

Tarea obligatoria UD5 – Parte I – Introducción a Python

Objetivo: Crear un paquete en Python para jugar al Blackjack.

En este juego de naipes gana quien más se acerque a 21 sin pasarse.

Paso 1. Clase Carta:

1. Crea la clase Carta con los siguientes atributos: valor (2-10, J, Q, K, A) y palo (♠, ♥, ♦, ♣).
2. Utiliza properties para acceder y modificar los atributos valor y palo.
3. Implementa un método **str** para imprimir la representación de la carta.
4. Implementa los métodos de comparación para comparar los valores de las cartas

Paso 2. Clase Mazo:

1. Crea la clase Mazo con un atributo cartas que contendrá una lista de las 52 posibles instancias de la clase Carta.
2. Implementa un método repartir_carta que devuelva una carta del mazo y la elimine de la lista.
3. Implementa un método barajar que mezcle las cartas del mazo (solo puede ser llamado antes de comenzar a repartir cartas la primera vez).

Paso 3. Clase Jugador:

1. Crea la clase Jugador con los siguientes atributos:
 - nombre: el nombre del jugador.
 - mano: lista de cartas que el jugador tiene en la mano.
2. Utiliza properties para acceder y modificar el atributo nombre.
3. Implementa un método recibir_carta que reciba una carta y la agregue a la mano del jugador.
4. Implementa un método calcular_puntuacion que calcule la puntuación actual del jugador en base a las cartas en su mano.
 - Para ello ten en cuenta que del 2 al 10 las cartas valen su valor, las figuras (J, Q y K) valen 10 y el as (A) vale 1 u 11 (lo que mejor convenga al jugador).
5. Implementa un método **str** para imprimir la información del jugador y sus cartas.
6. Implementa los métodos de comparación para comparar el valor de las manos de dos jugadores.

Paso 4. Clase Blackjack:

1. Crea la clase Blackjack con los siguientes atributos:
 - mazo: una instancia de la clase Mazo.
 - jugadores: una lista de instancias de la clase Jugador.
2. Implementa un método repartir_cartas_iniciales que reparta dos cartas a cada jugador al inicio del juego.
3. Implementa un método jugar que permita a cada jugador realizar turnos para pedir cartas adicionales.
4. Implementa un método **str** para imprimir la información del estado actual de la partida en juego.
5. Implementa la lógica para determinar si un jugador tiene un Blackjack o se ha pasado de 21.
6. Al final del juego, muestra la puntuación final de cada jugador.

Paso 5. Realiza las modificaciones necesarias para cumplir los siguientes requisitos de la versión final del juego:

1. El número de jugadores será como mínimo 2 y como máximo 5.
2. El último jugador será el croupier, este recibe las dos cartas iniciales quedando una de ellas oculta hasta que el resto de los jugadores haya terminado, momento a partir del cual jugará de forma automática mientras su mano no mejore la de los jugado:
 - Está obligado a plantarse si su mano vale 17 o más
 - Está obligado a pedir carta si su mano vale 16 o menos
3. Implementa un método repartir_cartas_iniciales que reparta dos cartas a cada jugador al inicio del juego.
4. Implementa un método jugar que permita a cada jugador realizar turnos para pedir cartas adicionales.
5. Implementa un método **str** para imprimir la información del estado actual de la partida en juego.
6. Implementa la lógica para determinar si un jugador tiene un Blackjack o se ha pasado de 21.
7. Al final del juego, muestra la puntuación final de cada jugador.

Criterios de evaluación:

- La lógica se evalúa sobre 8 puntos, teniendo en cuenta:
 - Ajustarse a los requisitos del enunciado
 - Ausencia de errores al ejecutar
- Documentación sobre 2 puntos. Deben documentarse como mínimo las funciones, las clases y los módulos.

Entrega:

Se entregará un único archivo zip con el paquete Python. Las rutas deben ser relativas a la hora de importar para que el profesor no tenga que modificar el código para probarlo.