

# **Módulo Entorno de desarrollo. 1º DAM.**

PROYECTO FINAL: Diagrama de clases.

UNIDAD DE TRABAJO 6.

Profesor: Manuel Jesús Rico Ruiz.

# 1. Introducción.

## 2. Materiales.

1. NetBeans, VSC u ArgoUML.

### 2.1. Diseñando diagrama de clases.

Se desea implementar el juego del solitario, modelar el diagrama de clases considerando que:

Se tiene una baraja en la que se encuentran las cartas tapadas.

También se tiene un montón de cartas en la que se van apilando las cartas que se destapan de la baraja, en este caso, destapadas.

En el juego también se tiene siete montones cada montón con una carta destapada en la cima y n-1 cartas tapadas.

Por último, se tiene cuatro montones, inicialmente vacíos, en las que se colocan cada uno de los 4 palos de la baraja, en la que se colocan empezando desde el uno hasta el rey cada una de las cartas, que pueden estar en los montones inferiores o en el montón de la baraja.

Funcionamiento del juego.

Se inicia o reinicia. Esto implica barajar, repartir en los 7 montones las cartas, colocando la que está en la parte superior descubierta, vaciar el montón de descubiertas y el montón de palos.

El usuario puede:

- Destapar una carta de la baraja, si se queda sin cartas, se vuelven a amontonar en el orden inicial.
- Mover una carta del montón destapado a montones inferiores, o al montón de palos siempre que sea posible.
- Mover una carta de montones inferiores a los de palos, siempre que sea posible.
- Mover un conjunto de cartas entre montones inferiores si es posible.
- Combinaciones de las acciones anteriores.

Para mover una carta del montón destapado a los montones inferiores:

- El montón inferior tiene que estar vacío, y la carta ha de ser un rey.
- La carta de la cima del montón inferior tiene que ser un número mayor que la carta que se desea mover y de diferente color.

Para mover una carta del montón destapado a los montones de palos:

- Tiene que estar vacía y ser un As/uno.
- Tiene que ser del mismo palo que la primera carta del montón inferior y ser la siguiente, si es el siete de corazones la carta superior, se puede colocar el ocho de corazones.

Para mover conjuntos de cartas entre montones inferiores:

- El destino ha de estar vacío y el origen ha de ser un rey, moviéndose todas las cartas que se encuentren encima del rey.
- No se encuentra vacío, pero el destino es un número inferior a la carta que se desea mover, se desplazan también las cartas que estén por encima de la de destino.

El juego finaliza cuando:

- No existen más movimientos posibles.
- Se han completado todos los montones de palos.

Paso 1.

Indicar las clases, enumerados, clases abstractas, interfaces...

Paso 2.

Establecer las relaciones entre clases: Herencia, asociación, composición, agregación, realización, dependencia.

Paso 3.

Detallar los atributos y métodos más destacados.

Paso 4.

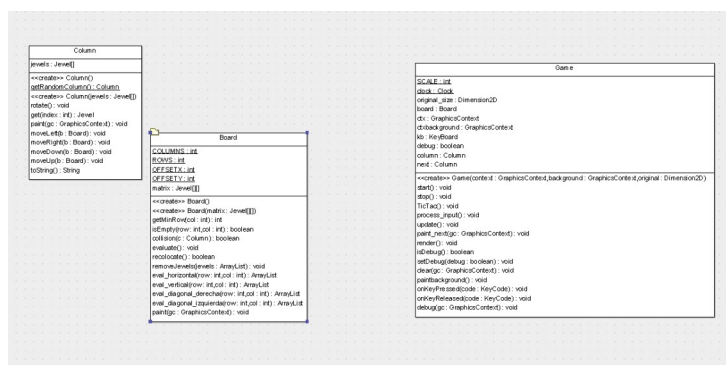
Si se ha creado el UML con un programa que permite crear código, generarlo.

Preguntas.

- ¿En qué se transforma una relación uno a muchos?
- Si se representan cardinalidades que no sean 0,1,\* ¿Cómo se implementa al generar el código de forma automática?
- ¿Cómo se implementa la herencia en el código? Relacionar con el módulo programación.
- ¿Existe alguna diferencia en cuanto a la implementación de la asociación, composición y agregación? ¿Cuál es la razón?
- ¿Si una clase depende de una interfaz, a esta relación se le denomina? ¿Cómo se representa?

## **2.2. De código a UML.**

## Representar también los paquetes.



### 3. Entrega.

La práctica se entrega comprimida, con los proyectos limpios, y la memoria y preguntas en un fichero en formato PDF, en el campus virtual [ww.aules.edu.gva.es](http://ww.aules.edu.gva.es). El documento de entrega ha de tener los siguientes puntos.

1. Portada. (Título de la práctica y autor).
2. Introducción.

3. Desarrollo de la práctica. **Capturas de pantalla, respuestas a las preguntas y comentarios.**
4. Conclusiones. (Pequeño comentario sobre la práctica: dificultad, problemas encontrados...).

## **4. Evaluación.**

**Se puede realizar una corrección de forma presencial donde el profesor preguntará cuestiones sencillas sobre la práctica para comprobar la autoría de la misma.**

RA5. Genera diagramas de clases valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.

CE5a. Se han identificado los conceptos básicos de la programación orientada a objetos.

CE5b. Se ha instalado el módulo del entorno integrado de desarrollo que permite la utilización de diagramas de clases.

CE5c. Se han identificado las herramientas para la elaboración de diagramas de clases.

CE5d. Se ha interpretado el significado de diagramas de clases.

CE5e. Se han trazado diagramas de clases a partir de las especificaciones de las mismas.

CE5f. Se ha generado código a partir de un diagrama de clases.

CE5g. Se ha generado un diagrama de clases mediante ingeniería inversa.