# 16. Fájlok 3.

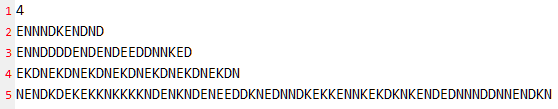
## Robot



Egy barátod robotot épített. A programozásban azonban már nem annyira jártas, ezért a segítségedet kéri.

A robot mozgását egysoros programok vezérlik. A sorokban négyféle karakter lehet: E, K, D, N. Ezek a négy égtáj (észak, kelet, dél és nyugat) irányába tett egységnyi lépéseket jelentik. Például a KKE program hatására két lépést megy keletre, majd egyet északra.

A programokat egy szöveges fájlban tárolják (*utak.txt*, UTF-8 kódolás). A fájl első sora a programok számát tartalmazza, a többi sor pedig egy-egy programot (a sorok végén sortöréssel).



## Beolvasás és kiírás

A program első változatának csak be kell olvasnia az adatokat az *utak.txt* fájlból, majd az első sor kivételével ki kell írnia egy *kimenet.txt* fájlba.

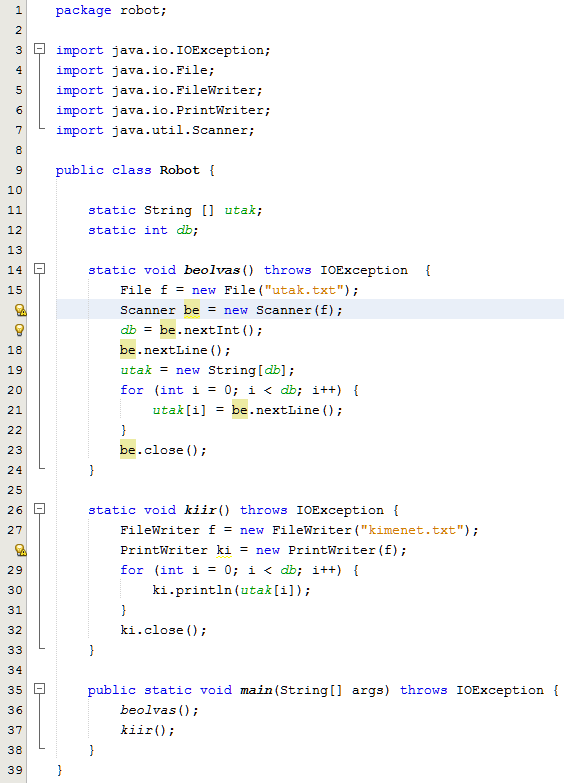
A beolvasást a beolvas() metódus, a kiírást pedig a kiir() metódus fogja végezni.

A beolvasott adatokat most a memóriában kell tárolni. Mivel itt rögtön a fájl elején beolvassuk az elemek számát, az utak beolvasása előtt létre tudjuk hozni a megfelelő méretű tömböt.

A beolvasott utakat az utak nevű, String típusú tömbben tároljuk, az utak számát pedig a db nevű egész típusú változóban. Mindkettőt használjuk majd több metódusban is, ezért a metódusok előtt, a Robot osztályon belül globális változóként deklaráljuk őket.

A hibakezelésnél az IOException kivételt adjuk meg, amely minden olvasási és írási hibához jó. A main() metódus meghívja a beolvas() és a kiir() metódusokat, és nem kezeli a kivételeket, azért a main() metódusnál is meg kell adni a kivételt.

Készítsd el a programot:



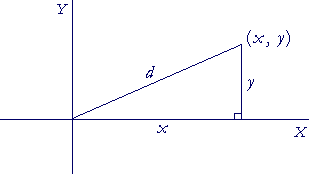
A 18. sorban a szám beolvasása után beolvassuk a sor vége jelet is, hogy az utak beolvasásánál ne okozzon hibát.

A kipróbálás előtt másold az *utak.txt* fájlt a projekt mappájába, utána pedig ellenőrizd a *kimenet.txt* fájlban az eredményt!

## Távolság

A következő változatban minden útnál kiszámítjuk, hogy az origóból indulva milyen távol került az út végére.

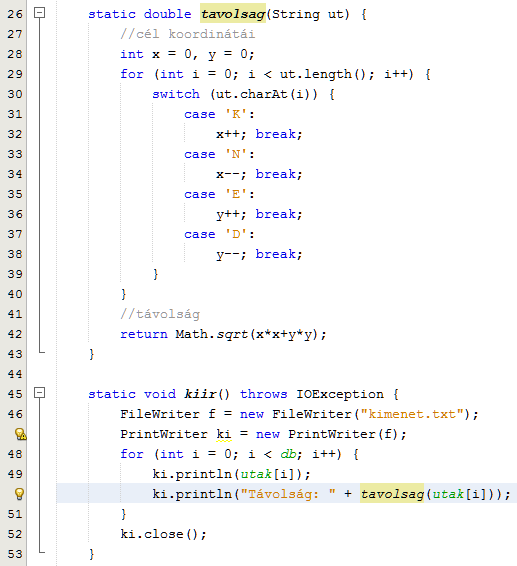
Ehhez először az utat végigjárva meghatározzuk az út végpontjának koordinátáit. Utána Pithagorász-tétel segítségével kapjuk meg a távolságot.



The Pythagorean theorem

Először oldd meg papíron az első úttal!  
Az eredményeket írdd ide: x = , y = , d =

Szúrd be a tavolsag() metódust a kiírás elé, majd egészítsd ki a kiir() metódust a távolság kiírásával (50.sor)! A program többi része nem változik.



Futtasd a programot, majd ellenőrizd a *kimenet.txt* fájlt!

## Energia

A robot a mozgásához szükséges energiát egy beépített akkuból nyeri. A robot 1 lépés megtételéhez 1 egység, az irányváltásokhoz és az induláshoz 2 egység energiát használ. Ennek alapján az EKK utasítássor végrehajtásához 7 egység energia szükséges.

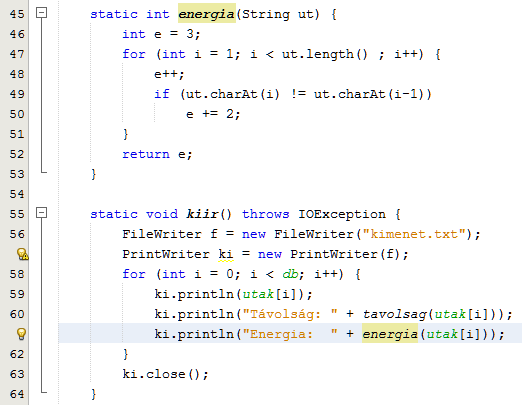
Mennyi energia kell a DND utasítássor végrehajtásához? Írd ide:

Alakítsuk át a programot úgy, hogy az eredeti utak és a távolságok után írja ki a szükséges energia mennyiségét is!

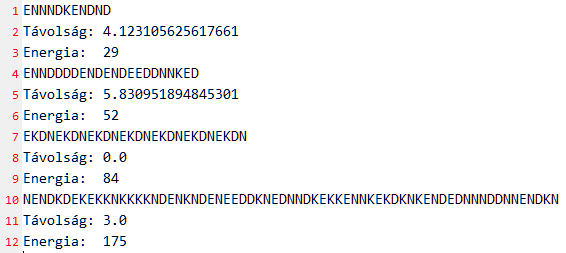
Az első lépéshez 3 egység kell (elindulás + 1 lépés). A további lépésekhez is kell 1 egység, és még 2 egység, ha irányváltás volt. Akkor történt irányváltás, ha a lépés nem egyezik meg az előtte lévővel. (Ezt csak a második, vagyis az 1-es indexű elemtől kezdődően tudjuk vizsgálni!)

Feltételeztük, hogy az út legalább egy lépést tartalmaz.

Szúrd be az energia() metódust a kiírás elé, majd egészítsd ki a kiir() metódust a szükséges energia kiírásával (61.sor)! A program többi része változatlan.



Futtasd a programot, majd ellenőrizd a *kimenet.txt* fájlt!



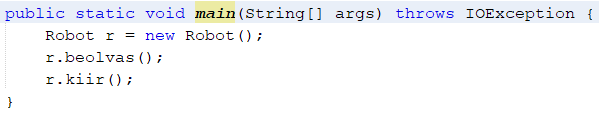
## Kiegészítés

Eddigi programjaink csak egy osztályból állnak, amely a main metódust is tartalmazza. Ilyen esetben az osztály minden változójának és metódusának statikusnak kell lennie.

Van azonban más megoldás is, amelyet gyakran alkalmaznak:

* A main metódusban létrehozunk egy példányt az osztályból. Ebben már nem fog lefutni a main metódus.
* A main-ből a példánynévvel együtt hívjuk meg a többi metódust. (pl. r.beolvas(); )
* Így az osztály változóinak és metódusainak nem kell statikusnak lenniük.

Próbáld ki ezt a Robot osztályban!

1. Alakítsd át a main metódust így:  
   
2. A main metódus kivételével mindenhonnan töröld a static szót!

Futtasd a programot, és ellenőrizd a kimenő fájlt!

## Önálló feladat

Kezeld a beolvas és a kiir metódusokban a keletkező kivételeket, majd távolítsd el a felesleges throws elemeket!