

Reglas del juego:

- Su nombre y su firma a la aceptación del compromiso de no hacer fraude, en la hoja de respuestas a este examen, deben ir en lapicero. Si emplea más de una hoja márkelas TODAS de igual forma.
- Por ningún motivo puede salir del salón, antes de terminar el examen. De manera que si se retira se considerará que terminó su trabajo.
- En la parte práctica, puede consultar sus notas, el libro o apuntes; pero se le recomienda que lo haga cuando esté seguro de qué quiere buscar, en caso contrario estas consultas se convierten en una distracción que le hace perder tiempo. NO puede consultar código (incluyendo talleres o ejercicios realizados).
- Debe escribir con letra clara

**I (20%) Teoría (Responder todas las preguntas en el cuadernillo)**

- 1) (Colecciones y API) Marque cuales de las siguientes afirmaciones son correctas (5%):
  - a) ArrayList es una Interfaz que hereda de List y permite crear colecciones.
  - b) Los conjuntos son un tipo de colección que no permiten elementos repetidos.
  - c) Para utilizar listas es necesario implementar la Interfaz Comparable e implementar compareTo
  - d) Las clases para el manejo de colecciones, se encuentran en el paquete java.util
  - e) En términos generales ArrayList y LinkedList son funcionalmente equivalentes.
- 2) (Colecciones) Explique clara y brevemente en qué casos utilizaría LinkedList en lugar de ArrayList (5%).
- 3) (Interfaces) Marque todas las que apliquen. Las interfaces permiten (5%):
  - a) Definir el comportamiento que deben tener las clases que las implementan.
  - b) Definir variables estáticas.
  - c) Definir constantes que pueden usarse en otras clases.
  - d) Definir el código de algunos métodos.
- 4) (Paquetes y API) Explique clara y brevemente qué son los paquetes y cómo se utilizan en el API de Java (5%).

## II (40%) Análisis

Se necesita tener el diagrama de clases para una aplicación que permita mostrar y controlar las ventanas en un pasillo de oficinas. No es necesario incluir dentro del análisis, la oficina como tal, solamente las ventanas.

Cada una de las ventanas, puede tener persianas para cubrir el vidrio de la misma. Es posible que una ventana tenga más de una persiana, en cuyo caso cada persiana se puede controlar independientemente. Cada ventana tiene unas dimensiones y una ubicación

Las persianas pueden ser de dos tipos, de PVC o de madera. Todas las persianas deben poderse subir y bajar. Adicionalmente, las persianas de PVC deben poderse abrir y cerrar (girar sus láminas).

Una persiana de PVC tiene un grupo de láminas de PVC. Cada lámina debe tener una posición, orientación y color. Las láminas son las que permiten bloquear la visibilidad hacia el interior de la oficina.

Una persiana de madera tiene un grupo de tablillas. Cada tablilla debe tener una posición y color.

Elabore el diagrama de clases correspondiente al sistema descrito, de manera que se puedan tener varias ventanas y se puedan controlar las persianas. Es decir, que dependiendo del tipo de persiana, se pueda subir, bajar, abrir y cerrar.

Tenga en cuenta que debe poder conocer el estado de las persianas, es decir, que tan cerradas o abiertas se encuentran y que tan arriba o abajo están.

No debe escribir el código de ninguna clase o método, solamente el diagrama UML completo.

### III (40%) Práctico: código

Se le ha proporcionado la clase abstracta Figura que se encuentra en moodle, Esta clase debe seguir siendo abstracta, pero se debe heredar de ella, ya que debe existir un tipo de figura horizontal y un tipo de figura vertical. El tipo de la figura determina su cambio de posición como se verá más adelante.

Utilizando colecciones, realice lo siguiente (Lea todo antes de empezar a desarrollarlo):

- Inicialización: (12%)
  - Definir un lienzo de 400 x 400.
  - Crear las clases de tipo horizontal y vertical. Todos los objetos de la primera clase, automáticamente deben ser verdes, mientras que los de la segunda deben ser rojos.
  - Crear tres objetos de cada clase (seis en total) y almacenarlo en una sola colección.
    - Los objetos deben estar ubicados en la parte inferior del lienzo inicialmente.
- Visualización: (6%)
  - Mostrar los objetos en la posición actual
    - Solamente se deben mostrar los que existen actualmente, es decir que si se borran deben dejar de aparecer
- Animación: (12%)
  - Al pasar sobre el objeto y hacer clic se debe mover de acuerdo al tipo que sea.
    - Para las figuras verticales, se debe mover aleatoriamente a otra posición dentro de la ventana, cambiando solamente el valor de Y.
    - Para las figuras horizontales, se debe mover aleatoriamente a otra posición dentro de la ventana, cambiando solamente el valor de X.
- Alterar la colección (10%)
  - Borrar un objeto, cuando se presiona el clic derecho sobre el mismo.

Tenga en cuenta que la clase abstracta Figura ya cuenta con los métodos validar, pintar y el constructor.

Restricciones:

- La clase Figura proporcionada debe seguir siendo abstracta.
- No es necesario crear nuevos métodos en la clase Figura.
- Los atributos de la clase figura deben seguir siendo privados.
- Se deben utilizar colecciones para almacenar los elementos.