# 30. Stringek 5.

## Titkosírás

A titkosírás (kriptográfia) lényege, hogy olyan kódolt üzenetet küldenek, amely közvetlenül nem olvasható. A címzett tudja, hogyan kell visszaállítani az eredeti üzenetet (dekódolás). Másoknak meg kell fejteni a kódolást, ha el akarják olvasni. Akkor hatékony, ha az üzenet megfejtése tovább tart, mint amennyi idő alatt az információ elévül.

A XX. század második feléig csak katonák és diplomaták használták. Ezután megjelent az üzleti életben, elsősorban a bankoknál. Manapság az interneten és a mobiltelefonoknál is nagy szerepe van az adatok titkosításának és védelmének.

Most egy egyszerű titkosírást fogunk kipróbálni. Az ATBAS egy egyszerű helyettesítő módszer a héber ábécéhez. Az ábécé első betűjét (א, alef) az utolsóval (ת, tav), a másodikat (ב, bet) hátulról a másodikkal (ש, sin) cseréli fel, és így tovább folytatva a többi betűvel, míg a teljes ábécé megfordul. Ezért kapta az elnevezését is: A→T, B→S. Az Ószövetségben, Jeremiás könyvében is alkalmazták.

Mi az angol ábécével fogjuk használni. A szöveget előtte átalakítjuk a következőképpen:

* csupa nagybetűsre alakítjuk,
* az ékezetes betűket nem ékezetessel helyettesítjük,
* kihagyjuk a szóközöket, írásjeleket.

Így nehezebb megfejteni a szöveget. Az angol ábécé betűi (kétféle számozással):

01234567890123456789012345  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
54321098765432109876543210

Kódoljuk először az ABLAK szót: A🡪Z, B🡪Y, L🡪O, A🡪Z, K🡪P, tehát a kódolt alak ZYOZP.

Kódold a POGANY szót! Írd ide a kódolt alakot:

## Szöveg átalakítása

Kézzel elég nehézkes a kódolás, ezért írunk hozzá programot. Ez a billentyűzetről fogja beolvasni a nyílt (kódolandó) szöveget.

Először csak azt a változatot készítjük el, amely beolvassa és átalakítja a szöveget a fenti három pont szerint, de a kódolást még nem végzi el.

A programban szükségünk lesz egy állandóra, amely az angol ABC betűit tartalmazza. Mivel ezt két metódusban is fogjuk használni, az osztályon belül, de a metódusokon kívül (az elején) fogjuk deklarálni. Így az osztályon belül minden metódusban elérhető lesz.

A szöveg átalakítását az atalakit() metódus fogja végezni, amely paraméterként megkapja a beolvasott és nagybetűsre alakított stringet.

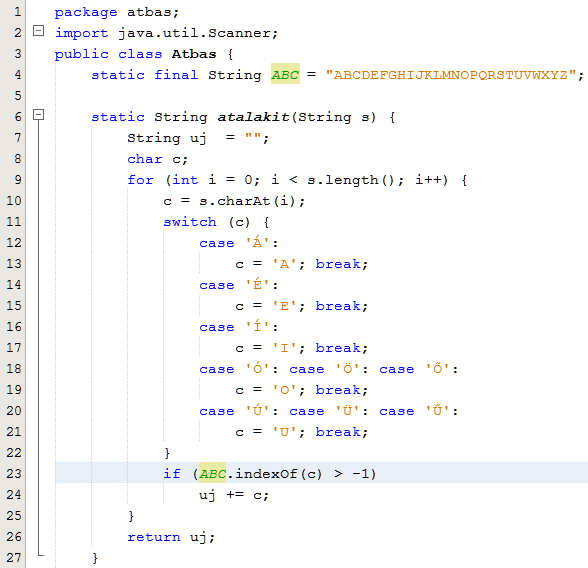
Az átalakított stringet az uj nevű változóban állítjuk össze. Ez a metódus elején üres lesz, és egyenként adjuk hozzá a karaktereket.

A program végigmegy a kapott s string összes karakterén, és a következőket teszi:

1. Az ékezetes betűket lecseréli ékezet nélkülire.
2. Ha a karakter szerepel az ABC-ben, hozzáadja az uj változóhoz.

Végül a metódus visszaadja az uj stringet.

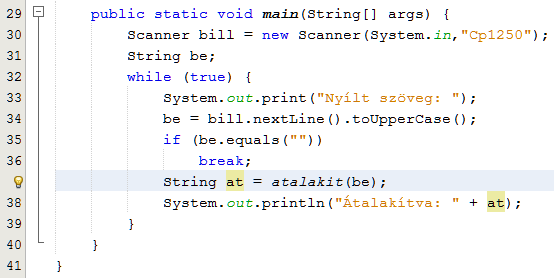
Nézzük a program elejét az atalakit() metódussal:



Az elágazást egy switch utasítással oldjuk meg. Figyeld meg, hogy több ághoz is tartozhatnak ugyanazok az utasítások! Például az Ó, az Ö és az Ő karaktereket is O-ra kell cserélni. A case-eket inkább egymás alá szokták írni, de így rövidebb, és így is működik.

Jöjjön a main() metódus! Ebben a szokott módon olvassuk be a szövegeket addig, amíg üres szöveget nem írnak be.

Ezután meghívjuk az atalakit() metódust, majd kiírjuk az eredményt.



Készítsd el és próbáld ki a programot!

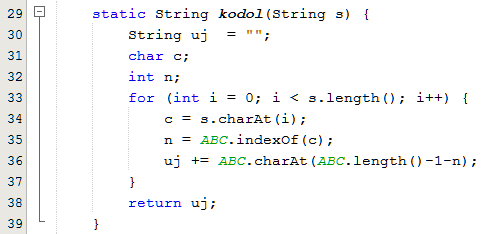
## Szöveg kódolása

A szöveg kódolásához azt kell végiggondolnunk, hogy ha egy karakter az ABC n-dik karaktere, akkor hogyan kapjuk meg a párját, azaz az ABC végétől visszafelé az n-dik karaktert.

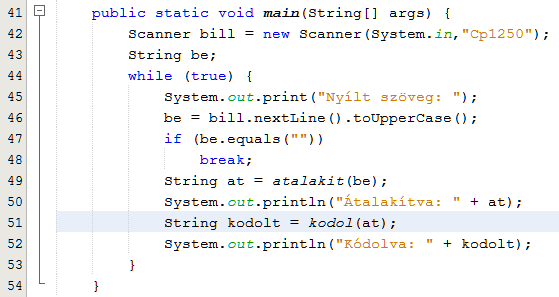
01234567890123456789012345  
ABCDEFGHIJKLMNOPQRSTUVWXYZ  
54321098765432109876543210

A megoldás egyszerű: az utolsó karakter sorszámából kivonjuk az n-et. Például a D karakter a 3-as indexű helyen van. Az utolsó karakter a 25-ös indexű. Akkor n párja a 25-3=22-es karakter, vagyis a W. Persze a programban a 25 helyett az ABC hossza-1 fog szerepelni.

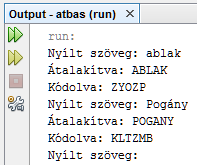
Nézzük a kodol() metódust! Ezt az atalakit() metódus után írd be, a program a 28-as üres sorig változatlan. A lényeg a 36-os sorban van.



Természetesen a kodol() metódust is meg kell hívni a main() metódusból. A program végét így módosítsd (51-52. sor):



Készítsd el és próbáld ki a programot! Ellenőrizd a lecke elején kézzel végzett kódolásokat!



## Kérdés

Hogyan lehet visszaalakítani a kódolt szövegeket? Teljesen meg fognak egyezni az eredeti nyílt szöveggel?

Írd ide: