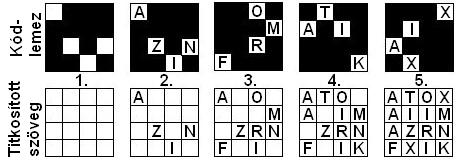
**Forgó rács 2020[[1]](#footnote-1)**

A forgó rács egy titkosítási módszer, melyet az I. világháborúban táviratoknál és tábori telefonoknál használtak. Ebben a feladatban egy 8x8-as táblázatot fogunk használni a szöveg titkosítására. A táblázat fölé egy kódlemezt helyezve a lemezen található „ablakokba” egy‑egy betűt írunk, **felülről lefelé haladva**, majd az oszlop végén a következő, jobbra lévő oszlopban folytatjuk a betűk írását. A kódlemezen 16 ablak található. Ha minden ablakba betű került, akkor a kódlemezt 90°-al balra elforgatva folytatjuk a betűk írását a megismert szabályok alapján. Ezzel a módszerrel a kódlemezt négyszer elforgatva maximum 4x16, azaz 64 betűből álló üzenet titkosítható.

A következő példában egy 4x4-es táblázatban titkosítjuk egy 4 ablakos kódlemez felhasználásával „AZ INFORMATIKAI.” szöveget. A titkosítást a szöveg átalakításával kezdjük. A szóközöket és írásjeleket eltávolítjuk az eredeti szövegből, majd a táblázat cellaszámának megfelelően (4x4) 16 karakterre egészítjük ki a titkosítandó szöveget jobbról az „X” karakterekkel: „AZINFORMATIKAIXX”. A titkosítás lépéseit a következő ábrák szemléltetik:



A továbbiakban egy **8x8-as** „forgó rács” titkosítási eljárással kapcsolatos feladatokat kell megoldania.

A megoldás során vegye figyelembe a következőket:

* *A képernyőre írást igénylő részfeladatok eredményének megjelenítése előtt írja a képernyőre a feladat sorszámát (például:* 3. feladat:*)!*
* *Az egyes feladatokban a kiírásokat a minta szerint készítse el!*
* *Az ékezetmentes kiírások is elfogadottak.*
* *Az azonosítókat kis- és nagybetűkkel is kezdheti.*
* *A megoldását úgy készítse el, hogy az azonos szerkezetű, de tetszőleges input adatok mellet is megfelelően működjön!*

1. Készítsen Forgoracs2020 néven **konzol típusú projektet**, melyben megoldja a következő feladatokat!
2. Forráskódjába tegye elérhetővé a java.txt vagy a csharp.txt állományból az Fracs osztályt definiáló kódrészletet!
3. Töltse be és tárolja egy szöveges változóban a titkosítandó szöveget a szoveg.txt állományból, majd írja ki a képernyőre!
4. Hozzon létre egy Fracs típusú osztálypéldányt (objektumot), melynek a konstruktora a kodlemez.txt forrásállomány nevét és a titkosítandó szöveget kapja aktuális paraméterként feldolgozásra!
5. Jelenítse meg a képernyőn az átalakított titkosítandó szöveget a Titkositando jellemző értékének kiírásával!
6. A KiirKodlemez() metódus hívásával jelenítse meg a Kodlemez mátrixban (adattagban) eltárolt karaktereket! Az „A” karakterek a kódlemez ablakait (16db) jelentik, ahová a titkosítandó szöveg karakterei kerülnek majd.
7. Definiáljon az Fracs osztályban metódust a következő algoritmus kódolásával!

Függvény ForgatKodlemez(): Karakter típusú mátrix

Változó ujKodlemez: Karakter típusú 8x8-as mátrix

Ciklus sor:=0-tól 7-ig egyesével

Ciklus oszlop:=0-tól 7-ig egyesével

ujKodlemez[7-oszlop, sor] = Kodlemez[sor, oszlop]

Ciklus vége

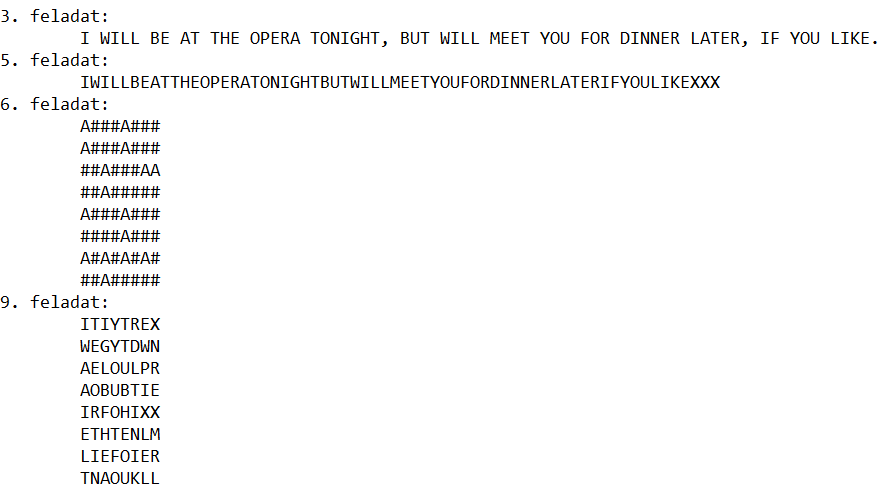
Ciklus vége

Térj vissza ujKodlemez

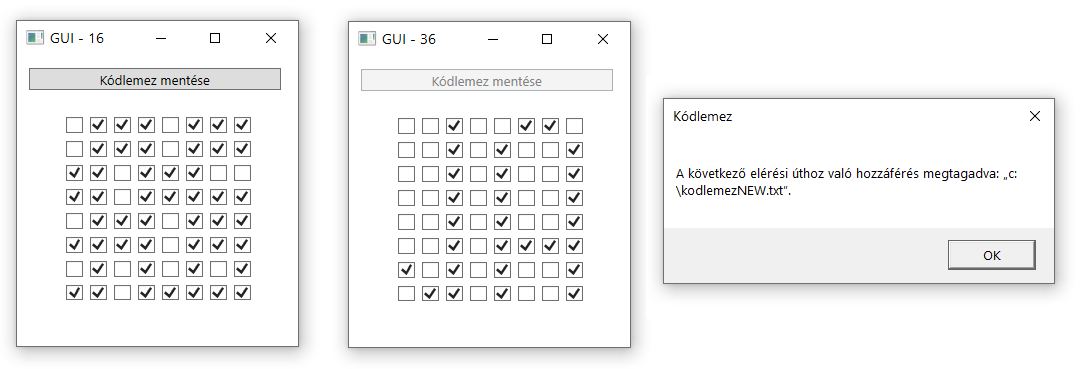
Függvény vége

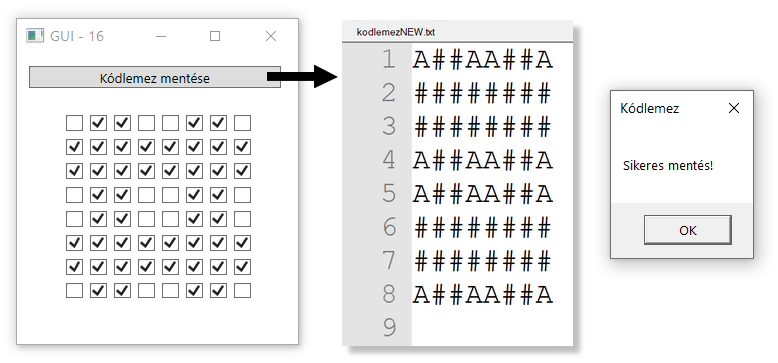
1. Készítsen az Fracs osztályban metódust Titkosit() azonosítóval, ami a **bevezetőben ismertetett eljárással** karakterenként titkosítja a Titkositando adattag (jellemző) karaktereit a Titkositott adattagba (mező) a Kodlemez adattag (mező) felhasználásával! A Kodlemez mátrixot minden 16 karakter titkosítása után (ennyi ablak van a kódlemezen) el kell forgatnia az előző feladatban definiált ForgatKodlemez() metódus hívásával! A Titkositando jellemző 64 karakterének kódoláshoz négyszer kell elforgatnia a kódlemezt!
2. Hívja meg az előző feladatban definiált Titkosit() metódust, majd jelenítse meg a képernyőn a kódolt szöveget (mátrixot) a KiirKodoltSzoveg() metódus hívásával!
3. Készítsen **grafikus alkalmazást**, mellyel 8x8‑as méretű kódlemezt tud készíteni! Az alkalmazás projektjét Forgoracs2020GUI néven mentse el!
4. Az alkalmazás grafikus felhasználói felületét alakítsa ki a minta szerint! Az ablak címsorában a „GUI ‑ 16” szöveget jelenítse meg! A 64 darab kapcsolókeretet futás közben kell létrehoznia a következő feladat szerint!
5. Hozzon létre programjával táblázat szerűen elhelyezett 8x8 kapcsolókeretet (checkbox‑ot) az indulás után, melyek aktív-inaktív (checked) állapotát a kodlemez.txt állományban elhelyezett karakterek határozzák meg! Az ablakok („A” karakterek) helyén „üres” legyen, ahol nincs a mátrixban ablak („#” karakterek), ott „kipipált” állapotú legyen a kapcsolókeret!
6. A kodlemez.txt állomány betöltése után és a kapcsolókeretek állapotának változásakor az „ablakok” számát folyamatosan jelezze a címsorban! A „Kódlemez mentése” parancsgomb csak akkor legyen használható (aktív), ha az ablakok száma 16!
7. Ha az aktív állapotú „Kódlemez mentése” parancsgombra kattintunk, akkor történjen meg a kapcsolókeretek állapota alapján a mátrix mentése a kodlemezNEW.txt állományba! Az állomány szerkezete egyezen meg a kodlemez.txt állomány szerkezetével (**minta**)! Sikeres mentés esetén a „Sikeres mentés!” felirat, ha az állomány mentése sikertelen, akkor a hibához tartozó beépített üzenet (message) jelenjen meg egy felugró ablakban!

**Minta a konzolos alkalmazás futására:**



**Minták a grafikus alkalmazás futására:**





1. Forrás: *Paul Lunde: Titkos kódok, Kossuth Kiadó 2010, 80-81.p* [↑](#footnote-ref-1)