

# Prueba 2

## Programación Orientada a Objetos – 28 de noviembre 2024

Nombre			RUT
Paralelo	( ) APaolini	( ) MMoraga	

### Antecedentes generales:

Puntaje total de la prueba	1 puntos	Puntaje Obtenido	
Puntos para nota aprobatoria (4.0)	0.6 puntos		
Duración de la prueba	3 horas	Nota final	
Resultados de Aprendizaje a evaluar	1. Aplicar técnicas de ingeniería de software en la creación de software legible, mantenable y testeable. 2. Aplicar técnicas de programación orientada al objeto en la resolución de problemas. 3. Crear tipos de datos abstractos con bajo acoplamiento entre la implementación y su comportamiento que permitan la resolución de problemas. 4. Analizar las relaciones causa efecto de los procesos en estudio. 6. Seleccionar los procesos, técnicas y herramientas adecuados de acuerdo a los requerimientos.		
Fecha de entrega de resultados	8 diciembre de 2024		

### Instrucciones:

1. Esta evaluación tiene 5 páginas (incluyendo la portada). Compruebe que dispone de todas las páginas.
2. Lea la prueba completamente **DOS** veces antes de hacer cualquier pregunta
3. Durante la prueba no se puede utilizar: teléfono móvil, apuntes. Está prohibido intentar conectarse a internet de cualquier manera (excepto a Campus Virtual, y solo para subir su solución). Si es sorprendido obtendrá la calificación mínima. Tampoco puede utilizar dispositivos de almacenamiento externos o cualquier otro dispositivo relojes inteligentes, ábacos, etc.
4. Una prueba respondida correctamente en un 60%, de acuerdo con las ponderaciones asignadas, corresponde a una nota 4,0.
5. La prueba es individual, cualquier sospecha de copia será calificada con la nota mínima y el caso será remitido al comité de ética.
6. En su espacio personal no debe haber nada más que hojas de papel en blanco, lápiz, goma.
7. El resto de sus implementos debe guardarlos dentro de su mochila/bolso y ésta debe posicionarse al frente debajo de la pizarra. Si leyó hasta este punto, felicidades, para saber que lo hizo dibuje un pato yeco al final de esta página.
8. Los estudiantes quienes se les compruebe falta de honestidad académica o cualquier otro acto contrario a las normas de permanencia universitaria o al espíritu universitario, serán sancionados, según sea la gravedad de la falta, con medidas desde la amonestación verbal hasta la suspensión o pérdida de la condición de estudiante, los estudiantes expulsados no podrán volver a ingresar a ninguna carrera, programa o curso de la institución. El estudiante que incurriere en falta de honestidad, durante la realización de un proceso evaluativo, será calificado con la nota mínima 1,0.

Acepto las condiciones firmando: \_\_\_\_\_

## Problema 1. Ruteo (0.3 puntos)

Rutee el siguiente código. Para ahorrar espacio no se muestran las líneas "import".

```
6  public class Main {
7      public static void main(String[] args) {
8          Configurator configurator = Configurator.getInstance();
9          configurator.setStrategyBasedOnRut();
10         configurator.applyLayout();
11     }
12 }

14 class Configurator {
15     private static Configurator instance;
16     private LayoutStrategy strategy;
17
18     private Configurator() {
19     }
20
21     public static Configurator getInstance() {
22         if (instance == null) {
23             instance = new Configurator();
24         }
25         return instance;
26     }
27
28     public void setStrategyBasedOnRut() {
29         int fourthDigit = askForFourthRutDigit();
30
31         if (fourthDigit >= 0 && fourthDigit <= 5) {
32             strategy = new HorizontalLayoutStrategy();
33         } else if (fourthDigit >= 6 && fourthDigit <= 9) {
34             strategy = new VerticalLayoutStrategy();
35         } else {
36             System.out.println("Dígito inválido. Se usará layout horizontal por defecto.");
37             strategy = new HorizontalLayoutStrategy();
38         }
39     }
40
41     public void applyLayout() {
42         if (strategy == null) {
43             System.out.println("Estrategia no configurada. Usa setStrategyBasedOnRut() primero.");
44             return;
45         }
46
47         JFrame frame = new JFrame("Ventana con Layout");
48         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
49         frame.setSize(400, 400);
50
51         JPanel panel = strategy.createLayout();
52         frame.add(panel);
53
54         frame.setVisible(true);
55     }
56
57     private int askForFourthRutDigit() {
58         System.out.print("Ingresa tu RUT (sin puntos ni guión, ejemplo: 12345678-9): ");
59         String rut = new java.util.Scanner(System.in).nextLine();
60
61         if (rut.length() < 4) {
62             throw new IllegalArgumentException("El RUT debe tener al menos 4 dígitos.");
63         }
64
65         char fourthChar = rut.charAt(rut.length() - 5); // Obtiene el cuarto dígito (de derecha a izquierda antes del dígito
66         // verificador)
67         if (!Character.isDigit(fourthChar)) {
68             throw new IllegalArgumentException("El cuarto carácter debe ser un dígito.");
69         }
70         return Character.getNumericValue(fourthChar);
71     }
72 }
73 }
```

```
74 class HorizontalLayoutStrategy implements LayoutStrategy {  
75     public JPanel createLayout() {  
76         JPanel panel = new JPanel();  
77         panel.setLayout(new BoxLayout(panel, BoxLayout.X_AXIS));  
78         addButtons(panel);  
79         return panel;  
80     }  
81  
82     private void addButtons(JPanel panel) {  
83         for (int i = 1; i <= 5; i++) {  
84             panel.add(new JButton("Botón " + i));  
85         }  
86     }  
87 }  
88  
89 interface LayoutStrategy {  
90     JPanel createLayout();  
91 }  
92  
93 class VerticalLayoutStrategy implements LayoutStrategy {  
94     public JPanel createLayout() {  
95         JPanel panel = new JPanel();  
96         panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));  
97         addButtons(panel);  
98         return panel;  
99     }  
100  
101    private void addButtons(JPanel panel) {  
102        for (int i = 1; i <= 5; i++) {  
103            panel.add(new JButton("Botón " + i));  
104        }  
105    }  
106 }  
107
```

Dibuje acá la interfaz gráfica resultante al ejecutar el código y después de haber ingresado su RUT:

¿Al presionar qué botón aparece un mensaje por la pantalla?

Indique lo que sucede al presionar el botón “X” de la ventana principal de su aplicación:

## Problema 2. Rent a Car (0.7 puntos)

Una empresa de renta de vehículos está haciendo un sistema de seguimiento de su flota. Los tipos de vehículos que ofrecen son 3, Camionetas, SUV y Autos. De ellos se guarda la información del VIN (número identificador único de un vehículo), la marca, modelo y la eficiencia de combustible medida en km/l.

Todos los vehículos son automáticos y cuentan con 3 modos de manejo:

Modo Eco	En este modo la eficiencia de combustible es un 10% menor que la nominal.
Modo Normal	La eficiencia de combustible es la nominal.
Modo Sport	En este modo la eficiencia de combustible es un 25% superior a la nominal.

Se te entregará un archivo de texto llamado fleet.txt que tiene la siguiente estructura:

```
N  
VIN,Marca,Modelo,Tipo_Vehículo,Eficiencia  
...  
VIN,Marca,Modelo,Tipo_Vehículo,Eficiencia  
VIN,Modo_Manejo,Distancia_1,...,Distancia_N  
...  
VIN,Modo_Manejo,Distancia_1,...,Distancia_N
```

Esto significa que la primera línea indica cuantos vehículos tiene la flota, luego las N líneas siguientes contienen la información de la flota y luego el resto de líneas (cantidad variable) corresponden a usos de los vehículos. Estos usos serán en un modo de manejo especificado y la cantidad de distancias podrá variar, existiendo al menos una por registro. Cada una de las distancias indica un viaje realizado por un usuario. Se entenderá que las n distancias serán recorridas en el mismo modo de manejo.

Se necesita lo siguiente:

- Calcular el kilometraje recorrido de cada vehículo.
- Calcular cuánto combustible fue utilizado por cada vehículo. Considere que el modo de conducción influye en el gasto de combustible final.
- Vehículos con mayor consumo de combustible por categoría.

### Ejemplo de Ejecución

```
Dodge Ram 2500 CONSUMO: 664.56875 1 ODÓMETRO: 5020 km
Ford F250 CONSUMO: 715.4142857142857 1 ODÓMETRO: 4865 km
Ford F150 CONSUMO: 423.0944444444446 1 ODÓMETRO: 3703 km
Mitsubishi L200 CONSUMO: 387.8 1 ODÓMETRO: 3665 km
Mazda CX-7 CONSUMO: 300.1541666666667 1 ODÓMETRO: 3518 km
Ford Explorer CONSUMO: 451.183333333334 1 ODÓMETRO: 4109 km
Chevrolet TrailBlazer CONSUMO: 364.945 1 ODÓMETRO: 3659 km
Chevrolet Tahoe CONSUMO: 513.1624999999999 1 ODÓMETRO: 3925 km
BMW X6 CONSUMO: 179.855 1 ODÓMETRO: 1696 km
Audi Q5 CONSUMO: 189.47272727273 1 ODÓMETRO: 2189 km
Jeep Grand Cherokee CONSUMO: 234.22777777777776 1 ODÓMETRO: 1889 km
Chevrolet Aveo CONSUMO: 149.81 1 ODÓMETRO: 1968 km
Audi A6 CONSUMO: 345.5281250000005 1 ODÓMETRO: 5621 km
BMW 3 Series CONSUMO: 598.7749999999999 1 ODÓMETRO: 8132 km
Honda Civic CONSUMO: 432.1694444444465 1 ODÓMETRO: 7241 km
Honda Accord CONSUMO: 623.2291666666665 1 ODÓMETRO: 7231 km
Hyundai Sonata CONSUMO: 593.6269230769228 1 ODÓMETRO: 7314 km
Hyundai Accent CONSUMO: 453.65 1 ODÓMETRO: 7093 km
Kia Rio CONSUMO: 383.869444444444 1 ODÓMETRO: 6544 km
Mazda Mazda6 CONSUMO: 426.03823529411767 1 ODÓMETRO: 6955 km

Mayor consumo:
Camioneta: WAUHF68P16A397756 (715.4142857142857 litros)
SUV: 1C4RDJEG9DC098442 (513.1624999999999 litros)
Auto: WBAEB53578C709570 (623.2291666666665 litros)
```

### Debe entregar:

- Código Java (100%)

## Consideraciones:

- Considere que los vehículos de la flota deben estar almacenados en **una sola lista**.
- Debe **usar orientación al objeto**. Debe usar herencia.
- Debe utilizar 3 patrones, de manera obligatoria el patrón **Visitor** y **Strategy**, pero puede elegir entre **Singleton** o **Factory**.
- Obtendrá puntaje extra si la salida por pantalla aparece en una GUI.
- Use la arquitectura vista en clase: DEBE crear un “Sistema”.
- No se deben utilizar ciclos dentro de ciclos. Use funciones para hacerse la vida más fácil.
- El código fuente debe exportarlo como .zip y subirlo a Campus Virtual.



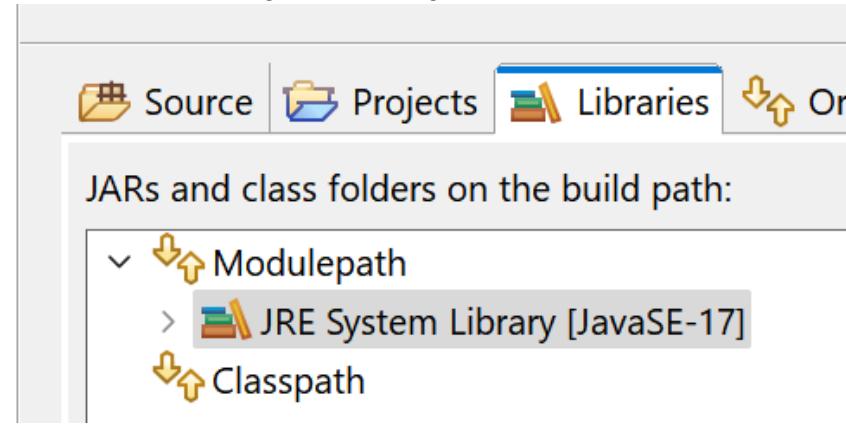
Use esta opción en el menú exportar > **Archive File**

- Archivos .txt de ejemplo se encuentran en CampusVirtual.
- Hojas sin nombre no se revisarán.
- Si el IDE no funciona correctamente haga clic derecho en el proyecto y luego en propiedades

## Properties

En **Java Build Path**, luego **Libraries** y luego doble clic en **JRE System Library**.

- > Resource Builders
- Coverage
- Java Build Path**
- > Java Code Style
- > Java Compiler
- Javadoc Location



Por último, seleccione Alternate JRE y aplique los cambios.

