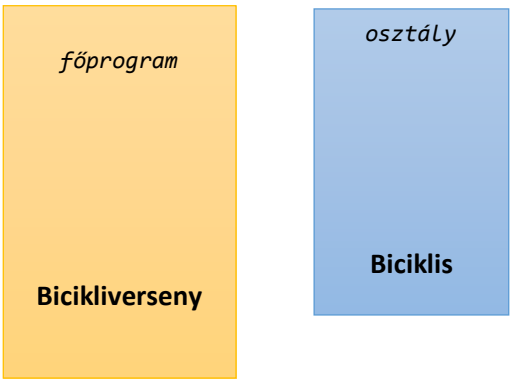
A „Megyénk kőrül gurulunk kőrül” 10 napos teljesítmény túrán Bács-Kiskun megye kőrül kell körbe-körbe kerekezni egy meghatározott útvonalon. Mindennap jegyzik a megtett utat (km).
Ha valaki nem tud, vagy nem szeretne tovább menni, akkor a lapján nincs több bejegyzése.

Az adatokat a `bringa.txt` tabulátorral tagolt fájlban tárolják.

Feladat: a `bringa.txt` fájl beolvasása és az adatok tárolása



Kezdjük a főprogrammal - *mivel nem tudjuk a rekordok számát – **lista*** adatszerkezetben tároljuk a **Biciklis** osztály objektumait – „űrlapok”

```
static ArrayList<Biciklis> biciklisek = new ArrayList<>();
```

a program nem ismeri fel az objektumot, ezért kilépünk a főprogramból és létrehozzuk a **Biciklis** *osztályt*

Biciklis osztály létrehozása:

- mivel nem mindenkinek egyforma hosszú a rekordja („fűrészfog”), ezért itt is lista kell, a listához a `java.util` csomagot használjuk → `import` java.util.*;

Biciklis osztály deklaráció
`- String nev;` változók deklarációja `- ArrayList<Integer> km = new ArrayList<>();`

`+ Biciklis(String sor)` konstruktor deklarációja
`String[] tmp = sor.split("\t");//mindenki a saját adatait szeleteli`
`this.nev = tmp[0];`
`this.fillKmlista(tmp); //naponkénti pontok feltöltése a km listába`

`- eljárás fillKmlista (String[] tmp)` eljárások, függvények
`N = tmp.hossza;`
`ciklus i = 1-től N-ig egyesével`
`km.add( egészszám(tmp[i]) );`
`ciklus vége`
`eljárás vége`

Egyelőre elég, most vissza a főprogramba

főprogram

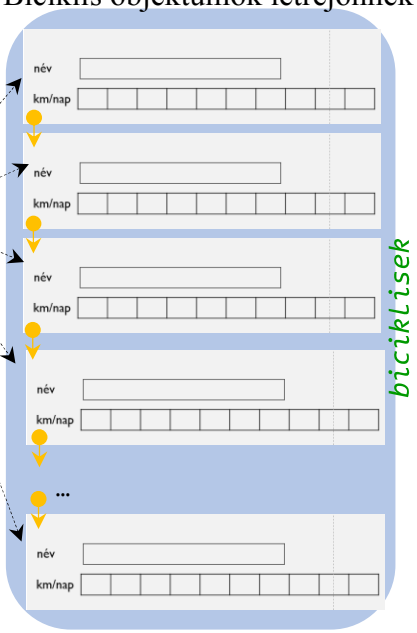
Bicikliverseny

Jöhet az adatok beolvasása:

eljárás **adatokBe**(String *f_neve*)

```
    f = f_neve fájl megnyitása olvasásra;
    sor = f.egy sor beolvasása;
    ciklus amíg (sor != null)
        biciklisek.add(new Biciklis(sor));
        sor = f.egy sor beolvasása;
    ciklus vége
    f.bezárása;
    hiba kezelése
eljárás vége
```

Biciklis objektumok létrejönnek




```
eljárás Teszt()
   N = biciklisek.mérete();
   ciklus i = 0-tól N-ig egyesével
       Ki: biciklisek.get(i).toString();
   ciklus vége
eljárás vége
```

a program nem ismeri fel a toString metódust, ezért kilépünk a főprogramból és a **Biciklis** *osztályban létrehozzuk*

Biciklis osztály: eljárások, függvények

```
@Override
+ függvény String toString()
      s, s1 = "";
      s = String.format("%16s ", nev); // név
      N = km.mérete();
      ciklus i = 0-tól N-ig egyesével
          s1 = s1 + String.format("%5d", km.get(i)); // a megtett utak összefűzése
      ciklus vége
      return s + s1;
   függvény vége
```

Most már tesztelhető a fájl beolvasás!

Feladat:

Készítsen eljárást, amelyik a hibás adatot kicseréli

Pl.:	Tibold Martin	**80**	76
	Tibold Martin	**111**	76

Ahhoz, hogy ez sikerüljön

1. kiválasztás-tételével kikeressük a nevet a *biciklisek* listából, ezt a **Biciklis** osztály **getNev()** metódusával tehetjük meg
2. a javítást a **Biciklis** osztály **setKm()** metódusával tehetjük meg

Biciklis osztály

```
+ függvény String getNev()
     return nev;
   függvény vége
```

```
+ eljárás setKm(int nap, int ujKm)
       km.set(nap, ujKm);
  eljárás vége
```

Főrogram:

```
eljárás pontJavitas()
   Scanner sc = new Scanner(System.in);
   Ki: "biciklis neve: ";  Be: String név = adatbekérés a konzolról
   Ki: "hányadik nap: ";  Be: egészszám nap = adatbekérés a konzolról; nap--;
   Ki: "új km: ";  Be: egészszám ujKm = adatbekérés a konzolról;
   i = 0;
   ciklus amig ( !biciklisek.get(i).getNev().equals(nev) )
       i++;
   ciklus vége
   biciklisek.get(i).setKm(nap, ujKm);
eljárás vége
```