

Nom:	Ús
Curs:	



Temps estimat

3 hores

Tipus d'activitat

Avaluació



Normes

- El codi lliurat ha de compilar sense problemes
 - De cada exercici en tindrà la pròpia nota, s'especificarà la nota mínim. Si aquesta nota mínima no s'assoleix, la nota de l'examen serà 4 com a màxim.
 - De cada exercici s'ha de proporcionar el main que has fet servir per veure el funcionament correcte. Si aquest main és escàs i es deixa porcions d'algoritmes sense cobrir baixarà la nota. Si és massa escassa, la nota màxima serà de 4.
 - L'estructura de cada fitxer serà el següent (si no, no es corregirà l'exercici)
 - enums? delegats?
 - Una classe
 - Classes noves dins de la classe
 - Atributs
 - Propietats
 - Constructors
 - Destructors
 - Getters i Setters
 - Altres mètodes
- Si cometeu algun dels errors comentats a classe (que són majoritàriament 3) que no es poden permetre, la nota màxima de l'examen serà de 4.
 - No pots fer trampes



(1 punt) (Nota mínima 7) Implementa la funció següent: Implementa una funció que, donat un número “n”, torna una llista amb els n primers números de la sèrie de fibonacci.

(2 punts) (Nota mínima 7) Implementa la funció següent.

Una funció anomenada Merge, que rebi dos arrays d'enters que estan ordenats i torneu un array ordenat amb el contingut dels dos arrays d'entrada. Prohibit fer servir algoritmes d'ordenació, ni de C#, ni teus. L'algorisme s'explicarà a la pissarra.

(2 punts) (Nota mínima 4) Implementa la classe següent: És una classe anomenada User que ha de gestionar una llista de claus (una clau és un string). Quan provis aquesta classe, al main posa el codi necessari per capturar les excepcions.

- **Atributs**
 - Els hauràs de definir tu mateix/a.
- **Propietats**
 - Name (El getter tornarà el nom en majúscules)
 - Code: S'obté sumant tots els caràcters del nom més l'edat multiplicat per 33
 - Age (sólo get)
 - KeyCount
- **Constructors**
 - Un únic constructor que tingui el nombre màxim de claus que podeu emmagatzemar i ledat. Si ledat és menor o igual que zero, cal llançar una excepció. Si el nombre de claus màxim és menor que 1 cal llençar una excepció
- **Mètodes**
 - Els que necessitis
 - AddKey(string) Pren la decisió que creguis convenient aquí
 - ContainsKey(cadena)
 - Elimina les claus (delegat)
 - ClearKeys

(2 punts) (nota mínima 3) Implementa la classe següent, posa els atributs, constructors i mètodes que vegis necessaris. La classe que cal implementar és un pool d'objectes que implementa una interfície, és a dir, cal fer una interfície i la classe. Entre el codi a desenvolupar que has de fer: ToArray, Clone i Count. És possible que hakis de fer servir “where T : class”. El constructor de la classe serà privat, si no, no es corregirà l'exercici, ia la interfície hi haurà una funció de classe (estàtica) anomenada CreatePool que crearà un pool.



(3 punts) (nota mínima 3) Es vol fer una aplicació que és un simulador del joc de l'oca amb alguna restricció especial.

- **Prohibit utilitzar funcions que afegixin elements a un array, tant teves com del sistema.**
- **Prohibit fer càstings.**
- El joc (la classe Joc) constarà d'una llista de caselles jugadors i un **array de caselles**. Tindrà almenys els mètodes següents
 - Afegir jugador: li passo un jugador
 - Esborrar jugadors: Deixarà el joc sense jugadors
 - Crear tauler. Crearà el tauler per jugar. Aquesta funció ocuparà com a màxim 20 línies.
 - *Pista: Cada cop que crees un tauler, comença creant un array amb fitxes normals*
 - Ordenar jugadors (privada): Abans de començar la partida cal determinar l'ordre dels jugadors. Cada jugador llança els daus i comencen en ordre d'aquest resultat.
 - En cas d'empat al dau de sortida, desempataran fent daus de nou fins que un aconsegueixi més que l'altre.
 - Cada jugador comença a la casella 1
 - Simular: tornarà el jugador que ha guanyat. Abans de començar cal determinar l'ordre de sortida.
- Les caselles heretaran d'una classe abstracta, i només podran tenir un únic atribut que és el número (començant per 1) que li passarà pel constructor. Les caselles són:
 - 63: Casella guanyadora. Si un jugador cau aquí, guanya la partida. Si es passa, traurà a la casella 55.
 - 6, 12, 18, 24, 30, 36, 42, 48, 54 i 60: D'oca a oca. Si un jugador cau aquí, saltarà a la casella següent i torna a tirar.
 - 8 i 14: De pont a pont. Si un jugador cau en una casella, passarà a l'altra casella i torna a llançar.
 - 13, 26, 39 i 52: Caselles de càstig. Si un jugador cau aquí esperarà 1, 2, 3 o 4 tornos sense jugar.
 - 27 i 53: De dau a dau: Si un jugador cau a la 26, passarà a la 53 i li torna a tocar
 - 58: La mort. Si un jugador cau aquí va a la casella de sortida.
- Els jugadors. Cal implementar diversos tipus de jugadors que heretaran tots de la classe Jugador.
 - Un jugador té:
 - Nom que no podeu canviar.
 - Altres aspectes que considereu
 - Tipus de jugadors
 - Jugador normal. Cada cop que li toqui llançarà un dau de 6 cares que serà el nombre de caselles a avançar.
 - Jugador ràpid. Cada cop que li toqui, llançarà entre 1 i 3 daus de 6 cares i es quedarà amb el més alt. El nombre de daus que llançarà se li passarà pel constructor.
 - Jugador trampós. Llançarà un dau de 8 cares per avançar.