invocar a un método que sólo pertenece a una subclase, a través de una variable de subclase.
- c) Si una superclase declara a un método como abstracto, una subclase debe implementar a ese método.
- d) Un objeto de una clase que implementa a una interfaz puede considerarse como un objeto de ese tipo de interfaz.

10.3 ¿Cómo es que el polimorfismo le permite programar “en forma general”, en lugar de hacerlo “en forma específica”? Hable sobre las ventajas clave de la programación “en forma general”.

10.4 ¿Qué son los métodos abstractos? Describa las circunstancias en las que un método abstracto sería apropiado.

10.5 ¿Cómo es que el polimorfismo fomenta la extensibilidad?

10.6 Describa cuatro formas en las que podemos asignar referencias de superclases y subclases a variables de los tipos de las superclases y las subclases.

10.7 Compare y contraste las clases abstractas y las interfaces. ¿Para qué podría usar una clase abstracta? ¿Para qué podría usar una interfaz?

10.13 (Interfaz ImpactoEcológico: polimorfismo) Mediante el uso de interfaces, como aprendió en este capítulo, es posible especificar comportamientos similares para clases que pueden ser dispares. Los gobiernos y las compañías en todo el mundo se están preocupando cada vez más por el impacto ecológico del carbono (las liberaciones anuales de dióxido de carbono en la atmósfera), debido a los edificios que consumen diversos tipos de combustibles para obtener calor, los vehículos que queman combustibles para producir energía, y demás. Muchos científicos culpan a estos gases de invernadero por el fenómeno conocido como calentamiento global. Cree tres pequeñas clases no relacionadas por herencia: las clases Edificio, Auto y Bicicleta. Proporcione a cada clase ciertos atributos y comportamientos apropiados que sean únicos, que no tengan en común con otras clases. Escriba la interfaz ImpactoEcológico con un método obtenerImpactoEcológico. Haga que cada una de sus clases implementen a esa interfaz, de modo que su método obtenerImpactoEcológico calcule el impacto ecológico del carbono apropiado para esa clase (consulte sitios Web que expliquen cómo calcular el impacto ecológico del carbono). Escriba una aplicación que cree objetos de cada una de las tres clases, coloque referencias a esos objetos en ArrayList<ImpactoEcológico> y después itere a través del objeto ArrayList, invocando en forma polimórfica el método obtenerImpactoEcológico de cada objeto. Para cada objeto imprima cierta información de identificación, además de su impacto ecológico.