

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

## 2. feladat

40 pont

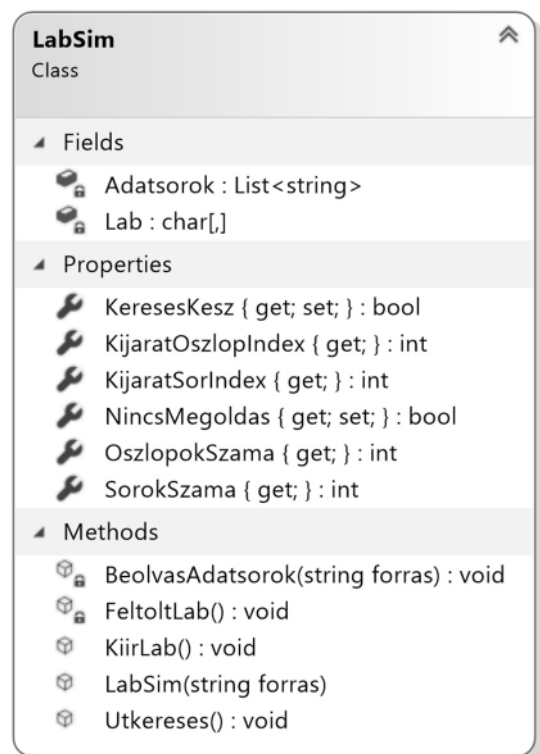
### Labirintus

```

XXXXXXXXXXXXX
.....X
X.....X..X
X.....X..X
X..XX..X..X
X.....XXX
X.....X
X..X.....X
X..X.....X
X..X..XXXXX
X.....X
X..XX.....
XXXXXXXXXXXXX
    
```

A következő feladatban egy szöveges állományban labirintust kódoltunk karakterekkel, mátrix-szerűen. A labirintus méretét nem ismerjük, de feltételezheti, hogy maximum 100x100-as méretű. Az „X” karakterek jelölik a labirintus külső és belső falait. Szóköz karakterrel kódoltuk a bejárható útvonalakat. A külső fal mindig két helyen van megszakítva: balra fent van a bejárat, jobbra lent a kijárat a függőleges falakban a bal oldali minta szerint. A mondatszerű leírással **megadott** algoritmus kódolásával és hívásával kell megtalálnia egy útvonalat a kijárhoz. Az algoritmus működését a feladatleírásban leírt módszerrel kell szemléltetnie. Megoldásában vegye figyelembe a következőket:

- *Megoldását elkészítheti saját osztály definiálása és alkalmazása nélkül is, de úgy az nem lesz teljes értékű.*
- *A képernyőre írást igénylő feladatok eredményének megjelenítése előtt írja a képernyőre a feladat sorszámát (például: 5. feladat)!*
- *Az egyes feladatokban a kiírásokat a minta szerint készítse el!*
- *Az ékezetmentes kiírások is elfogadottak.*
- *Az azonosítókat kis- és nagybetűkkel is kezdheti.*
- *A program megírásakor az állományokban lévő adatok helyes szerkezetét nem kell ellenőriznie, feltételezheti, hogy a rendelkezésre álló adatok a leírtaknak megfelelnek.*
- *A megoldását úgy készítse el, hogy az azonos szerkezetű, de tetszőleges bemeneti adatok mellett is helyes eredményt adjon!*



1. Készítsen **konzolos alkalmazást** a következő feladatok megoldására, melynek projektjét Labirintus néven mentse el!
2. Projektjében hozzon létre saját osztályt LabSim azonosítóval, melynek kód- és adattagjainak azonosítóit és láthatósági szintjét az osztálydiagram szemlélteti! A privát adattagokat és metódusokat egy lakat szimbólum különbözteti meg a publikusaktól.

*A feladat a következő oldalon folytatódik*

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

3. A LabSim osztály konstruktora kapja paraméterül a forrásállomány nevét! A konstruktor töltse be az Adatsorok azonosítójú, tetszőleg típusú adatszerkezetbe a forrásállomány adatait a BeolvasAdatsorok() metódus hívásával! A következő (4.) feladat megoldása után, ugyancsak a konstruktorban inicializálja a Lab azonosítójú karaktermátrixot a megfelelő mérettel! Az inicializálás után töltse fel a FeltoltLab() metódus hívásával „X” és szóköz karakterekkel az Adatsorok adattagot használva forrásként!
4. Határozza meg a beolvasott mátrix (labirintus) méretét a SorokSzama és OszlopokSzama jellemzőkkel/függvényekkel! Ugyancsak határozza meg a kijárat cellájának indexét a mátrixban a KijaratSorIndex és KijaratOszlopIndex jellemzőkkel/függvényekkel! Feltételezheti, hogy a kijárat és a bejárat mindig azonos pozícióban van. A bejárat a második sor első oszlopában, a kijárat az utolsó előtti sor utolsó oszlopában. Az oszlopok és sorok indexelését 0-val kezdjük.
5. A főprogramban hozzon létre egy osztálypéldányt (objektumot) a LabSim osztályból, forrásként a Lab1.txt állományt adja meg, majd jelenítse meg a beolvasott labirintus adatait!
6. Készítsen metódust a Lab mátrix (labirintus) megjelenítéséhez KiirLab azonosítóval, majd hívásával jelenítse meg a beolvasott labirintust a képernyőn!
7. Kódolja a LabSim osztályban az Utkereses() metódust a következő mondatszerű leírás alapján:

```
Metódus Utkereses()
    KeresesKesz:=hamis
    NincsMegoldas:=hamis
    változó egész r:=1    //segéd, a vizsgált cella sorindexe
    változó egész c:=0    //segéd, a vizsgált cella oszlopindexe
    Ciklus_ámíg (NEM KeresesKesz ÉS NEM NincsMegoldas)
        Lab[r,c]:='O'
        Ha (Lab[r,c+1]=' ') c++    //jobbra lépünk
        Egyébként Ha (Lab[r+1,c]=' ') r++    //lefelé lépünk
        Egyébként
            Lab[r,c]:='- '    //erre többet nem jövünk
            Ha (Lab[r,c-1]='O') c--    //visszalépés balra
            Egyébként r--    //visszalépés felfelé
        KeresesKesz:= r=KijaratSorIndex ÉS c=KijaratOszlopIndex
        Ha (KeresesKesz) Lab[r,c]:='O'
        NincsMegoldas:= r=1 ÉS c=0
    Ciklus vége
Metódus vége
```

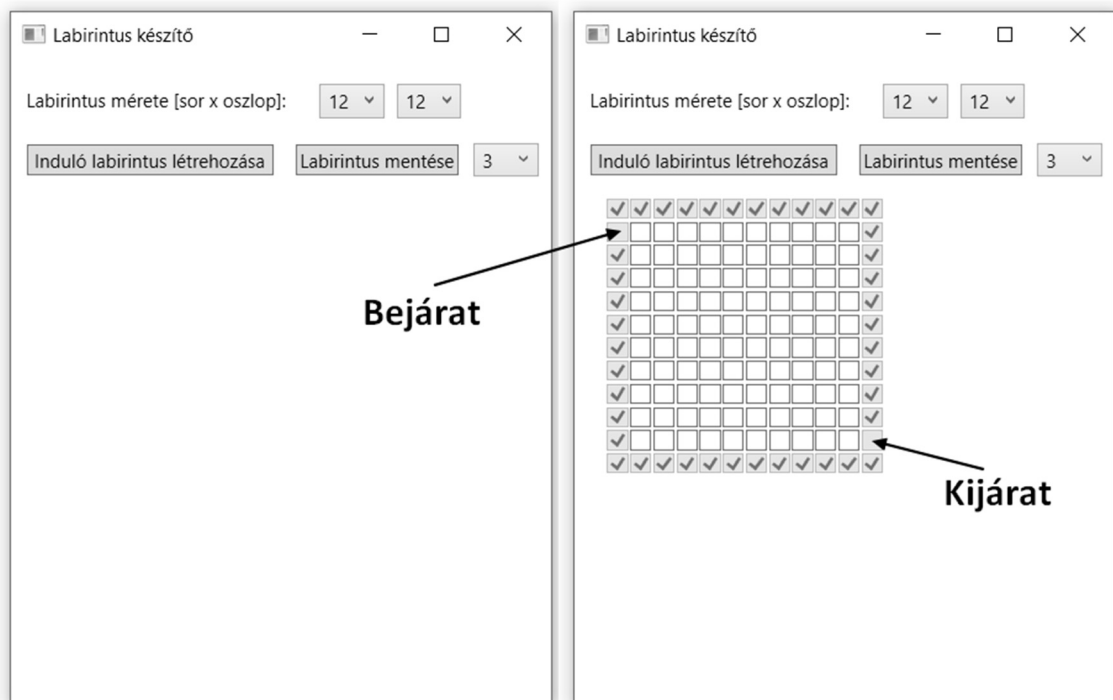
*A fenti, terjedelmi okokból egyszerűsített metódus hiányossága, hogy csak olyan labirintusokban találja meg a kijáratot, ahol jobbra és lefelé haladással és ellenkező irányú visszalépésekkel az megkereshető. A megtalált útvonalat az „O” karakterek jelölik a mátrixban, a kötőjel a már bejárt, helyes útvonalnak alkalmatlan cellák jelölésére szolgál.*

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

8. Az `Utkereses()` metódus bővítésével és hívásával készítsen az algoritmus működését bemutató szimulációt, melyben minden ciklusmag lefutása után megjeleníti a `Lab` mátrixot a `KiirLab()` metódus hívásával! A szimulációt várakozások (billentyűleütés, eltelt idő) és/vagy törlések beiktatásával tegye követhetővé! A szimuláció lefutása után jelenítse meg az útkeresés eredményét!

9. Készítsen **grafikus alkalmazást**, melynek a projektjét `LabirintusGUI` néven mentse el!

A grafikus alkalmazásban a következő feladatokat végezze el:

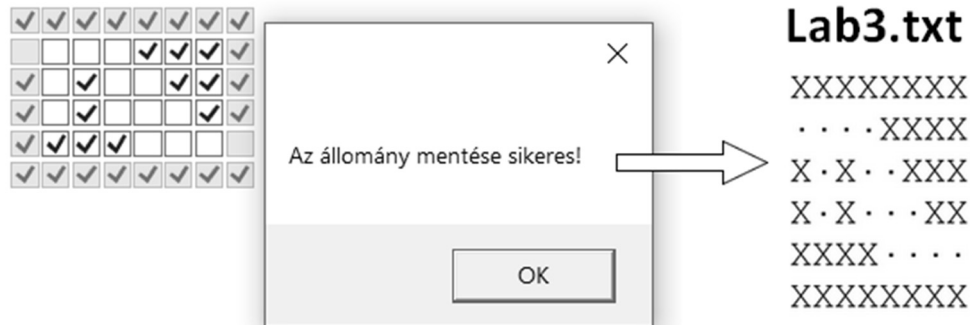


- Alakítsa ki a felhasználói felületet a fenti, bal oldali minta szerint! Az alkalmazás címsorában megjelenő felirat „Labirintus készítő” legyen! A mátrix méretét a legördülő listák segítségével lehessen kiválasztani, melyek értékeit töltsé fel az alkalmazás indulásakor 5-20 számokkal! Oldja meg, hogy mindkét listában a 12-es érték legyen az alapértelmezett! A „Labirintus mentése” parancsgomb melletti listában legyen beállítható az állománynévben szereplő index 1-16 között, az alapértelmezett érték 3 legyen! Ez utóbbi lista is az alkalmazás indulásakor legyen feltöltve!
- Az „Induló labirintus létrehozása” feliratú parancsgomb lenyomása után hozzon létre programjával jelölőnégyzet (`CheckBox` példányokat) mátrix-szerűen elrendezve a kiválasztott méretnek megfelelően a fenti, jobb oldali minta szerint! A labirintus külső falait jelölje automatikusan inaktív állapotú, bekapcsolt jelölőnégyzetekkel! A bejárat és a kijárat jelölőnégyzetei szintén inaktívak, de ne legyenek kijelölve a labirintus létrehozásakor! Ha korábban már volt létrehozott jelölőnégyzet-mátrix (labirintus) a felhasználói felületen, akkor a létrehozás előtt törölje azt!

*A feladat a következő oldalon folytatódik*

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

- c. A „Labirintus mentése” parancsgomb lenyomása után hozzon létre szöveges állományt, melyben a labirintus sorait kódolja! Az állomány azonosítója `Lab{index}.txt` legyen, ahol az `index` értéke (1-16) a legördülő listából kerüljön beszúrára! Az üres jelölőnégyzeteket szóközzel, a kipipált állapotúakat „X”-szel jelölje a következő minta szerint! Sikeres mentés esetén jelenítse meg a mintán látható üzenetet! Ha a mentéskor hiba jelentkezik, akkor a hiba szövege jelenjen meg a felugró ablakban!



### Konzolos alkalmazás minta:

```

5. feladat: Labirintus adatai
    Sorok száma: 12
    Oszlopok száma: 12
    Kijárat indexe: sor:10 oszlop:11
6. feladat: A labirintus
XXXXXXXXXXXXX
                X
X      X  X
X      X  X
X  XX  X  X
X      XXX
X  X      X
X  X      X XX
X  X  XXXX X
X      X
X  XX
XXXXXXXXXXXXX
    
```

A szimuláció néhány kiválasztott, egymást fedő lépései a 8. feladatban (az utolsó minta a `Lab2.txt` állomány adataival készült):

XXXXXXXXXXXXX	XXXXXXXXXXXXX	XXXXXXXXXXXXX	XXXXXXXXXXXXX
00000000000X	000000000--X	000000-----X	0--X      X
X      X  OX	X      XO--X	X      0-X---X	X--X      X  X
X      X  X	X      XO--X	X      0-X---X	X--X      X  X
X  XX  X  X	X  XX  XO--X	X  XX0-X---X	X--X      X  X
X      XXX	X      OXXX	X      0---XXX	X-----X  XX
X  X      X	X  X      O  X	X  X  0-----X	X--XXXXX  X
X  X      X XX	X  X      X XX	X  X  0--X-XX	X--X      XX
X  X  XXXX X	X  X  XXXX X	X  X  OXXXX X	X--XXXXXXXXX
X      X	X      X	X      000000X	X--X      X
X  XX	X  XX	X  XX      00	X-----X
XXXXXXXXXXXXX	XXXXXXXXXXXXX	XXXXXXXXXXXXX	XXXXXXXXXXXXX

Útvonal megtalálva! Nincs megoldás!