

Algorítmica y Programación por Objetos 2

Ejercicio de Nivel 10

Tangram

Descripción global

El equipo Cupi2 quiere hacer una aplicación que recree un antiguo juego chino llamado Tangram. Un Tangram es un cuadrado que se divide en varias figuras y el objetivo del juego es usarlas para armar diferentes siluetas sin superponerlas.

Objetivos

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

- El desarrollo de aplicaciones siguiendo un proceso incremental
- La utilización de interfaces de Java (Interface) para disminuir el acoplamiento de clases en una aplicación.
- La utilización del concepto de herencia como mecanismo para la reutilización de código.
- La utilización de Interfaces ya existentes en Java como son Collection, List y otras.
- El recorrido de colecciones mediante Iteradores.
- El dibujo simple de 2D.
- El uso de eventos del ratón.
- La escritura de clases autopersistentes.
- El desarrollo de pruebas unitarias en JUnit para las clases del ejercicio.

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-n10>.
2. Descargue el esqueleto del ejercicio que se encuentra en el siguiente enlace:
<https://cupi2.virtual.uniandes.edu.co/ejercicios-del-semester-apo2/ejercicio-n10>.

3. Descomprima este archivo e importe el proyecto llamado **n10_tangram** a Eclipse.

El siguiente video explica detalladamente la forma en la cual se descarga, descomprime e importa a Eclipse, para Windows (en el enlace: <https://youtu.be/4PDAn55uNTI>) y para MAC (en el siguiente enlace: <https://youtu.be/XlkMktjIN6c>). Es muy importante que consulte este video, porque lo explicado allí se realizará a lo largo de todo el semestre en el curso y usted deberá hacerlo con especial agilidad en todos los laboratorios y parciales prácticos.

No se preocupe si algunas clases le reportan errores (como marcas rojas) al importar el ejercicio por primera vez. Estos desaparecerán cuando termine correctamente los cambios requeridos en el mundo.

Finalmente, 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/mbcY46wXS0>.

4. Lea el enunciado del problema del ejercicio disponible en:
n10_tangram/docs/specs/Descripcion.pdf
5. Lea el documento de requerimientos funcionales disponible en:
n10_tangram/docs/specs/RequerimientosFuncionales.pdf
6. Lea el documento de requerimientos no funcionales disponible en:
n10_tangram/docs/specs/RequerimientosNoFuncionales.pdf
7. Lea el documento de consideraciones adicionales de diseño en:
n10_tangram/docs/specs/ConsideracionesAdicionalesDeDiseno.pdf
8. Explore el modelo del mundo diseñado para este ejercicio. Este modelo se encuentra en:
n10_tangram/docs/specs/ModeloConceptual.png.
9. Desde Eclipse identifique los elementos del mundo en Java (paquetes, clases, métodos y atributos).

Se recomienda que consulte los siguientes videos:

- <https://youtu.be/KmhDGxPIGwl>. Este video le explicará cómo acceder al explorador de Eclipse.
- <https://youtu.be/j2AQKcB8pG4> Este video le explicará cómo leer los contratos de un método.
- <https://youtu.be/tYxAI0F6s-c>. Este video le explicará cómo listar y acceder directamente a los métodos y atributos de una clase en forma rápida y eficiente.

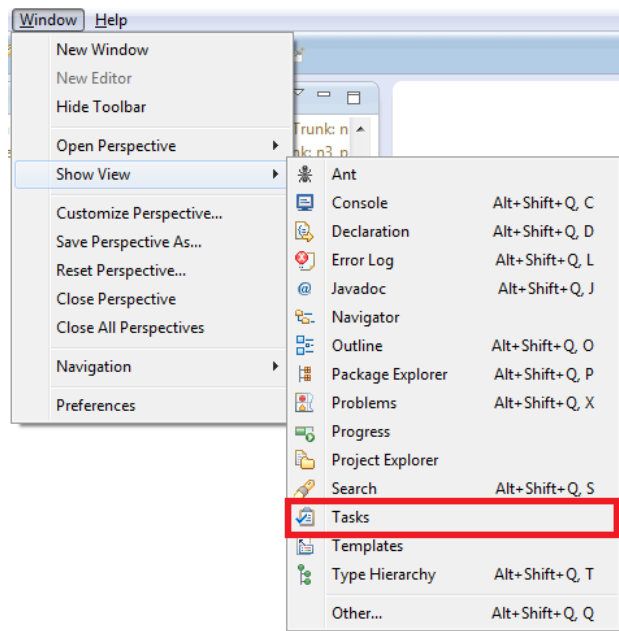
- <https://youtu.be/Lnu9udEXTew>. Este video le explicará cómo dirigirse a la implementación (declaración) de una clase, método y atributo (objeto). Este conocimiento le será de gran utilidad para agilizar el desarrollo de su ejercicio y de parciales prácticos.

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 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.

Parte 1: Completar la clase Figura.

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

Parte 2: Crear las clases de tipo Figura.

Basado en el modelo conceptual, la descripción del problema y el documento de consideraciones adicionales de diseño, desarrolle los siguientes puntos:

1. Estudie y familiarícese con la implementación de la clase `Triangulo`. Esta le puede servir de guía para la implementación de las siguientes clases.
2. Cree e implemente la clase **Cuadrilatero**. Esta clase extiende de la clase **Figura**.
3. Estudie y familiarícese con la implementación de la clase `Cuadrado`. Esta le puede servir de guía para la implementación de las siguientes clases. Observe la implementación del método de `establecerPuntos` donde se usan los métodos `reset` y `addPoint` de la clase `Polygon`.
4. Cree e implemente la clase **Trapeccio**. Esta clase extiende de la clase **Cuadrilatero**.
5. Cree e implemente la clase **TrianguloIsocelos**. Esta clase extiende de la clase **Triangulo**.

Ayudas:

- El atributo tipo al inicializarlo debe quedar de la forma `TIPO+" "+ pProporcion` para estas dos nuevas clases.
- Tenga en cuenta que dependiendo de la proporción el ancho y el alto cambian

Parte 3: Completar la clase Tangram.

Invariantes

1. Documente el invariante de la clase en la cabecera de la clase, siguiendo las normas explicadas para ello.
2. Cree al final de la clase el método `verificarInvariante` y utilice aserciones para validar el invariante que definió en el punto anterior. Si lo considera necesario, implemente métodos privados para hacer verificaciones internas.

3. Utilice (invoque) el método `verificarInvariante` para verificar el invariante en todos aquellos métodos de la clase que modifican el estado.
4. Completar los TODOs que comienzan por: **//TODO Parte3 PuntoY**

Parte 4: Completar la clase BarraMenu.

El objetivo de esta parte es manejar las opciones “Nuevo”, “Abrir”, “Guardar”, “Guardar Como” y “Salir”, a través de un menú. Para esto debe:

1. Declarar los atributos faltantes según la documentación.
2. Completar el método constructor según la documentación proporcionada.
3. Completar el método `actionPerformed` según la documentación suministrada. Este método debe controlar las acciones a seguir si alguna de las cinco opciones del menú es seleccionada.

Parte 5: Completar la clase PanelBotones.

Completar los TODOs que comienzan por: **//TODO Parte7 PuntoY**

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: **n10_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.