

Algorítmica y Programación por Objetos 2

Ejercicio Nivel 9

Circo

Descripción global

El Circo Cupi2 requiere un programa para planear su itinerario anual y sus giras. Para ello, necesita manejar la información de las ciudades donde va a estar y las funciones que va a presentar en cada una de ellas

Objetivos

El objetivo de este ejercicio es que el estudiante comprenda y adquiera práctica en:

- El desarrollo de aplicaciones siguiendo un proceso incremental.
- La implementación y uso de estructuras lineales sencilla y doblemente encadenadas.
- La creación de algoritmos de adición, eliminación, búsqueda y ordenamiento en listas.
- El desarrollo de nuevos elementos gráficos para construir interfaces de usuario.

Los siguientes pasos conforman el plan sugerido para desarrollar el ejercicio. La idea es ir desarrollando y probando incrementalmente los métodos de las clases.

Este ejercicio debe ser realizado de manera **INDIVIDUAL**.

Preparación

Esta sección presenta una lista de chequeo de todas las tareas necesarias para la preparación del ejercicio. Por favor, revise que cada tarea haya sido completada **antes** de pasar a la siguiente sección de esta guía de trabajo.

Nota: En el siguiente enlace se encuentran las instrucciones de instalación de Java y Eclipse:
<https://cupitaller.uniandes.edu.co/instaladores/>.

1. Para conocer el funcionamiento esperado de la aplicación, descargue y/o ejecute el archivo demo del ejercicio que se encuentra en el siguiente enlace:
<https://cupi2.virtual.uniandes.edu.co/ejercicios-del-semestre-apo2/ejercicio-n9>.
2. Descargue el esqueleto del ejercicio que se encuentra en el siguiente enlace:
<https://cupi2.virtual.uniandes.edu.co/ejercicios-del-semestre-apo2/ejercicio-n9>.
3. Descomprima este archivo e importe el proyecto llamado **n9_circo** a Eclipse. Limpie el ejercicio; para ello en Eclipse vaya a: **Project > Clean > Ok**. En el siguiente enlace hay un video con un ejemplo de cómo limpiar un proyecto en eclipse: <https://youtu.be/mbcP46wXS0>.
4. Lea el enunciado del problema disponible en:
n9_circo/docs/specs/Descripcion.pdf.
5. Estudie el documento de requerimientos funcionales disponible en:

n9_circo/docs/specs/RequerimientosFuncionales.pdf.

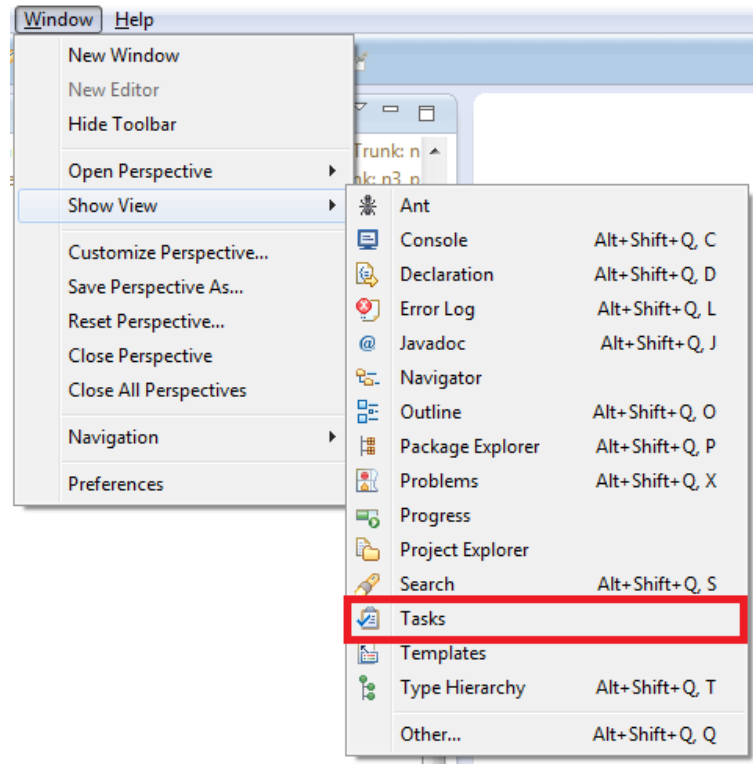
6. Estudie el documento de consideraciones adicionales de diseño disponible en:
n9_circo/docs/specs/ConsideracionesAdicionalesDeDiseno.pdf.
7. Estudie el documento de requerimientos no funcionales disponible en:
n9_circo/docs/specs/RequerimientosNoFuncionales.pdf.
8. Estudie el modelo del mundo diseñado para este ejercicio. Este modelo se encuentra en:
n9_circo/docs/specs/ModeloConceptual.png. Identifique las clases, relaciones entre clases, constantes, atributos y métodos.
9. Asegúrese de tener activado el uso de aserciones para la ejecución del programa. Puede ver el tutorial en: <http://cupitaller.virtual.uniandes.edu.co/videos-guia/>.

Desarrollo

En esta sección se explican detalladamente las tareas necesarias para completar el desarrollo del ejercicio. Una vez importado el esqueleto del ejercicio a Eclipse, es necesario identificar los puntos en donde usted deberá realizar alguna modificación puntual (por ejemplo: añadir atributos, completar métodos, etc.). Estos puntos se identifican mediante comentarios especiales denominados: **TODO** (*por hacer*) y tienen la siguiente estructura:

// TODO ParteX PuntoY: Breve explicación de la modificación que debe realizar.

Donde X hace referencia a una parte de la guía de trabajo, y Y hace referencia a un punto de la parte X. Para visualizar los **TODOs** vaya al menú **Window > Show View > Tasks** como se muestra en la siguiente figura:



En el siguiente enlace <https://youtu.be/pV54O42D2ow> encuentra un video que muestra cómo consultar los TODOs de un programa. Esto le permitirá desarrollar el ejercicio en forma ordenada.

Para facilitar el desarrollo de los TODOs, se recomienda ver el siguiente video: <https://youtu.be/NnRvintGJAo>, el cual enseña a autocompletar los nombres de clases, métodos y atributos.

Opcionalmente, se recomienda ver el video: https://youtu.be/zYXTlvrOZ_Y, el cual le enseña a renombrar un método o un atributo de forma automática y segura en caso de haber cometido un error en su nombramiento.

Por último, el video disponible en <https://youtu.be/O-dirtGpy50> le enseña cómo determinar en dónde se está usando (se llama) un atributo o un método en un programa.

Parte1: Listas doblemente encadenadas

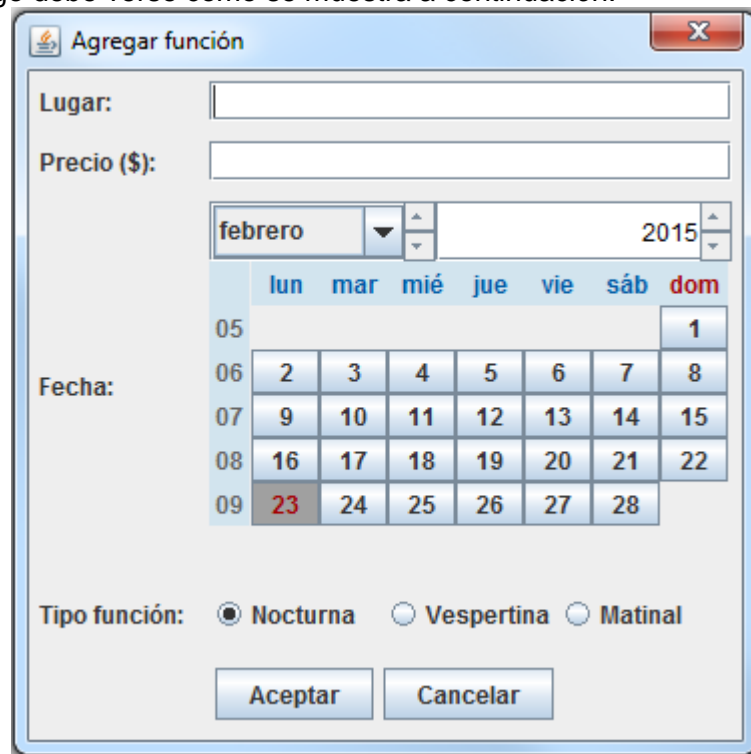
Completar los TODOs que comienzan por: **//TODO Parte1 PuntoY**

Parte2: Listas sencillamente encadenadas

Completar los TODOs que comienzan por: **//TODO Parte2 PuntoY**

Parte 3: Interfaz Gráfica

- Cree un diálogo `DialogoAgregarFuncion` (a partir de la clase `JDialog`) que permita agregar una función a una ciudad. **Nota:** Puede usar como referencia `DialogoAgregarCiudad`.
- Implemente el diálogo usando un distribuidor gráfico de tipo `GridBagLayout`.
- Use el siguiente código para agregar el calendario.
`JCalendar calendarioFecha = new JCalendar();`
Este objeto se puede agregar al layout del dialogo.
- Use `JRadioButton` y `ButtonGroup` para ingresar el tipo de la función.
- Valide los datos ingresados por el usuario.
- Llame al método de la interfaz principal `agregarFuncion` para agregar la función a la ciudad.
- El diálogo debe verse como se muestra a continuación:



The screenshot shows a Java Swing dialog box titled "Agregar función". It has a standard window title bar with a close button. The dialog contains the following elements:

- Two text input fields: "Lugar:" and "Precio (\$):".
- A date selection component showing "febrero" in a dropdown and "2015" in a year spinner.
- A calendar grid for the month of February. The days of the week are labeled "lun", "mar", "mié", "jue", "vie", "sáb", "dom". The date "23" is highlighted in red.
- A "Fecha:" label to the left of the calendar.
- Three radio buttons under "Tipo función:" with labels "Nocturna", "Vespertina", and "Matinal". The "Nocturna" radio button is selected.
- Two buttons at the bottom: "Aceptar" and "Cancelar".

Validación

Para comprobar el funcionamiento de su ejercicio usted puede:

- Ejecutar el programa e interactuar con todas las opciones disponibles en la interfaz. Los resultados obtenidos deben ser iguales a aquellos mostrados en el video demo.
- Ejecutar las pruebas automáticas disponibles en el ejercicio. En el siguiente video <https://youtu.be/rVd4AD8XMJk> se explica cómo efectuar esas pruebas. Estas pruebas deben

presentar resultados en verde (0 errores y 0 fallas) cuando el ejercicio ha sido completado correctamente.

Tenga en cuenta que esas pruebas no son exhaustivas y que su correcto funcionamiento no garantiza que no haya ningún error en su programa.

Entrega

Este ejercicio debe ser realizado de manera **INDIVIDUAL**.

1. Indente el código fuente de todas las clases del mundo. En el siguiente enlace <https://youtu.be/cwQ9QiauaSc> encuentra un video que explica cómo indentar el código fuente de su ejercicio.
2. Limpie el proyecto para que la entrega no contenga archivos ejecutables ni temporales (<https://youtu.be/mbcY46wXS0>).
3. Construya el archivo entregable con su ejercicio desarrollado y validado completamente. En el siguiente video <https://youtu.be/xuSDFfEZW78> se explica detalladamente el proceso para producir el comprimido del ejercicio y enviarlo a SicuaPlus. Renombre el archivo a entregar con su login de la siguiente forma:

n<nivel del ejercicio>_<login estudiante>.zip
Por ejemplo: **n9_tsuarez.zip**

La no indentación del código fuente o el nombramiento incorrecto del ejercicio en su entrega es una acción penalizada en la plantilla de calificación del mismo.

4. Entregue el archivo del ejercicio vía SicuaPlus, de acuerdo con las normas, fecha y hora de entrega.