# 27. Stringek 2.

## Keresés a szövegben

Gyakori feladat, hogy azt kell megvizsgálni, hogy egy karakter vagy egy szövegrészlet hol szerepel egy stringben. Erre való az indexOf() metódus, amely azt adja meg, hogy a stringben a keresett karakter vagy string melyik indexű helyen fordul elő.

Például:

String s = "Sziget Fesztivál";  
System.out.println(s.indexOf('F')); // 7  
System.out.println(s.indexOf("ige")); // 2  
System.out.println(s.indexOf('f')); // -1

A nagy F karakter a 7-es indexű helyen van. Az "ige" String a 2-es indexű helytől kezdve található.

A kis f karakter nem szerepel a Stringben. Ebben az esetben -1 az eredmény. Ennek segítségével vizsgálhatjuk, hogy szerepel-e valami a szövegben vagy nem. Ha az eredmény -1-nél nagyobb, akkor igen, egyébként nem.

Megfigyelhetjük azt is, hogy az indexOf() metódus különbséget tesz a kis- és a nagybetűk között.

Ha egy szövegrész többször is előfordul, akkor az elsőt találja meg.

Mi lesz az alábbi kifejezés értéke? Írd ide: 8

s.indexOf("eszti");

## Kis- és nagybetűre alakítás

Szintén gyakori feladat a szöveg kis- vagy nagybetűsre alakítása. A toLowerCase() metódus csupa kisbetűsre, a toUpperCase() metódus pedig csupa nagybetűsre alakítja a stringet.

Például:

String kicsi = s.toLowerCase(); // sziget fesztivál  
String nagy = s.toUpperCase(); // SZIGET FESZTIVÁL

Fontos! A stringekre alkalmazott metódusok nem az eredeti stringet (s) módosítják, hanem egy újat hoznak létre! Ha meg akarod őrizni az eredményt, tárolnod kell egy változóban!

## Tuvudsz ívígy beveszévélnivi?

Készítsünk programot, amely a beírt szöveget nagybetűsre alakítja, majd „titkos nyelvre” fordítja úgy, hogy minden magánhangzó után beszúr egy V-t, és megismétli a magánhangzót. Üres szöveg megadásával lehessen kilépni.

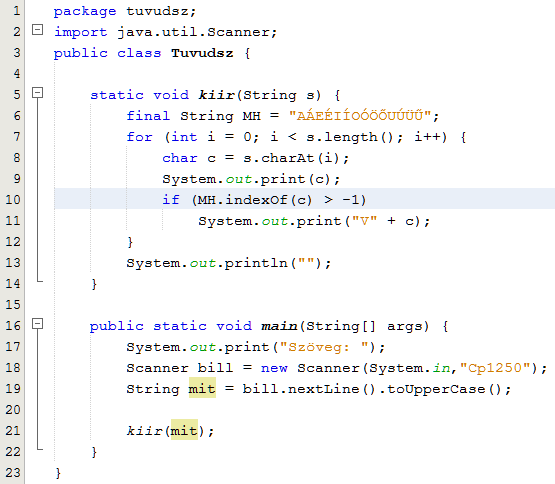
Példa:

Tudsz így beszélni? 🡪 TUVUDSZ ÍVÍGY BEVESZÉVÉLNIVI?

Mi lesz a következő szövegből: Pogány Frigyes

Írd ide:

Nézzük a programot!



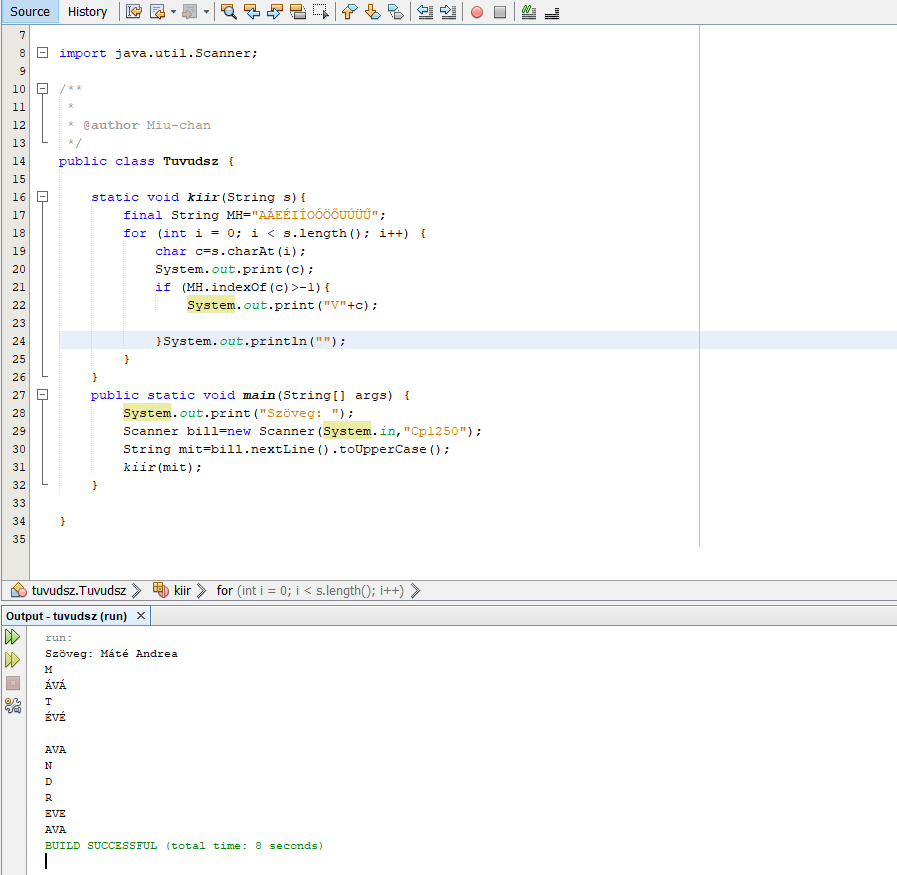
A kiírást egy külön metódussal (kiir) oldjuk meg.

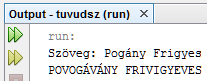
A 6. sorban az összes (nagybetűs) magánhangzót elhelyezzük egy String típusú állandóban (MH). (Az MH csak ebben a metódusban használható.)

Végigmegyünk a beolvasott szöveg karakterein, kiírjuk őket, majd megvizsgáljuk, hogy szerepelnek-e a magánhangzók között (10. sor). Ha igen, akkor kiírunk egy V betűt, majd újra a magánhangzót.

A metódus végén sort emelünk.

A main() metódusban a beolvasott szöveget nagybetűsre alakítjuk (19. sor), és ezt adjuk át a kiir() metódusnak.

Készítsd el és próbáld ki a programot!  



## Stringek egyenlősége

Ciklusok és elágazások feltételeiben gyakran vizsgáljuk, hogy egyenlő-e két string. Ezt azonban nem vizsgálhatjuk az == jelekkel!

Ez azért van, mert String típusú változóknál (és minden más objektumnál) a változó csak az objektum címét tartalmazza. Így az egyenlőségjelek csak azt vizsgálják, hogy a két string címe azonos-e. Ez egyáltalán nem biztos akkor sem, ha ugyanazt a szöveget tartalmazzák.

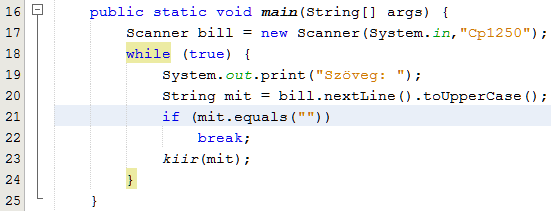
Tulajdonképpen végig kellene nézni a stringek összes karakterét, hogy megegyeznek-e. Szerencsére ezt nem kell megírnunk, mert minden stringnek van egy equals() metódusa, amely megvizsgálja, hogy a paraméterként kapott string megegyezik-e azzal, amelyikre meghívjuk.

Például:

String a = "Olimpia";  
String b = "Olimpia";  
if (a.equals(b)) Sytem.out.println("Megegyeznek");

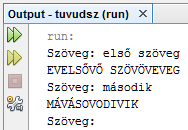
Módosítsuk *tuvudsz* programunkat úgy, hogy ne csak egy szöveget alakítson át, hanem többet egymás után! Abból tudjuk, hogy a felhasználó már nem akar több szöveggel dolgozni, hogy nem ír be semmit, csak megnyomja az entert. (Vagyis üres szöveget ír be.)

Csak a main() metódust kell megváltoztatnunk. Kell bele egy végtelen ciklus, amely bekéri a következő szöveget, és ha nem üres, akkor meghívja a kiir() metódust, egyébként pedig megszakítja a ciklust:



Figyeld meg a 21. sorban a mit változó vizsgálatát! Az üres szöveget az idézőjelek jelentik. Azt vizsgáljuk, hogy a mit változó egyenlő-e ezzel.

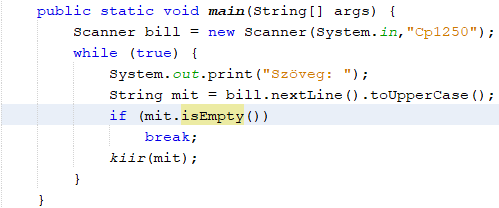
Készítsd el és próbáld ki a programot!



## Kiegészítés

Azt, hogy egy string üres-e, az isEmpty metódussal is vizsgálhatjuk. Ennek értéke akkor igaz, ha a string üres.

Próbáld ki így is a programot:



## Kérdés

Eddigi ismereteink alapján egy harmadik módon is meg lehet vizsgálni, hogy üres-e egy string. Hogyan? mit.length()==0