

Practica de clases abstractas

Abstracción:

Todos los lenguajes de programación proporcionan abstracciones. Se puede argumentar que la complejidad de los problemas que puedes resolver está directamente relacionada con el tipo y la calidad de la abstracción. Un elemento esencial de la programación orientada a objetos es una abstracción. Para obtener más información, lee este artículo.

1. Escribe un programa Java para crear una clase abstracta `Animal` con un método abstracto llamado `sound()`. Crea las subclases `Lion` y `Tiger` que extiendan la clase `Animal` e implementan el método `sound()` para hacer un sonido específico para cada animal.

[Haz clic aquí para ver la solución](#)

2. Escribe un programa Java para crear una clase abstracta `Shape` con los métodos abstractos `calculateArea()` y `calculatePerimeter()`. Crea las subclases `Circle` y `Triangle` que extiendan la clase `Shape` e implementa los métodos respectivos para calcular el área y el perímetro de cada forma.

[Haz clic aquí para ver la solución](#)

3. Escribe un programa Java para crear una clase abstracta `BankAccount` con los métodos abstractos `deposit()` y `retreat()`. Cree las subclases `SavingsAccount` y `CurrentAccount` que extiendan la clase `BankAccount` e implementen los métodos respectivos para manejar depósitos y retiros para cada tipo de cuenta.

[Haga clic aquí para ver la solución](#)

4. Escriba un programa Java para crear una clase abstracta Animal con los métodos abstractos eat() y sleep(). Cree las subclases Lion, Tiger y Deer que extiendan la clase Animal e implementen los métodos eat() y sleep() de manera diferente según su comportamiento específico.

[Haga clic aquí para ver la solución](#)

5. Escriba un programa Java para crear una clase abstracta Employee con los métodos abstractos calculateSalary() y displayInfo(). Cree las subclases Manager y Programmer que extiendan la clase Employee e implementen los métodos respectivos para calcular el salario y mostrar información para cada función.

[Haga clic aquí para ver la solución](#)

6. Escriba un programa Java para crear una clase abstracta Shape3D con los métodos abstractos calculateVolume() y calculateSurfaceArea(). Cree las subclases Sphere y Cube que extiendan la clase Shape3D e implementen los métodos respectivos para calcular el volumen y el área de superficie de cada forma.

[Haz clic aquí para ver la solución](#)

7. Escriba un programa Java para crear una clase abstracta Vehicle con los métodos abstractos startEngine() y stopEngine(). Cree las subclases Car y Motorcycle que extiendan la clase Vehicle e implementen los métodos respectivos para encender y apagar los motores de cada tipo de vehículo.

[Haz clic aquí para ver la solución](#)

8. Escriba un programa Java para crear una clase abstracta Person con los métodos abstractos eat() y exercise(). Cree las subclases Athlete y LazyPerson que extiendan la clase Person e implementen los métodos respectivos para describir cómo come y hace ejercicio cada persona.

[Haz clic aquí para ver la solución](#)

9. Escribe un programa Java para crear una clase abstracta Instrument con los métodos abstractos play() y tune(). Crea subclases para Glockenspiel y Violin que extiendan la clase Instrument e implementan los métodos respectivos para tocar y afinar cada instrumento.

[Haz clic aquí para ver la solución](#)

10. Escribe un programa Java para crear una clase abstracta Shape2D con los métodos abstractos draw() y resize(). Crea las subclases Rectangle y Circle que extiendan la clase Shape2D e implementa los métodos respectivos para dibujar y redimensionar cada forma.

[Haz clic aquí para ver la solución](#)

11. Escribe un programa Java para crear una clase abstracta Bird con los métodos abstractos fly() y makeSound(). Crea las subclases Eagle y Hawk que extiendan la clase Bird e implementa los métodos respectivos para describir cómo vuela y emite un sonido cada ave.

[Haz clic aquí para ver la solución](#)

12. Escribe un programa Java para crear una clase abstracta GeometricShape con los métodos abstractos area() y periods(). Crea las subclases Triangle y Square que extiendan la clase GeometricShape e implementa los métodos respectivos para calcular el área y el perímetro de cada forma.

[Haz clic aquí para ver la solución](#)