

UNIVERSIDAD DE COSTA RICA

Sede de Occidente Puntos totales: 45
Profesor: Denis González Herrera.
IF-1300 Introducción a la Computación e Informática

Simulacro Examen
Porcentaje: 00%
Tiempo probable: 1 hora

INSTRUCCIONES GENERALES.

- La prueba es **individual**, por lo que cualquier infracción a esta regla anularía su examen.
- El estudiante es responsable de guardar continuamente el examen y tener respaldos en caso de un corte de corriente, ya que de perder el examen no se podrá reponer.
- Para entregar las respuestas deberá crear una carpeta con el siguiente nombre: **apellidoNombre**; dentro de la carpeta se debe copiar el proyecto con todos los archivos. Deberá subir a mediación virtual la carpeta comprimida (**.ZIP**). (De no realizar el nombre correcto se descontarán 3 puntos)
- Dentro de cada **clase** del ejercicio, en la primera línea se debe **comentar** la **información del estudiante** (nombre completo y carné). (De no realizar el nombre correcto se descontarán 3 puntos)
- Los ejercicios del examen se debe adjuntar como un archivo en mediación virtual antes de finalizar el tiempo.

Enunciado y descripción del problema (45 puntos): El juego consiste en que dos jugadores **tienen una moneda, tienen un nombre y tienen un puntaje** cada uno. cada jugador tira su moneda al aire, un jugador juega con la cara del escudo y el otro jugador juega con la cara de la corona. Cuando las dos monedas caen con la misma cara, el jugador que juega con esa cara gana un punto, en caso contrario ninguno gana puntos. El juego permite jugar las veces que sea necesario hasta que se decida salir. El juego debe mostrar el nombre del jugador, su puntaje y la cara de la moneda con la que juega y si ganó o perdió en cada turno. El final se muestra el puntaje total de ambos jugadores, quien ganó y quien perdió el juego.

REQUERIMIENTOS DEL PROGRAMA

1. **Clase GestorES:** este objeto se encarga de mostrar y solicitar datos al usuario según sea necesario, debe utilizar la librería JOptionPane. Además tiene los métodos constructores, set, get y toString. **La clase GestorES es igual que la clase ESCRITOR Y LECTOR pero en una misma clase.** Todos los mensajes de entradas y salidas se deben realizar únicamente con la clase GestorES.
2. **Clase JuegoMoneda:** esta clase tiene toda la **lógica del juego** dentro del método **iniciarJuego**, este método se encarga de mostrar los mensajes o menú del juego, solicitar datos al usuario, controlar los turnos del juego y mostrar los resultados del juego, entre otras cosas. Además, **NO** es necesario programar los métodos set, get y toString de este objeto.
3. **Clase PruebaJuego:** esta clase tiene el método main y debe crear una instancia del GestorES y del objeto **JuegoMoneda**. La instancia del objeto GestorES deben ser enviado por parámetros al objeto juegoMoneda.

4. **Clase Jugador:** este objeto tiene el **nombre** del jugador, el **puntaje** del jugador y también tiene una **Moneda**. Además tiene los métodos constructores, set, get y toString.
5. **Clase Moneda:** este objeto tiene el atributo caraActual que representa el valor escudo o corona (estos valores son representados por dos **constantes**: ESCUDO Y CORONA). Además, tiene un método llamado lanzar que permita simular la acción de lanzar una moneda al aire y obtener alguno de los dos valores al azar. Por último, tiene los métodos constructores, set, get y toString.

¡IMPORTANTE:

1. Se debe programar únicamente los objetos **PruebaJuego y JuegoMoneda**. Todos los demás objetos no se pueden modificar y se deben utilizar tal y como fueron entregados (GestorES, Moneda, Jugador).
2. El estudiante debe cambiar el nombre del proyecto por: APELLIDO_NOMBRE (de lo contrario se descontarán 3 puntos)
3. El estudiante debe colocar al inicio de todas las clases un comentario con su nombre completo y carné

Evaluación: el programa debe estar **libre de errores** de sintaxis para poder ser evaluado, si presenta errores de compilación será calificado **con 0**.

Evaluación del ejercicio	Valor	Obtenidos
Tipos y nombre correctos de (variables, constantes, clases y métodos)	3	
Uso correcto de la clase GestorES	5	
Uso correcto de la Clase Jugador	5	
Uso correcto de la Clase Moneda	5	
Uso correcto de la Clase JuegoMoneda	15	
Uso correcto de la Clase PruebaJuego	2	
Solución correcta del enunciado y acorde con los requerimientos	10	
Resultado:	45 pts	