

Java

Nombre: _____ JOSE ADRIAN FLORES AGUILAR _____
Fecha: 24/09/2024 _____ Telefono : _____

Parte I. Java Básico

Elabore correctamente lo que se pide a continuación:

1. Escriba una clase Java que represente un círculo; el círculo queda perfectamente definido si se conoce su radio. Defina además, para esta clase, tres métodos (públicos) que permitan calcular el área del círculo ($A = \pi \cdot r^2$), el perímetro de la circunferencia ($p = 2 \cdot \pi \cdot r$) que delimita el círculo y compara dos círculos que determina el mayor en función de su radio. Para probar la funcionalidad antes definida, escriba un pequeño programa que cree un círculo con un radio dado, y que calcule (y muestre por pantalla) el área y el perímetro de su circunferencia.

2. Dada la clase **Viaje** siguiente escribir un método denominado **uneViaje** que reciba como parámetros dos tipos **Viaje** y devuelva un nuevo objeto de esa misma clase con: el origen del primero; el destino del segundo; y como distancia la suma de las distancias de los dos viajes originales si el destino del primero coincide con el origen del segundo, sino se cumple dicha condición se devolverá NULL.

```
class Trayecto {  
    public String origen;  
    public String destino;  
    public double distancia;  
}
```

3. Desarrolla un programa de JAVA que almacene en un vector de tamaño inicialmente indeterminado un conjunto de enteros y en otro de las mismas características un conjunto de reales. Se debe mostrar por pantalla el mayor de los enteros y dado valor leído por teclado todos los valores reales que sean menores que el proporcionado por el usuario.



4. Se desea llevar un control del estado de una cuenta corriente; la cuenta corriente está caracterizada por su saldo y sobre ella se pueden realizar tres tipos de operaciones:

saldo: devuelve el saldo de la cuenta (puede ser negativo).

depósito: (cantidad): ingresa en la cuenta una cantidad de dinero.

retiro: (cantidad): saca de la cuenta una determinada cantidad de dinero.

Suponga que la cuenta inicialmente tiene un saldo de cero. Escriba una clase CuentaCorriente que implemente la funcionalidad descrita; escriba un pequeño programa principal para probar su funcionamiento.

5. Queremos modelar una casa con muchas bombillas, de forma que cada bombilla se puede encender o apagar individualmente. Para ello haremos una clase Bombilla con una variable privada que indique si está encendida o apagada, así como un método que nos diga si una bombilla concreta está. Cada objeto Bombilla se enciende y se apaga individualmente; pero sólo responde que está encendida si su interruptor particular está activado y además hay luz general. El código Java tiene esta estructura (Escriba un programa “main” que lo pruebe)

```
class Bombilla {
    private ...; // interruptor
    public void enciender () { ... }
    public void apagar () { ... }
    public boolean encendida () { ... }
    public Bombilla () { ... }
}
class Iluminacion {
    private ...; // Bombillas de la casa
    private ...; // Número de bombillas de la casa
    public void activaGeneral () { ... }
    public void desactivaGeneral () { ... }
    public apagar_bombilla() { ... }
    public encender_bombilla() { ... }
    public numero_bombillas_encendidas () { ... }
    public numero_bombillas_apagadas () { ... }
    public Iluminación { ... }
}
```

6. Escriba una función en javascript en la que se obtenga la cadena contenida en un campo de texto llamado 'inputUsuario', obtenga la longitud de la cadena y en caso de que sea mayor a 10 caracteres muestre en una alerta la cadena a partir del décimo carácter.

Conteste correctamente las siguientes preguntas:

1. Defina el modelo **MVC**.
Es un modelo de programación que nos permite tener mejor estructurado y modularizado nuestro código, el controlador recibe la solicitud, esta la envía al modelo el modelo responde la solicitud y el controlador la recibe y envía la respuesta a la vista.
2. ¿De qué manera implementa struts el modelo MVC?
Separa las responsabilidades de la lógica de negocio (modelo), la presentación (vista) y el manejo de la entrada del usuario (controlador)
3. ¿Para qué sirve el archivo **struts-config.xml**?
Sirve para configurar y definir diversos aspectos del comportamiento de una aplicación
4. ¿Para qué se usa el archivo **validation.xml**?
5. define las reglas de validación de los formularios, el permite especificar cómo se deben validar los datos ingresados por el usuario antes de que sean procesados.
6. Mencione 3 de los taglibs principales de Struts.
<html:form>: Crea un formulario HTML que se vincula a un bean.
<html:text>: Genera un campo de texto.
<html:submit>: Crea un botón de envío para el formulario.
7. Escriba un breve ejemplo con un taglib de iteración.
<logic:iterate id="usuario" name="usuarios">
8. ¿De qué clase extiende la clase Action de Struts?.
extiende de java.lang.Object
9. ¿Cómo se ocupa la clase ActionForm de Struts?.
se utiliza para el encapsulamiento de los datos de entrada de un formulario.
10. ¿A qué se denomina **filtro** en el ámbito de Struts?
permiten realizar operaciones específicas en las solicitudes y respuestas
11. ¿Para qué se usa el método que devuelve un objeto del tipo **ActionForward**?
se utiliza para redirigir el control después de que se haya procesado una petición
12. Escriba un ejemplo de configuración para el archivo **struts-config.xml** con un action mapping y dos forwards como mínimo.
<struts-config>

```
<form-beans>
  <form-bean name="loginForm" type="com.ejemplo.LoginForm" />
</form-beans>
```

```
<action-mappings>
  <action path="/login"
    type="com.ejemplo.LoginAction"
    name="loginForm"
    scope="request"
    input="/login.jsp">
```

```
    <forward name="success" path="/home.jsp" />
```

```
        <forward name="failure" path="/login.jsp" />
    </action>
</action-mappings>
```

```
</struts-config>
```

13. Al struts-config.xml anterior agregue un form bean y un global forward.

```
<struts-config>
  <form-beans>
    <form-bean name="loginForm" type="com.ejemplo.LoginForm" />
    <form-bean name="registroForm" type="com.ejemplo.RegistroForm" />
  </form-beans>

  <action-mappings>
    <action path="/login"
      type="com.ejemplo.LoginAction"
      name="loginForm"
      scope="request"
      input="/login.jsp">
      <forward name="success" path="/home.jsp" />
      <forward name="failure" path="/login.jsp" />
    </action>

    <action path="/registro"
      type="com.ejemplo.RegistroAction"
      name="registroForm"
      scope="request"
      input="/registro.jsp">

      <forward name="success" path="/bienvenida.jsp" />
      <forward name="failure" path="/registro.jsp" />
    </action>
  </action-mappings>

  <global-forwards>
    <forward name="error" path="/error.jsp" />
  </global-forwards>
</struts-config>
```

14. Defina ajax.

Es una técnica de desarrollo web que permite la actualización asíncrona de partes de una página web

15. Mencione un ejemplo de uso de ajax.

```
$ajax({
    type:'post',
    url:'http://localhost:9000/WebService/eliminar',
    data: JSON.stringify(json),
    contentType:'application/json; charset=UTF-8',
```

```
        success:function(respuesta){  
            $('#.alert-danger').html("Se elimino  
proveedor").fadeIn().delay(3000).fadeOut('snow');  
            listarRegistros(); //Actualiza la lista de registros despues de guardar  
        },  
        error: function(){  
            console.log("error al eliminar");  
        }  
    });
```

16. ¿Qué es el Document Object Model?

Es una interfaz de programación que proporciona una representación estructural de documentos HTML y XML.