CESUR
Tu Centro Oficial de FP

EXAMEN PARCIAL – UNIDADES 3 Y 4		
TRIMESTRE: PRIMERO		**Fecha:**
CICLO: Desarrollo de Aplicaciones Web.	*CURSO: 1º*	*CALIFICACIÓN:*
MÓDULO: Programación	*Turno: Mañana*	
Nombre:	***Apellidos:***	

Instrucciones: *Esta prueba tiene como finalidad evaluar los aprendizajes de Programación. Lee atentamente y responde escribiendo el código más adecuado.*

Si las instrucciones no se siguen como se especifican el examen no será evaluado

PARTE PRÁCTICA. TIPO B.

- El examen práctico tiene una puntuación máxima de 10 puntos.
- Para superar la parte práctica se requiere alcanzar un mínimo de 5 puntos.

CESUR
Tu Centro Oficial de FP

1. **(3 puntos)** Desarrolla un sistema en Java que simule una calculadora básica y una calculadora científica:

Clase Calculadora

- **Atributos:**
  - No es necesario tener atributos en esta clase para este caso.
- **Métodos:**
  - **Método add(int a, int b)**: Suma dos números enteros.
  - **Método add(double a, double b)**: Suma dos números de tipo double.
  - **Método add(int a, int b, int c)**: Sobrecarga del método add para sumar tres números enteros.

Subclase CalculadoraCientifica

- **Métodos adicionales:**
  - **Método add(int a, int b)**: Sobrescribe el método de la clase base para realizar una suma avanzada, por ejemplo, sumando el valor de la raíz cuadrada de los números antes de realizar la suma.
  - **Método raizCuadrada(double numero)**: Calcula la raíz cuadrada de un número.

Probar con este main:

```
public static void main(String[] args) {
    // Instancia de la clase base Calculadora
    Calculadora calc = new Calculadora();
    System.out.println("Suma de enteros (Calculadora): " +
                                                    calc.add(2, 3));
    System.out.println("Suma de doubles (Calculadora): " +
                                                    calc.add(2.5, 3.7));
    System.out.println("Suma de tres enteros (Calculadora): " +
                                                    calc.add(1, 2, 3));

    // Instancia de la clase derivada CalculadoraCientifica
    CalculadoraCientifica calcCientifica = new
                                            CalculadoraCientifica();
    System.out.println("Suma avanzada (CalculadoraCientifica): " +
                                            calcCientifica.add(2, 3));
    System.out.println("Raíz cuadrada (CalculadoraCientifica): " +
                                    calcCientifica.raizCuadrada(16));
}
```

Ejemplo:

```
Suma de enteros (Calculadora): 5
Suma de doubles (Calculadora): 6.2
Suma de tres enteros (Calculadora): 6
Suma avanzada (CalculadoraCientifica): 25
Raíz cuadrada (CalculadoraCientifica): 4.0
```

2. **(3 puntos)** Desarrolla un sistema en Java que simule el comportamiento alimenticio de diferentes tipos de animales:

   Clase Animal

   - **Atributos**:
     - nombre (String): Nombre del animal.
   - **Métodos**:
     - **eat()**: Método genérico que representa que el animal está comiendo (comportamiento común para todos los animales).
     - **eat(String food)**: Sobrecarga de eat que recibe un parámetro de tipo String representando el tipo de alimento que el animal está comiendo.

   Subclase Carnivoro (Hereda de Animal)

   - Representa a los animales carnívoros.
   - **Método sobrescrito eat()**:
     - Debe imprimir que el animal come carne, por ejemplo: *"El león está comiendo carne."*.
   - **Sobrecarga del método eat(String food)**:
     - Si el alimento es carne, debe imprimir *"El león está comiendo carne."*.
     - Si el alimento es cualquier otro tipo (como "hierba"), debe imprimir *"El león no puede comer este tipo de alimento."*

   Subclase Herbivoro (Hereda de Animal)

   - Representa a los animales herbívoros.
   - **Método sobrescrito eat()**:
     - Debe imprimir que el animal está comiendo vegetales, por ejemplo: *"El conejo está comiendo vegetales."*.
   - **Sobrecarga del método eat(String food)**:
     - Si el alimento es hierba, debe imprimir *"El conejo está comiendo hierba."*.
     - Si el alimento es carne, debe imprimir *"El conejo no puede comer carne."*.

   Subclase Omnivoro (Hereda de Animal)

   - Representa a los animales omnívoros.
   - **Método sobrescrito eat()**:
     - Debe imprimir que el animal come tanto carne como vegetales, por ejemplo: *"El oso está comiendo carne y vegetales."*.

- **Sobrecarga del método eat(String food)**:
  - Si el alimento es carne, debe imprimir *"El oso está comiendo carne."*.
  - Si el alimento es hierba o frutas, debe imprimir *"El oso está comiendo vegetales."*.
  - Si el alimento es otro, debe imprimir *"El oso no puede comer este tipo de alimento."*.

Prueba con este main:

```
public static void main(String[] args) {
    Carnivoro leon = new Carnivoro("Leon");
    Herbivoro conejo = new Herbivoro("Conejo");
    Omnivoro oso = new Omnivoro("Oso");
    System.out.println("Comportamiento general:");
    leon.eat();
    conejo.eat();
    oso.eat();
    System.out.println("\nComportamiento específico con
                                                    sobrecarga:");
    leon.eat("carne");
    leon.eat("hierba");
    conejo.eat("hierba");
    conejo.eat("carne");
    oso.eat("frutas");
    oso.eat("carne");
}
```

Ejemplo:

```
Comportamiento general:
El carnívoro está comiendo carne.
El herbívoro está comiendo plantas.
El omnívoro está comiendo una dieta variada.

Comportamiento específico con sobrecarga:
El carnívoro disfruta comiendo carne.
El carnívoro no puede comer hierba.
El herbívoro disfruta comiendo hierba.
El herbívoro no puede comer carne.
El omnívoro está comiendo frutas.
El omnívoro está comiendo carne.
```

3. **(4 puntos)** Desarrollar un programa en Java que modele un sistema de transporte.

Clase Vehiculo

- **Atributos:**
  - nombre (String): El nombre o tipo del vehículo (por ejemplo, "Genérico").
  - velocidad (int): La velocidad promedio del vehículo.
- **Métodos:**
  - **mover()**: Imprime un mensaje genérico indicando que el vehículo está en marcha.
  - **mover(int distancia)**: Sobrecarga del método para mostrar cuántos kilómetros recorrió el vehículo a una velocidad definida.
  - **mover(int distancia, String terreno)**: Sobrecarga adicional que incluye el tipo de terreno por el que se movió el vehículo.
- **Método toString()**:
  - Proporciona una representación en texto del vehículo, incluyendo su nombre y velocidad.

Subclase Coche

- **Atributos adicionales:**
  - modelo (String): El modelo del coche (por ejemplo, "Toyota").
- **Métodos sobrescritos:**
  - Sobrescribe los métodos mover() para personalizar los mensajes para los coches, indicando que el coche está en marcha.
  - Sobrescribe las versiones sobrecargadas de mover(int distancia) y mover(int distancia, String terreno) para incluir información del modelo.

Subclase Bicicleta

- **Atributos adicionales:**
  - tipo (String): El tipo de bicicleta (por ejemplo, "BMX").
- **Métodos sobrescritos:**
  - Sobrescribe el método mover() para personalizar el mensaje para las bicicletas.
  - Sobrescribe las versiones sobrecargadas de mover(int distancia) y mover(int distancia, String terreno) para incluir información del tipo de bicicleta.

Probar con este main:

```
public static void main(String[] args) {
    // Crear instancias
```

```
        Vehiculo vehiculo = new Vehiculo("Genérico", 50);
        Coche coche = new Coche("Toyota", 120);
        Bicicleta bicicleta = new Bicicleta("BMX", 15);

        // Mostrar información de cada vehículo
        System.out.println(vehiculo);
        System.out.println(coche);
        System.out.println(bicicleta);

        // Llamar a los métodos mover
        vehiculo.mover();
        vehiculo.mover(100);
        vehiculo.mover(50, "montañoso");

        coche.mover();
        coche.mover(200);
        coche.mover(120, "carretera");

        bicicleta.mover();
        bicicleta.mover(10);
        bicicleta.mover(5, "terreno llano");
    }
```

Ejemplo:

```
Vehiculo{nombre='Genérico', velocidad=50}
Vehiculo{nombre='Toyota', velocidad=120}
Vehiculo{nombre='BMX', velocidad=15}
El vehículo está en marcha.
El vehículo recorrió 100 kilómetros.
El vehículo recorrió 50 kilómetros en un terreno montañoso.
El coche está en marcha.
El vehículo recorrió 200 kilómetros.
El vehículo recorrió 120 kilómetros en un terreno carretera.
La bicicleta está en marcha.
El vehículo recorrió 10 kilómetros.
El vehículo recorrió 5 kilómetros en un terreno terreno llano.
```