Primer Parcial Programación con Objetos 1 **UNO**
Fecha: 06-10-2025 (Turno Mañana)

Alumno:

Una empresa de transporte desea un sistema para gestionar sus vehículos y conductores.

1 - Encapsulamiento: Define una clase base `Vehiculo` con los atributos privados: `marca` y `patente(strings)`. Agrega getters y setters solo cuando sea necesario. Validar que los atributos no estén vacíos.

2 - Herencia y Polimorfismo: Crea dos subclases de `Vehiculo`: `Auto` (tiene un atributo adicional: `cantidad_puertas`). `Moto` (tiene un atributo adicional: `cilindrada`). En `Vehiculo` define un método `calcular_costo_mantenimiento()` que devuelva un valor base de `1000`.
Sobrescribe el método en las subclases:

`Auto`: `valor_base + 200 * cantidad_puertas`.
En `Moto`: `valor_base + 0.5 * cilindrada`.

3 - Composición: Define una clase `Conductor` con atributos privados `nombre` y `dni`. Un `Conductor` tiene un `Vehiculo`. Agrega un método `mostrar_informacion()` que imprima el nombre del conductor, los datos básicos de su vehículo y el costo de mantenimiento (usando polimorfismo).

4 - Prueba del sistema Crea al menos **un Auto y una Moto**, cada uno con su Conductor. Muestra la información de cada conductor usando `mostrar_informacion()`.

5 - Indique si las siguientes afirmaciones son **Verdaderas (V)** o **Falsas (F)**. Justifique en 1 o 2 líneas.

____ El **encapsulamiento** busca ocultar los detalles internos de una clase y controlar el acceso a sus atributos mediante métodos definidos por el programador.

____ Una subclase siempre está obligada a sobrescribir todos los métodos de su superclase.

____ El **polimorfismo** permite que distintas clases respondan de manera diferente a un mismo mensaje o método, siempre que compartan una relación de herencia.

____ El **polimorfismo** requiere que el código cliente conozca la subclase concreta de cada objeto para poder invocar el método sobrescrito correspondiente.

____ Una colección que almacena objetos de una superclase puede contener también instancias de sus subclases, y al invocar métodos se ejecuta el comportamiento correspondiente al tipo real del objeto.

Requisitos:
- Código bien estructurado y comentado.
- Implementado en Java o Python.
- No debe demorar más de 1:30 horas en resolverse.
- 60% de la parte practica correcta para aprobar.
- 4 de 5 preguntas teóricas correctas ademas del 70% de la pate práctica, para acceder a la promoción.

Primer Parcial Programación con Objetos 1 **UNO**
Fecha: 06-10-2025 (Turno Mañana)

Alumno:

Una empresa de transporte desea un sistema para gestionar sus vehículos y conductores.

1 - Encapsulamiento: Define una clase base `Vehiculo` con los atributos privados: `marca` y `patente(strings)`. Agrega getters y setters solo cuando sea necesario. Validar que los atributos no estén vacíos.

2 - Herencia y Polimorfismo: Crea dos subclases de `Vehiculo`: `Auto` (tiene un atributo adicional: `cantidad_puertas`). `Moto` (tiene un atributo adicional: `cilindrada`). En `Vehiculo` define un método `calcular_costo_mantenimiento()` que devuelva un valor base de `1000`.
Sobrescribe el método en las subclases:

`Auto`: `valor_base + 200 * cantidad_puertas`.
En `Moto`: `valor_base + 0.5 * cilindrada`.

3 - Composición: Define una clase `Conductor` con atributos privados `nombre` y `dni`. Un `Conductor` tiene un `Vehiculo`. Agrega un método `mostrar_informacion()` que imprima el nombre del conductor, los datos básicos de su vehículo y el costo de mantenimiento (usando polimorfismo).

4 - Prueba del sistema Crea al menos **un Auto y una Moto**, cada uno con su Conductor. Muestra la información de cada conductor usando `mostrar_informacion()`.

5 - Indique si las siguientes afirmaciones son **Verdaderas (V)** o **Falsas (F)**. Justifique en 1 o 2 líneas.

____ El **encapsulamiento** busca ocultar los detalles internos de una clase y controlar el acceso a sus atributos mediante métodos definidos por el programador.

____ Una subclase siempre está obligada a sobrescribir todos los métodos de su superclase.

____ El **polimorfismo** permite que distintas clases respondan de manera diferente a un mismo mensaje o método, siempre que compartan una relación de herencia.

____ El **polimorfismo** requiere que el código cliente conozca la subclase concreta de cada objeto para poder invocar el método sobrescrito correspondiente.

____ Una colección que almacena objetos de una superclase puede contener también instancias de sus subclases, y al invocar métodos se ejecuta el comportamiento correspondiente al tipo real del objeto.

Requisitos:
- Código bien estructurado y comentado.
- Implementado en Java o Python.
- No debe demorar más de 1:30 horas en resolverse.
- 60% de la parte practica correcta para aprobar.
- 4 de 5 preguntas teóricas correctas ademas del 70% de la pate práctica, para acceder a la promoción.