

EIF206 Programación 3

Examen Parcial #1

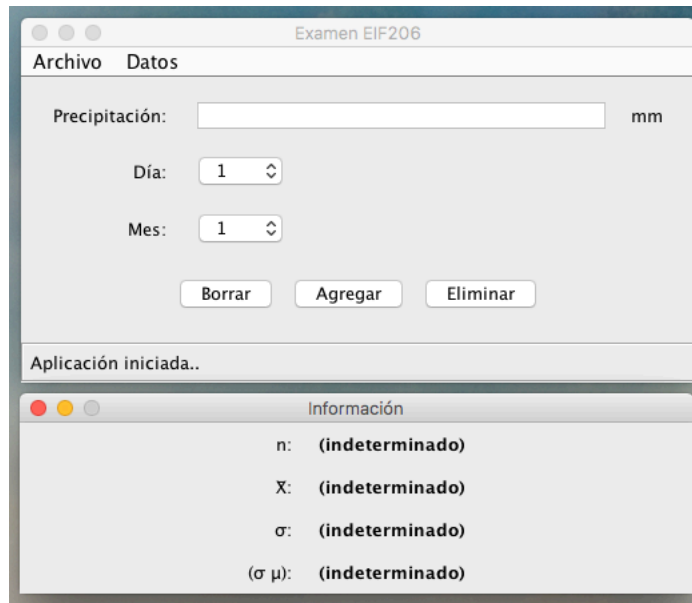
Prof. Georges Alfaro S.

Complete cada uno de los ejercicios solicitados.
La resolución del examen es **estrictamente individual**.

PARTE ÚNICA (100%)

Un grupo de investigación está haciendo un estudio sobre la manera en que han cambiado los patrones de lluvia en el país. Para ello, se ha estado recolectando información de la precipitación en diferentes zonas geográficas. Para cada observación registrada se anota el valor de la precipitación ($mm \times m^2 \times t$), el día y el mes de obtención de dicho valor (el año no se registra). La unidad de tiempo no se ha especificado, pero se asume generalmente un período de 24 horas. No hay un máximo predeterminado de observaciones que pueden registrarse. Como apoyo a la investigación, se le ha encomendado construir un pequeño programa para capturar y almacenar dicha información.

1. Escriba un programa definiendo las clases necesarias para representar cada uno de sus componentes lógicos, usando la arquitectura MVC (*Modelo-Vista-Controlador*). Coloque cada uno de los componentes en paquetes separados según su estereotipo.
2. Defina la interfaz del programa usando **dos ventanas independientes**.
En la ventana principal de la aplicación, coloque una barra de menú que contenga dos menús: un primer menú "Archivo", contendrá únicamente una opción para salir de la aplicación. También habrá un segundo menú "Datos", con dos opciones: una para borrar la última observación registrada y otra para eliminar todos los datos almacenados.
3. Usando las etiquetas y campos de texto necesarios, incluya los controles para capturar los datos de cada observación: precipitación, día y mes. Todos los valores son numéricos y positivos. El monto de precipitación es un valor de punto flotante, que se mostrará usando 3 posiciones decimales. El valor máximo permitido para la precipitación es de 10.000,0. El día y el mes deberán verificarse adecuadamente. Cuando el día o el mes sean demasiado grandes, deberán sustituirse por los valores máximos permitidos. Por ejemplo, si el usuario digita el valor 20 para el mes, se sustituye por un 12 (diciembre). Si el número de mes es correcto (4 para abril, por ejemplo), pero el día es incorrecto, éste se sustituye por un 30, que es el valor máximo permitido para el mes de abril. Considere que el mes de febrero tiene 28 días en todos los casos.
El uso de *combo boxes* permitirá capturar correctamente los datos.
Coloque tres botones para completar la captura: el primero que borre todos los campos del formulario; el segundo, que tome los datos y los ingrese en el conjunto de observaciones. El último botón elimina una observación, si ésta se encuentra en el conjunto (Con los mismos valores para la precipitación, el día y el mes). Si el dato no se encuentra en el conjunto, deberá indicarlo en la barra de estado de la ventana. Cada vez que se ingresa un nuevo conjunto de datos, se borrarán los campos y se indicará en una barra de estado al pie de la ventana cuáles valores se han ingresado (es decir, los tres valores capturados), usando una abreviatura de tres letras para el mes, en lugar del valor numérico. Recuerde que es importante que dicha información de estado sea mostrada como respuesta a un evento de actualización desde el modelo y no por la interacción del usuario.



4. En la segunda ventana del programa, se mostrarán indicadores estadísticos de la información registrada. La ventana de información se abre automáticamente y el usuario no puede cerrarla directamente. Deberá indicarse:

- El total de datos ingresados (n)
- El promedio de los valores de precipitación (\bar{x})
- La desviación estándar de las muestras (σ)
- El error estándar de las muestras (σ_{μ})

Use los símbolos adecuados para mostrar cada valor, usando los caracteres UNICODE correspondientes. Consulte la página <http://www.unicode.org/charts/> para encontrar los códigos adecuados. Los valores deben actualizarse como respuesta a eventos recibidos desde el modelo. Utilice las siguientes fórmulas para calcular cada uno de los parámetros indicados:

$$\bar{x} = \frac{\sum x}{n}$$

$$\sigma = \sqrt{\frac{\sum x^2 - \frac{(\sum x)^2}{n}}{n}}$$

$$(\sigma_{\mu}) = \frac{\sigma}{\sqrt{n}}$$

donde:

n es la cantidad de datos recolectados

$\sum x$ es la suma de las observaciones (precipitaciones)

$\sum x^2$ es la suma de los **cuadrados** de las observaciones (precipitaciones)

Considere los siguientes datos de prueba para comprobar los resultados de su programa:

Muestra	Precipitación	Día	Mes
1	220.186	14	1
2	118.817	25	3
3	707.698	16	4
4	93.844	16	4
5	255.851	4	5
6	242.817	6	6
7	567.460	12	7
8	351.463	24	7
9	374.094	30	7
10	527.317	2	8
11	77.670	21	8
12	38.387	31	8
13	319.526	17	9
14	230.777	22	9
15	445.105	3	10
16	570.316	10	10
17	172.692	16	10
18	562.949	6	11
19	88.663	9	12
20	242.902	15	12

n	20
Σx	6,208.534
Σx^2	266,0045.463

media	310.427
desviación	191.409
error	42.800

Todos las observaciones y los resultados se muestran con 3 decimales.

- Utilice menús desplegables (*combo boxes*) para facilitar la captura de la información del día y el mes.
- Utilice un método `toString()` para mostrar el conjunto completo de datos en la consola, para comprobar que han sido capturados correctamente.

EVALUACIÓN

El examen se evaluará según los siguientes criterios:

Implementación de la solución	100,0%			
Organización del código	20,0%	Todo el código está separado en unidades lógicas y bibliotecas según su función.	El código está separado pero hay errores en la la arquitectura general de la aplicación.	El código está separado pero no existe una arquitectura (MVC) o modelo de capas correctamente definido.
		20,0%	12,0%	5,0%
Diseño de la interfaz	20,0%	El programa es fácil de utilizar: las opciones del usuario son claras y consistentes. Se utilizan adecuadamente los administradores de disposición (<i>layout managers</i>)	Algunas opciones del programa no se pueden usar o se emplean los administradores de disposición incorrectamente.	La interfaz de usuario es confusa o no permite la utilización correcta del programa.
		20,0%	12,0%	5,0%
Control (básico) de errores	5,0%	El programa controla todas las entradas del usuario y los posibles valores incorrectos.	Algunos datos de entrada no se verifican.	Algunos datos de entrada no se verifican o hay errores de ejecución por valores incorrectos.
		5,0%	2,0%	0,0%
Manejo del modelo	20,0%	Todas las operaciones de actualización funcionan correctamente y los eventos son manejados de acuerdo con la arquitectura.	Algunas operaciones de actualización no se completan de manera satisfactoria. Uno o más elementos de la interfaz son actualizados directamente, sin responder a los eventos en el modelo.	La mayoría (> 50%) de las operaciones no funcionan correctamente.
		20,0%	12,0%	5,0%

Funcionalidad		Se asigna el mismo puntaje a cada opción implementada. La opción puede estar implementada completamente, parcialmente pero correcta en general, parcialmente e incorrecta en general o totalmente, o la opción puede no haber sido implementada.			
	Inclusión de datos	5,0%	5,0%	2,0%	0,0%
	Borrado (última observación)	5,0%	5,0%	2,0%	0,0%
	Borrado (todas las observaciones)	5,0%	5,0%	2,0%	0,0%
	Borrado (dato específico)	5,0%	5,0%	2,0%	0,0%
	Cálculo (promedio)	5,0%	5,0%	2,0%	0,0%
	Cálculo (desviación)	5,0%	5,0%	2,0%	0,0%
	Cálculo (error)	5,0%	5,0%	2,0%	0,0%
		35,0%			
	Uso de combo boxes para la captura de datos		Los controles se utilizan correctamente.	Existe algún error en el uso del control.	
	10,0%	10,0%	5,0%		

Observe que el examen está calificado sobre 105 puntos. Es decir, se pueden obtener hasta 5 puntos extras sobre la calificación.

NOTAS Y OBSERVACIONES

1. Escriba todo el código en Java. No utilice bibliotecas de clases adicionales a las provistas en la instalación estándar de J2SE.
2. El programa debe separarse en paquetes, utilizando por lo menos un paquete para cada componente lógico de la arquitectura de la aplicación (clases de entidad, control e interfaz). Además, debe construir el programa empleando la arquitectura **MVC** (*Modelo-Vista-Controlador*).
3. Incluya la carpeta completa del proyecto de NetBeans. En caso de no utilizar el IDE, incluya un archivo de comandos para el *shell* de Linux que permita compilar la aplicación. **Los programas serán probados en el ambiente de Linux.** En caso de desarrollar los ejercicios usando Windows, asegúrese de que puedan compilarse correctamente en Linux.
4. Todas las actualizaciones sobre el modelo deberán mostrarse adecuadamente en la interfaz usando eventos generados desde el modelo.
5. El programa permite ingreso, borrado y consulta de información, pero **NO implementará operaciones de modificación** de los datos ya registrados. Los datos tampoco se guardarán en ningún archivo o usando otra técnica. Los datos se almacenan solamente durante la ejecución del programa. Puede sin embargo, utilizar archivos para cargar los datos de prueba del examen.
6. Ninguna de las ventanas de la aplicación puede cambiar de tamaño. Deberá usar uno o varios de los administradores de disposición siguientes para distribuir adecuadamente los componentes en la ventana:
 - `FlowLayout`
 - `GridLayout`
 - `GridBagLayout`
 - `BorderLayout`
 - `BoxLayout`En particular, **NO** puede usar un administrador nulo (*null*), absoluto o el `GroupLayout`.
7. Los datos numéricos deben validarse adecuadamente según su formato. Los valores a capturar son únicamente positivos.