

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, los documentos pdf del curso, el API, 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 en internet fuera de lo especificado arriba, ni código (incluyendo talleres, tareas, proyectos o ejemplos realizados).
- Debe escribir con letra clara.

**I (25%) Teoría: (Responder todas las preguntas en el cuadernillo)**

- 1) ¿En qué se diferencian las listas de los conjuntos? Marque de las siguientes afirmaciones todas las que establecen diferencias con los conjuntos (5%).
  - a) Las listas permiten establecer la igualdad entre objetos.
  - b) Las listas permiten recorrer directamente los elementos.
  - c) Las listas permiten conocer las posiciones de los elementos.
  - d) Las listas permiten guardar cualquier cantidad de elementos.
  - e) Las listas permiten evitar elementos repetidos.
- 2) ¿Al definir una clase que hereda de otra, qué es correcto? Marque de las siguientes afirmaciones todas las correctas (5%).
  - a) No definir en la nueva clase los atributos heredados.
  - b) Sobreescribir los métodos abstractos.
  - c) Definir la nueva clase abstracta, si se definen métodos abstractos.
  - d) Modificar solo el tipo de retorno en los métodos sobreescritos.
  - e) Acceder directamente a atributos privados y protegidos de la clase padre.
- 3) Al implementar en una clase de una interfaz (5%)
  - a) Se debe escribir el código de los métodos definidos en la interfaz.
  - b) No se puede heredar de otra clase.
  - c) Se pueden crear nuevos métodos.
  - d) Es necesario que la clase sea estática.
  - e) Ninguna de las anteriores, las interfaces no se implementan.
- 4) ¿Cuál es la utilidad las clases Date y SimpleDateFormat? Sea claro en las diferencias entre las dos. (5%)
- 5) Explique clara y brevemente la utilidad los paquetes en el API de Java (5%).

**II (70%) Práctico: diagrama y código:**

Se requiere realizar un programa que permita mostrar unas figuras en pantalla con diferentes características. Previamente se ha iniciado el trabajo, pero lo han dejado abandonado. Su trabajo consiste en corregir y completar el diagrama de clases y el código (descargue de Moodle el código inicial) para que permita realizar lo siguiente:

Deben existir 4 tipos de objetos: rectángulos, triángulos, cuadrados y círculos, que deben dibujar la figura correspondiente en la ventana. De acuerdo a su tipo cada elemento puede tener diferentes atributos para pintarlo, pero todos deben heredar de la clase Figura.

Los triángulos y círculos son figuras arrastrables y como tal deben implementar la interfaz apropiada; así mismo, los cuadrados y rectángulos son figuras animables y como tal también deben implementar la interfaz apropiada. Tenga en cuenta que la clase abstracta Figura ya cuenta con algunos métodos.

Utilizando colecciones, realice lo siguiente en Processing (Lea todo y mire el diagrama antes de empezar a desarrollarlo):

- Diagrama de clases: (23%)
  - Diagrama de clases terminado y ajustado acorde al código resultante y al enunciado.
- Inicialización: (22%)
  - Definir un lienzo de 400 x 400.
  - Crear las clases de los tipos pedidos anteriormente, con los atributos y herencias necesarias.
  - Definir cómo se va a pintar cada elemento.
  - Crear dos objetos de cada clase (ocho en total) y almacenarlos en una sola colección.
    - Los objetos deben estar ubicados en la parte inferior del lienzo inicialmente.
- Visualización: (8%)
  - Mostrar los objetos en la posición actual
    - Solamente se deben mostrar los que existen actualmente en sus posiciones actuales, es decir que si se borran deben dejar de aparecer.
- Arrastre: (11%)
  - Al hacer clic sostenido sobre un elemento arrastrable, debe moverse con el mouse.
    - Solamente para una figura Arrastrable (se debe usar la interfaz), debe cambiar de posición, siguiendo el movimiento del mouse.
- Animación: (11%)
  - Al pasar sobre un objeto animable y hacer clic se debe animar (cambiar de posición).
    - Solamente para una figura Animable (se debe usar la interfaz), debe pasar aleatoriamente a otra posición X, Y.

**Restricciones:**

- La clase Figura proporcionada debe seguir siendo abstracta.
- No se deben crear nuevos métodos en la clase Figura.
- Los atributos de la clase figura deben seguir siendo privados y deben usarse para pintarla.
- Se deben utilizar una sola colección para almacenar los elementos.