

LIBRERÍA DE LA BARAJA. DOCUMENTACIÓN

Nomenclatura	2
Idioma	2
Formas de escribir el código	2
Card	2
Enums	2
Atributos	2
Constructores	2
Métodos	3
Deck	3
Atributos	3
Constructores	3
Métodos	4

Nomenclatura

Idioma

Inglés para todo lo que no sea comentarios.

Formas de escribir el código

[Convención de C#](#)

Card

Clase de la carta.

Enums

```
private enum ESuits { Diamonds, Hearts, Spades, Clubs }
```

Palos de las cartas.

```
private enum ENums { A = 1, Two, Three, Four, Five, Six, Seven, Eight, Nine, Ten, J, Q, K }
```

Números de las cartas.

Atributos

```
private ESuits suit
```

Enumeración de los palos;

```
private ENums num
```

Enumeración de los números;

Constructores

```
public Card()
```

Constructor por defecto.

```
public Card(int suit, int num)
```

Constructor definido. Define el palo (suit) y el número (num).

Métodos

```
public int GetSuit()
```

Devuelve el palo de la carta

```
public int GetNum()
```

Devuelve el número de la carta.

```
private string GetSymbol()
```

Devuelve el símbolo que corresponde a cada palo.

```
public string GetCharacter()
```

Devuelve del “Two” al “Ten” de ENums en formato numérico y el resto en letra.

```
public string GetAsciiCard()
```

Devuelve el objeto Card en forma de dibujo *ASCII*.

```
public string[] GetArrayAsciiCard()
```

Devuelve un array de strings para poder dibujar cartas en horizontal usado adecuadamente.

Deck

Clase que contendrá una baraja de cartas.

Atributos

```
List<Card> cards;
```

Lista de Card.

Constructores

```
public Deck()
```

Constructor por defecto que rellena la lista de cards.

Métodos

```
private int Rand(int min, int max)
```

Devuelve un número aleatorio entre min y max - 1.

```
public Card DrawBottom()
```

Devuelve la última carta de la baraja eliminandola de esta. Devuelve NULL si no quedan cartas en la baraja.

```
public Card[] DrawBottom(int num)
```

Devuelve un array de cartas con el número de cartas especificado en el parámetro robando desde la parte inferior de la baraja. Si se piden más cartas de las que hay en la baraja, se devuelven las que haya en la baraja.

```
public Card DrawRandom()
```

Devuelve una carta aleatoria de la baraja eliminandola de esta. Devuelve NULL si no quedan cartas en la baraja.

```
public Card[] DrawRandom(int num)
```

Devuelve un array de cartas con el número de cartas especificado en el parámetro robando aleatoriamente de la baraja. Si se piden más cartas de las que hay en la baraja, se devuelven las que haya en la baraja.

```
public Card Draw()
```

Devuelve la primera carta de la baraja eliminandola de esta. Devuelve NULL si no quedan cartas en la baraja.

```
public Card[] Draw(int num)
```

Devuelve un array de cartas con el número de cartas especificado en el parámetro robando de la parte superior de la baraja. Si se piden más cartas de las que hay en la baraja, se devuelven las que haya en la baraja.

```
private Card[] RemoveNull(Card[] cardSet)
```

Devuelve un array de cartas que ha sido redimensionado hasta el tamaño exacto para no tener null en ninguna posición. Utilizado en los métodos sobrecargados Draw, DrawRandom y DrawBottom por si se piden más cartas de las que hay en la baraja.

```
public void Shuffle()
```

Desordena la baraja.

```
public void PrintCards(Card[] cardSet)
```

Muestra tantas cartas como posiciones tenga cardSet en horizontal.