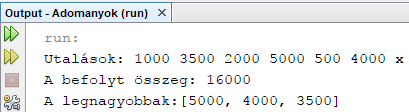
# 13. Listák 2.

## Adományok

A mai feladatban beolvasunk egy jótékony célra érkező felajánlásokat, kiszámítjuk az összegüket, majd kiíratjuk a három legnagyobbat:



Eközben gyakoroljuk a listák használatát.



## Lista elemeinek beolvasása

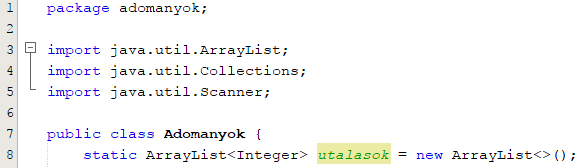
Először be kell olvasnunk a billentyűzetről a lista elemeit.

Nem tudjuk előre, hogy hány számot fognak beírni, ezért egy egész számokból álló listát fogunk alkalmazni.

Mivel a lista elemei csak objektumok lehetnek, ezért az egész számokat be kell csomagolni az Integer osztály példányaiba.



Kezdj egy új projektet *adomanyok* néven! Hozz létre az Adomanyok osztályban egy utalasok nevű, Integer típusú elemekből álló listát!

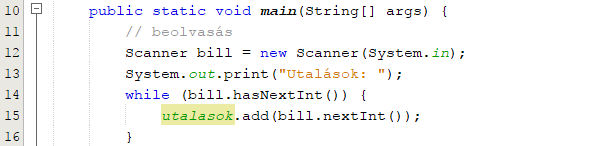


A program csak egyszer kérdezi meg az utalásokat, de a kérdés után akárhány utalást beírhatunk szóközökkel, tabulátorokkal és sorvégjelekkel (Enterekkel) elválasztva. A számsorozat végét egy olyan karakterrel jelezzük, amely nem értelmezhető számként vagy elválasztójelként. A fenti példában ez az x.

A beolvasáshoz felhasználjuk a Scanner osztály hasNextInt() metódusát, amely jelzi, hogy a bemeneten a következő karakterek értelmezhetők-e egész számként vagy elválasztójelként. Az Enter megnyomása után vár a további karakterekre.

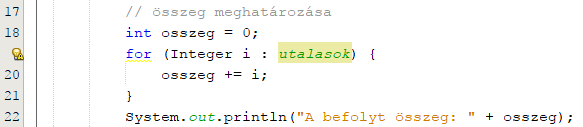
Ahogyan az előző leckében tanultuk, a listához az add() metódussal lehet újabb elemet hozzáfűzni.

A beolvasást így valósítjuk meg:



## Az összeg meghatározása

Az utalások összegének meghatározását egy egyszerűsített for ciklussal végezzük el, amely a lista összes elemét hozzáadja az osszeg változóhoz:



Írd be, majd próbáld ki a programot!

## Listák rendezése

A listák rendezéséhez a Collections osztály sort() metódusa használható. A fordított sorrendet a Collections.reverseOrder() metódussal adhatjuk meg (mint a tömböknél). Ezért importáltuk a program elején a Collections osztályt.

Rendezzük csökkenő sorrendbe az utalasok listát:



## Listák másolása, részlisták

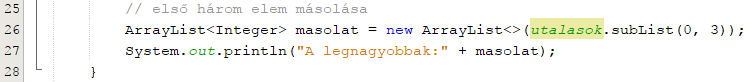
A listák nem másolhatók úgy, hogy egy változót egyenlővé teszünk a lista nevével. Ha például ezt írnánk:   
masolat = utalasok;  
akkor mindkét változó ugyanarra a listára mutatna.

Ehelyett létre kell hozni egy új listát, és az inicializálásnál a zárójelek között meg kell adni az eredeti listát:  
masolat = new ArrayList<>(utalasok);

Most azonban nem a teljes listát szeretnénk egy új listába másolni, csak az első három elemét. Ilyen részlistát a subList() metódussal tudunk előállítani. Például:  
utalasok.subList(0, 3)

Ez a kifejezés egy olyan részlistát állít elő, amely az utalasok lista 0. elemével kezdődik, és a 3. elem már nincs benne. Vagyis a 0., az 1. és a 2. elem lesz benne. (Ez az első három elem.)

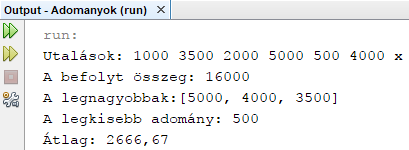
Ennek alapján így tudjuk kimásolni az utalások lista első három elemét a másolat listába:



Utána ki is íratjuk őket. Próbáld ki!

## Önálló feladatok

1. Egészítsd ki a programot úgy, hogy írja ki a legkisebb adományt is!
2. Egészítsd ki a programot még azzal is, hogy ha 0-nál több adomány érkezett, akkor írja ki az átlagukat is, két tizedesjegyre kerekítve!



## Segítség a megoldáshoz

Az alábbi táblázatban kiegészítettük a múltkori összefoglalót a legfontosabb listákkal kapcsolatos műveletekről:

|  |  |
| --- | --- |
| **Példa** | **Leírás** |
| lista.add("Mosogatás") | Hozzáadja a Mosogatás elemet a lista végéhez. |
| lista.add(1,"Vasalás") | Beszúrja az 1-es indexű elem elé a Vasalás elemet. |
| lista.set(1,"Főzés") | Átírja az 1-es indexű elemet Főzésre. |
| lista.get(1) | Megadja a lista 1-es indexű elemét. |
| lista.remove(2); | Törli a 2-es indexű elemet. |
| lista.size() | Megadja a lista elemeinek számát. |
| lista.subList(1,4) | Részlista az 1-től a 3-as indexű elemig (a 4-es indexű már nincs benne!) |