# 26. Stringek 1.

## Karakterláncok használata

A karakterlácok vagy stringek karakterek sorozatából állnak. Néhány dolgot már tanultunk róluk:

* típusuk a String osztály (nagybetűvel!),
* értékeiket "idézőjelek között" adjuk meg,
* összefűzhetők a + jellel.

Anak ellenére, hogy a String osztály, egy új String típusú változó létrehozásához nem kell new utasítás (mint például a Scanner vagy a Random osztályoknál).

Például:

String s = "Pogány Frigyes";  
String isk = s + "Szakgimnázium";

Ha az egyes karaktereket külön szeretnénk kezelni, akkor a String osztály charAt() metódusát kell alkalmaznunk. Ennek a metódusnak paraméterként meg kell adnunk, hogy hányadik karakterről van szó. A karakterek számozása 0-val kezdődik, és a következő ábra szemlélteti:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| P | o | g | á | n | y |
| 0 | 1 | 2 | 3 | 4 | 5 |

Tehát az első karakter: s.charAt(0)  
a második karakter: s.charAt(1)  
...  
az utolsó (6.) karakter: s.charAt(5)

A karakter sorszámát **indexnek** szoktuk hívni.

Ha az utolsó karakterre van szükségünk, és nem tudjuk előre a szöveg hosszát, akkor lekérdezhetjük a length() metódussal:

int hossz = s.length();

Ennek segítségével egy szöveg utolsó karakterét így adhatjuk meg a hossz ismerete nélkül:

char c = s.charAt(s.length()-1);

Ha egyesével szeretnénk végigmenni egy string összes karakterén, és minddel valamilyen műveletet végezni, akkor for ciklust kell használnunk. Például az alábbi kódrészlet egyenként kiírja az s string karaktereit, mindet külön sorba:

for (int i = 0; i < s.length(); i++)  
 System.out.println(s.charAt(i));

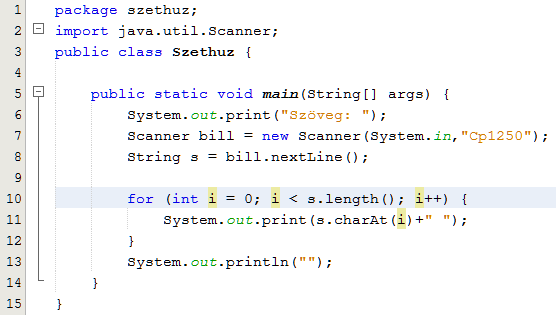
Az i ciklusváltozó az indexeken megy végig.

Fontos! Egy String típusú változó egyes karaktereit külön-külön nem tudjuk módosítani!

## Szöveg széthúzása

Készítsünk az előzőek felhasználásával egy olyan programot, amely bekér egy szöveget, majd kiírja úgy, hogy minden karakter után tesz egy szóközt!

Nézzük a programot:

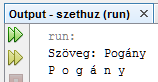


A program első felében beolvassuk az átalakítandó szöveget. Mivel ékezetes betű is lehet benne, megadjuk a 1250-es Windows kódlapot is a Scanner objektum létrehozásakor.

Utána egy for ciklussal végigmegyünk az indexeken, és mindegyiknél kiíratjuk az adott indexű karaktert, majd egy szóközt is.

A program végén még új sort kezdünk.

Készítsd el és próbáld ki a programot!



## Szöveg egy részének kivágása

Ha nem csak egy karakterre, hanem több egymás utáni karakterre van szükségünk egy stringből, akkor a substring() metódust használhatjuk.

Például:

String s = "Ez egy mondat.";  
String vege = s.substring(7); // "mondat."  
String kozepe = s.substring(3,6); // "egy"

A vege változóba a 7-es indexű karakterétől a végéig lévőket teszi.

A kozepe változóba a 3-as indexűtől az 5(!)-ös indexűig kerülnek a karakterek. A második paraméter azt adja meg, hogy melyik karakter **előtt** kell megállni!

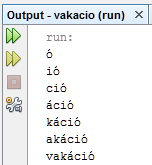
Vigyázat! Ha olyan indexet adunk meg, amely nagyobb, mint a string hossza, akkor hibaüzenettel leáll a program!

Például:  
String kozepe = s.substring(10,15); // hibás!

## Vakáció

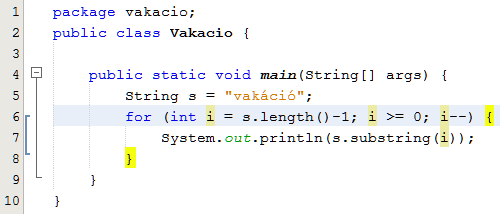


Készítsünk programot, amely a vakáció szót írja ki az alábbi minta szerint!



Először csak az utolsó karaktert kell kiírni, azután az utolsó kettőt, és így tovább. Ezért az i ciklusváltozó először a string hossza-1 lesz, utána pedig egyesével csökken nulláig, és mindig az i-dik karaktertől kezdve kell kiíratni a szöveget.

A megoldás:



Készítsd el és próbáld ki!

## Feladat

Mit ír ki az alábbi program? Írd ide:

String s = "Európa Bajnokság";  
System.out.println(s.substring(7,10));