# 23. Osztályok 4.

## Kéktúra

Az Országos Kéktúra Magyarország északi részén végighaladó folyamatos, jelzett turistaút. A Kéktúrának a Balaton-felvidéken is több, rövidebb idő alatt bejárható túrája van. Egy ilyen túra adatait kell feldolgoznia ebben a feladatban.

A *kektura*.*csv* állomány első sorában a túra kezdetének tengerszint feletti magassága található méterben megadva. A további sorok a túra egy-egy szakaszához tartozó adatokat tárolják. Az adatokat pontosvessző választja el egymástól. Például:

Hidegkuti major;Leteres a Sztupahoz;3,903;153;53;n

A sorokban lévő adatok rendre a következők:

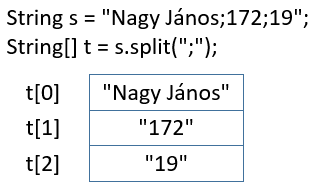
* A túraszakasz kiindulópontjának a neve.   
  Például: „Hidegkuti major”
* A túraszakasz végpontjának a neve.   
  Például: „Leteres a Sztupahoz”
* A túraszakasz hossza [km]. Például: „3,903”
* A túraszakaszon mért emelkedések összege [m].   
  Például: „153”
* A túraszakaszon mért lejtések összege [m]: Például: „53”
* A szakasz végpontja pecsételőhely-e: „i” vagy „n”

Nézd meg a *kektura.csv* fájl tartalmát!

## String szétvágása

A fájlban egy soron belül sok adat van, pontosvesszővel elválasztva. Ezt úgy lehet könnyen feldolgozni, hogy a sorokat szétvágjuk a pontosvesszőknél, és a részeket elhelyezzük egy string tömbben.

Erre szolgál a stringek split() metódusa. Paraméterként azt a karaktert vagy azokat a karaktereket kell megadni, amelyeknél a vágásokat el kell végezni. Például:



Arra vigyázz, hogy a tömb elemei stringek, akkor is, ha számjegyeket tartalmaznak!

## Osztályok létrehozása

Először meg kell terveznünk, hogy milyen adatszerkezetben tároljuk az adatokat.

Minden túraszakasz adatait egy-egy objektumban fogjuk tárolni, és ezekből hozunk majd létre egy listát. Az objektum típusa Szakasz lesz. A Szakasz osztály hat adattagot fog tartalmazni a fenti felsorolásnak megfelelően.

A Szakasz osztályt külön fájlba tesszük:

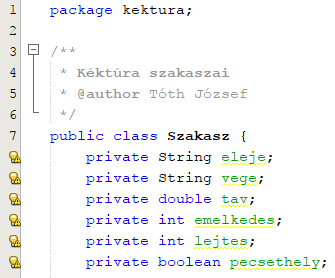
1. Először kezdj egy új projektet *kektura* néven, a szokott módon!
2. A Projects ablakban kattints a jobb gombbal a *kektura* csomagra, majd a menüből választd a New, Java Class... parancsot!
3. Az osztály neve legyen Szakasz (nagy S-sel!), majd kattints a Finish gombra!

Így most már két osztályból áll a programunk:

* A Szakasz osztályban definiáljuk az egyes szakaszok kezeléséhez szükséges adattagokat és metódusokat.
* A Kektura osztályban van a main() metódus, a program indításakor az fog elindulni.

## A Szakasz osztály

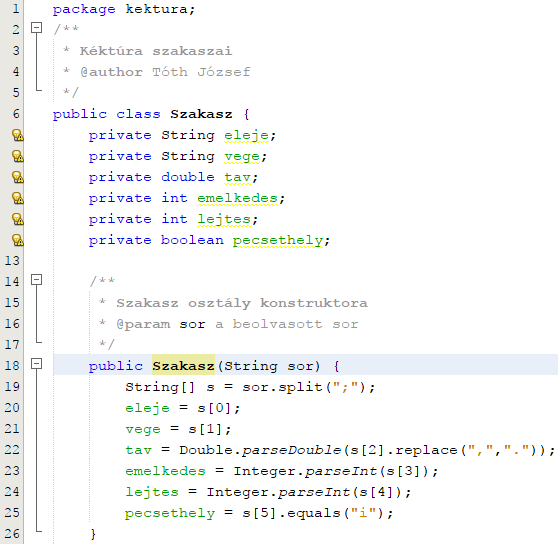
A Szakasz osztályban először megadjuk az adattagokat. Írd be a következőket: (A szerző nevéhez írd a saját nevedet!)



Mivel a Szakasz osztály példányait használjuk majd, itt nem kell a static szó sem az adattagok, sem a metódusok elé.

A privát adattagokat publikus metódusokkal fogjuk kezelni.

Egészítsd ki a Szakasz osztályt egy olyan konstruktorral, amely egy stringet kap, és ebből beállítja az adattagok kezdőértékeit:

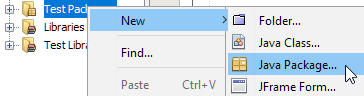
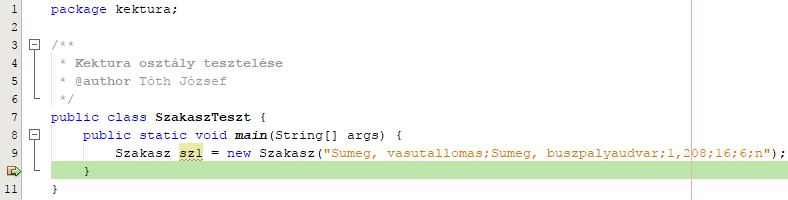


Figyeld meg a split() metódus használatát és a típusátalakításokat!

A 22. sorban azért helyettesítettük a vesszőket pontokkal, mert a parseDouble() metódus csak tizedespontot fogad el (tizedesvesszőt nem).

## Tesztelés

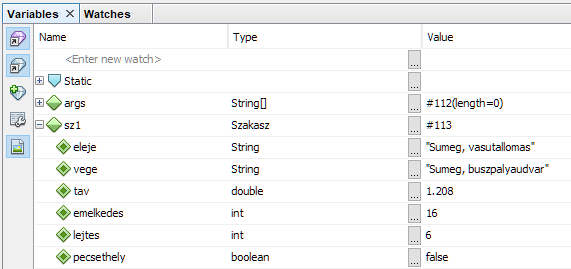
Készíts egy külön teszt osztályt a Szakasz osztály teszteléséhez:

1. Futtasd le egyszer a programot (F6)! Egyelőre nem fog semmit csinálni, de a Project ablakban megjelenik egy Test Packages nevű mappa. Ebben szokták elhelyezni a teszt osztályokat.
2. Kattints a jobb gombbal a Test Packages mappára, és hozz létre egy csomagot!  
   
3. A csomag neve legyen ugyanaz, mint a programé! (kektura)
4. A csomagban hozz létre egy új osztályt (New Java Class) SzakaszTeszt néven!
5. Az új osztályban készíts egy main metódust a következőképpen:  
   

A paraméter szövegét átmásolhatod a *kektura.txt* fájl első sorából.

Helyezz el egy töréspontot a 10. sorban, majd indítsd el a debuggert (Ctrl+Shift+F5, vagy jobb kattintás a SzakaszTeszt osztályra és Debug File)!

Ellenőrizd az sz1 objektum adattagjait, majd kapcsold ki a töréspontot!

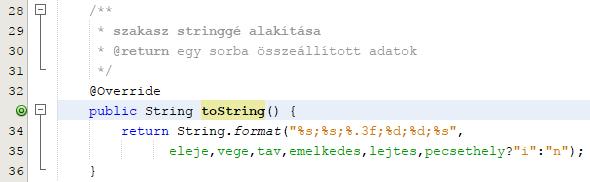


## Objektum kiíratása

Minden objektumnak van egy toString() metódusa. Ez azonban csak az objektum nevét és egy címet ad meg, így nem sok mindenre használható.

Mivel a programban több helyen is ki kell majd íratnunk egy-egy szakasz adatait, célszerű ezt az (Object osztálytól) örökölt metódust felülírni egy használhatóval.

Egészítsd ki a Szakasz osztályt a toString() metódussal:

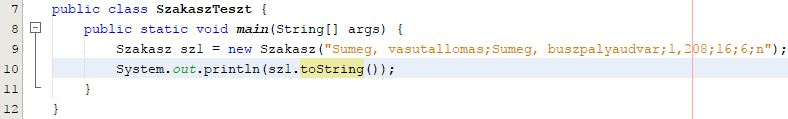


Az @override annotáció jelzi, hogy ez egy örökölt metódus átírása.

És van a 35. sorban még egy újdonság: az egysoros if utasítás.

Ennek a kifejezésnek:   
pecsethely?"i":"n"  
az értéke a pecsethely változó értékétől függ. Ha a pecsethely értéke igaz, akkor az eredmény "i", egyébként pedig "n".

Egészítsd ki a tesztet az alábbiak szerint, majd próbáld ki:



A következő leckében folytatjuk.

## Kérdések

1. Ha az x változó értéke 8, akkor mi lesz az s változó értéke?  
   s = (x % 2 == 0 ? "páros" : "páratlan")
2. Írd le ugyanezt if utasítással:
3. Ha m egy több szóból álló string, a szavak között egy-egy szóközzel, akkor mit ad meg az alábbi kifejezés?  
   m.split(" ").length