

Prueba 2

Programación Orientada a Objetos – 22 de junio 2024

Nombre		RUT	
Paralelo	() APaolini () MMoraga () ERoss		

Antecedentes generales:

Puntaje total de la prueba Puntos para nota aprobatoria (4.0)	1 puntos 0.6 puntos	Puntaje Obtenido	
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, mantenible 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 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	5 julio 2024		

Instrucciones:

1. Esta evaluación tiene 6 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 una esfera 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)

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

```
public class Main
{
    public static void main(String []args)
    {
        Configurador.preguntar();

        JFrame frame = new JFrame("Prueba 2");

        frame.setDefaultCloseOperation(JFrame.DO_NOTHING_ON_CLOSE);
        frame.setSize(300, 300);

        frame.getContentPane().add(createGUI());

        frame.setVisible(true);
    }

    private static JPanel createGUI()
    {
        JPanel panel = MiFactory.getInstance().crearPanel();

        List<String> nombres = new ArrayList<>();

        nombres.add("B1");
        nombres.add("B2");
        nombres.add("B3");
        nombres.add("B4");
        nombres.add("B5");

        JButton b = null;
        for(String n : nombres) {
            b = new JButton(n);
            MiFactory.getInstance().agregarBoton(panel, b);
        }
        b.addActionListener(e -> {
            System.out.println("Hola, soy el botón");
        });

        return panel;
    }
}
```

```

public class Configurador
{
    public static void preguntar()
    {
        Scanner s = new Scanner(System.in);

        int a = preguntarInt(s, "Ingresa el primer dígito de tu RUT: ");
        int b = preguntarInt(s, "Ingresa el segundo dígito de tu RUT: ");
        int c = preguntarInt(s, "Ingresa el tercer dígito de tu RUT: ");
        int d = preguntarInt(s, "Ingresa el cuarto dígito de tu RUT: ");

        int t = a + 10*b + 100*c + 1000*d;
        int q = t % 2;
        MiFactory.configurar(q);
    }

    private static int preguntarInt(Scanner s, String string)
    {
        System.out.print(string);
        return Integer.parseInt(s.nextLine());
    }
}

public abstract class MiFactory
{
    private static MiFactory factory = null;

    protected MiFactory() { }

    public static void configurar(int num)
    {
        if ((num % 2) == 0) {
            factory = new MiFactory1();
        } else {
            factory = new MiFactory2();
        }
    }

    public static MiFactory getInstance() { return factory; }

    public abstract JPanel crearPanel();
    public abstract void agregarBoton(JPanel panel, JButton b);
}

public class MiFactory2 extends MiFactory
{
    public JPanel crearPanel()
    {
        JPanel p = new JPanel();
        p.setLayout(new BorderLayout(p, BorderLayout.PAGE_AXIS));
        return p;
    }

    public void agregarBoton(JPanel panel, JButton botón)
    {
        panel.add(botón);
    }
}

```

```
public class MiFactory1 extends MiFactory
{
    private int contador = 0;

    public JPanel crearPanel()
    {
        return new JPanel(new BorderLayout());
    }

    public void agregarBoton(JPanel panel, JButton button)
    {
        switch(contador) {
            case 0: panel.add(button, BorderLayout.NORTH); break;
            case 1: panel.add(button, BorderLayout.SOUTH); break;
            case 2: panel.add(button, BorderLayout.EAST); break;
            case 3: panel.add(button, BorderLayout.WEST); break;
            case 4: panel.add(button, BorderLayout.CENTER); break;
        }
        contador++;
        if (contador > 4) contador = 0;
    }
}
```

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. New Jurassic World (0.7 puntos)

Hace millones de años en la tierra existieron un montón de criaturas que hoy conocemos como dinosaurios. Pero los científicos han descubierto otro planeta relativamente cercano a nosotros donde algunas de estas criaturas todavía existen. También lograron ver que hay inteligencia alienígena rondando el planeta y analizando a estos dinosaurios (probablemente ellos llevaron algunas de las especies de la Tierra a ese planeta).

Estos alienígenas recorren en su nave el planeta y abducen a cada dinosaurio para saber ciertas estadísticas de ellos.

Nuestros científicos en la tierra pudieron interceptar las comunicaciones de la nave alienígena y obtuvieron la siguiente información:

Existen 4 tipos de dinosaurios: **Tiranosaurio**, **Triceratops**, **Velociraptor** y **Brachiosaurus**.

En general obtienen la siguiente información de cada uno:

- Peso (kg)
- Altura (m)
- Velocidad (km/h)
- Región donde viven (hemisferio norte/sur)

En particular cada especie tiene algunas características extra:

Especie	Característica
Tiranosaurio	Cicatrices (cantidad)
Triceratops	Planta Favorita (puede ser Helechos o Cicadáceas)
Velociraptor	Distancia recorrida (en km)
Brachiosaurus	Largo del cuello (en m)

Todo lo guardaron en un archivo llamado **fauna.txt** con el siguiente formato:

```
id_dinosaurio,especie,peso,altura,velocidad,región,característica_extra
```

En la Tierra quieren replicar lo que hacen los alienígenas y pretenden obtener la siguiente información:

- 1) Por hemisferio, cantidad de dinosaurios por cada especie.
- 2) Dinosaurio más pesado, más alto y más veloz (no necesariamente es el mismo).
- 3) Dinosaurio con mayor tasa de adaptación: Esto se calcula de manera diferente según la especie, pero siempre será el que tenga un valor más alto.

Tiranosaurio	$\frac{peso}{altura^{cicatrices}}$
Triceratops	Si come Helechos: $\frac{peso*velocidad}{2}$
	Si come Cicadáceas: $\frac{peso*velocidad}{3}$
Velociraptor	$\frac{distancia\ recorrida}{velocidad}$
Brachiosaurus	$(altura - largo\ del\ cuello) * peso$

Debe entregar:

- Diagrama de clases (20%)
- Código Java (80%)

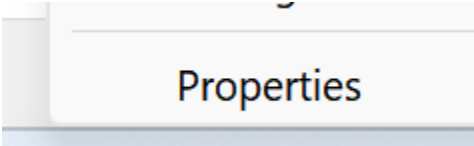
Consideraciones:

- Considere que los dinosaurios del nuevo planeta deben estar almacenados en **una sola lista**.
- Debe usar orientación al objeto. Debe usar herencia.
- Debe utilizar el patrón de diseño **Visitor** para resolver el problema.
- Otros patrones serán bienvenidos, pero no son obligatorios ni llevan puntaje.
- 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.

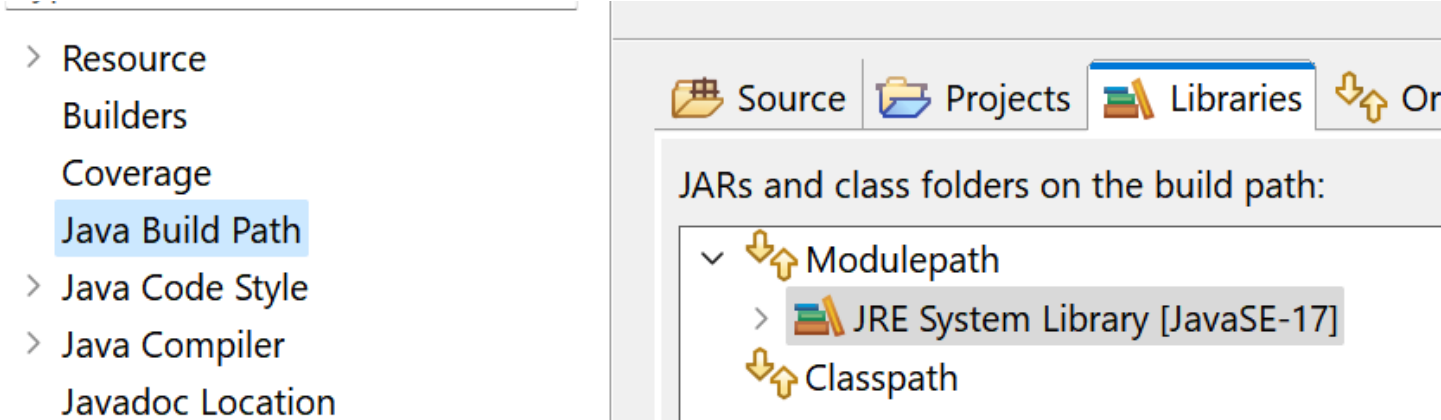
- El código fuente debe exportarlo como .zip y subirlo a Campus Virtual.

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

- Archivos .txt de ejemplo se encuentran en CampusVirtual.
- Los diagramas de clases debe escribirlos en papel y entregarlos junto a la prueba.
- Hojas sin nombre no se revisarán.
- Si el IDE no funciona correctamente haga clic derecho en el proyecto y luego en propiedades



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



Por último, selecciones Alternate JRE y aplique los cambios.

