

Algorítmica y Programación por Objetos 2

Ejercicio Nivel 7

Juguetería

Descripción global

Una tienda de juguetes desea una aplicación para manejar su catálogo de forma digital. Dicho catálogo debe manejar la información más importante de cada juguete de la tienda: número de identificación (debe ser único y positivo), nombre, marca, descripción, categoría a la cual pertenece (video juegos, juegos mecánicos, juegos electrónicos, juegos exteriores), edad recomendada, precio e imagen.

Objetivos

Con el presente ejercicio el estudiante:

- Repasará algunos de los conceptos vistos en el curso anterior (Mundo e Interfaz).
- Desarrollará una aplicación siguiendo un proceso incremental.
- Construirá los invariantes de las clases del mundo del ejercicio.
- Utilizará la instrucción `assert` de Java para verificar invariantes.
- Desarrollará pruebas unitarias en Junit para las clases del ejercicio.
- Entenderá y aplicará el concepto de comparación de objetos.
- Entenderá y desarrollará tres algoritmos de ordenamiento (burbuja, inserción y selección).
- Entenderá y desarrollará algoritmos de búsqueda binaria y secuencial sobre una lista ordenada o no ordenada.
- Aprenderá a representar un objeto como un texto con el método `toString()`.
- Utilizará `JList` y `JScrollPane` para presentar listas en la interfaz gráfica.

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-semester-apo2/ejercicio-n7>.

2. Descargue el esqueleto del ejercicio que se encuentra en el siguiente enlace:

<https://cupi2.virtual.uniandes.edu.co/ejercicios-del-semester-apo2/ejercicio-n7>.

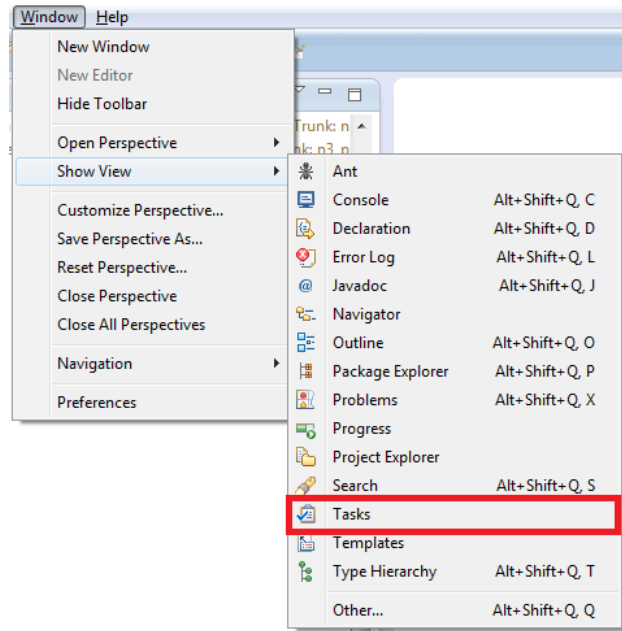
3. Descomprima este archivo e importe el proyecto llamado **n7_jugueteria** a Eclipse. Limpie el ejercicio; para ello en Eclipse vaya a: **Project > Clean > Ok**. En el siguiente enlace hay un video con un ejemplo de cómo limpiar un proyecto en eclipse: <https://youtu.be/mbcpY46wXS0>.
4. Lea el enunciado del problema disponible en:
n7_jugueteria/docs/specs/Descripcion.pdf.
5. Estudie el documento de requerimientos funcionales disponible en:
n7_jugueteria/docs/specs/RequerimientosFuncionales.pdf.
6. Estudie el modelo del mundo diseñado para este ejercicio. Este modelo se encuentra en:
n7_jugueteria/docs/specs/ModeloConceptual.png. Identifique las clases, relaciones entre clases, constantes, atributos y métodos.
7. Asegúrese de tener activado el uso de aserciones para la ejecución del programa. Puede ver el tutorial en: <https://youtu.be/Qt0clQmflPw>.

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: completar métodos, crear la invariante, 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 se debe realizar.

Donde, **X** hace referencia a una parte de esta guía de trabajo y **Y** hace referencia a una subsección de **X**. Para visualizar y acceder a los TODOS, en Eclipse vaya al menú: **Window > Show View > Tasks**, tal y como se muestra en la siguiente figura.



En el siguiente enlace <https://youtu.be/NnRvintGJAo> 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/xHIRI-yX6HY>, 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: Construcción de invariantes

1. Revise el modelo del mundo en `n7_jugueteria/docs/specs/ModeloConceptual.JPG` e identifique las clases del mundo.
2. Complete la clase **Juguete**
 - a. Completar los TODOs que comienzan por: `//TODO Parte1`.
 - b. Utilice (invoque) el método `verificarInvariante()` para verificar el invariante en todos aquellos métodos de la clase que modifican el estado.
3. Complete la clase **Jugueteria**

- a. Completar los TODOs que comienzan por: `//TODO Parte1`.
- b. Utilice (invoque) el método `verificarInvariante()` para verificar el invariante en todos aquellos métodos de la clase que modifican el estado.

Parte2: Comparaciones, ordenamiento y búsqueda

1. Complete la clase **Juguete**:
 - Completar los TODOs que comienzan por: `//TODO Parte2`
2. Complete la clase **Jugueteria**:
 - Completar los TODOs que comienzan por: `//TODO Parte2`

Parte3: Creación de listas en la interfaz

1. Cree el método `toString()` de la clase **Juguete** para que retorne una cadena de caracteres con el nombre del juguete y su marca entre paréntesis.
2. Complete la clase **PanelCatalogo**:
 - Completar los TODOs que comienzan por: `//TODO Parte3`

Validación

En esta sección se explica cómo validar el desarrollo del ejercicio. Para comprobar el completo y correcto funcionamiento de su ejercicio usted debe:

1. Ejecutar el programa e interactuar con todas las opciones disponibles en la interfaz. Los resultados obtenidos al ejecutar el programa deben ser iguales a los del video demo, disponible en el [sitio web del curso](#).
2. 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/mbcpY46wXS0>).
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: **n7_tsuares.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.