

## Debuggolás NetBeans-ben

A programozás egyik fontos része a hibakeresés. ☹

Hiába örülünk, hogy jól lefordult a kód, bizony ennek ellenére előfordulhat, hogy hibásan működik. Hogyan lehet megkeresni a hibát?

### a) Fapados módon

Az igazi programozók ezt az ördögtől való megoldásnak tekintik, de ha valakinek nincs jobb ötlete, akkor ez még mindig jobb, mint üres szemekkel bámulni a képernyőt, és várni a sült galambra.

Ez a megoldás az, hogy kiíratjuk a gyanús változók értékét. Szerencsére a NetBeans-ben ez is könnyen megy (grafikus felületen való programozás esetén is mindig elérhető az output ablak). Vagyis ha nincs más ötlete, írassa ki a gyanús értékeket.

### b) Debuggolás

Az igazi megoldás azonban az, ha tisztességesen debuggolunk, azaz nyomon követjük a programot.

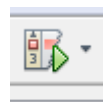
Erre a NetBeans kényelmes lehetőséget biztosít. Képernyőképeken keresztül bemutatok néhány lehetőséget, a többit próbálja meg önállóan felfedezni, vagy nézze át valamelyik tutorialt.

Tegyük fel, hogy ezt a programrészletet szeretnénk nyomon követni:

```
54 private void ujsagIras() {
55     int n = 20;
56     int hanyszor, veletlenCikk, veletlenUjsag;
57     hanyszor = (int) (Math.random()*n);
58     Ujsag ujsag;
59     Cikk cikk;
60     for (int i = 0; i < hanyszor; i++) {
61         veletlenCikk = (int) (Math.random()*cikkek.size());
62         veletlenUjsag = (int) (Math.random()*ujsagok.size());
63         ujsag = ujsagok.get(veletlenUjsag);
64         cikk = cikkek.get(veletlenCikk);
65         ujsag.cikketKozol(cikk);
66     }
67 }
```

Ehhez a kódba úgynevezett breakpoint-okat kell illeszteni. Ezt úgy tehetjük meg, hogy a kiválasztott sor sorszáma kattintunk, vagy ha nincs bekapcsolva a sorszám-mutató (view menüpont), akkor a szürke csíkra. Ekkor a fent látható módon változik a program, a töréspont hatására pedig a program futása (debug módban) megáll majd ennél a pontnál.

Debug módban vagy a Debug menüpontból lehet indítani, vagy ennek a gombnak a hatására.



Ha elindítottuk, akkor a Variables nevű ablakban máris láthatjuk a kódrészletben szereplő változókat, és megnézhetjük az értéküket:

Watches	Variables	Breakpoints	Output	
Name	Type	Value		
+	this	Vezerles	#298	
	n	int	20	
	hanszor	int	2	
	i	int	0	
	veletlenCikk	int	2	
	veletlenUjsag	int	2	
+	ujsgag	Ujsag	#299	
+	cikk	Cikk	#300	

Természetesen a részleteket is láthatjuk, ha akarjuk:

Watches	Variables	Breakpoints	Output	
Name	Type	Value		
+	this	Vezerles	#298	
	n	int	20	
	hanszor	int	2	
	i	int	0	
	veletlenCikk	int	2	
	veletlenUjsag	int	2	
-	ujsgag	Ujsag	#299	
	nev	String	"Sztár világ"	
	datum	String	"2014.09.11"	
	cikkek	ArrayList	"size = 0"	
	osszHazugsag	int	0	
	atlagHazugsagSzazalek	double	0.0	
-	cikk	Cikk	#300	
	szerzo	String	"Hazug Henrik"	
	cim	String	"Fantasztikus horror"	
	karakterSzam	int	2000	
	hazugsagSzazalek	int	80	



A Step Over (F8) gombbal léphetünk tovább. Ekkor lépésről lépésre végig lehet követni, hogy mi történik:

```

private void ujsagIras() {
    int n = 20;
    int hanyszor, veletlenCikk, veletlenUjsag;
    hanyszor = (int) (Math.random()*n);
    Ujsag ujsag;
    Cikk cikk;
    for (int i = 0; i < hanyszor; i++) {
        veletlenCikk = (int) (Math.random()*cikkek.size());
        veletlenUjsag = (int) (Math.random()*ujsagok.size());
        ujsag = ujsagok.get(veletlenUjsag);
        cikk = cikkek.get(veletlenCikk);
        ujsag.cikketKozol(cikk);
    }
}

private void osszesites() {
}

```

Vezerles > ujsagIras > for (int i = 0; i < hanyszor; i++) >

Name	Type	Value
this	Vezerles	#298
n	int	20
hanyszor	int	2
i	int	1
veletlenCikk	int	0
veletlenUjsag	int	3
ujsag	Ujsag	#335
nev	String	"Ki kivel"
datum	String	"2014.09.10"
cikkek	ArrayList	"size = 0"
osszHazugsag	int	0
atlagHazugsagSzazalek	double	0.0



A Step Into (F7) hatására be tudunk lépni a vizsgált metódus belsejébe:

A Step Out (Ctrl F7) hatására pedig vissza.

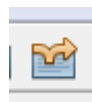


```

35  /**
36  * A paraméterében adott cikket hozzáadja a cikkek listájához.
37  *
38  * @param cikk
39  */
40  public void cikketKozol(Cikk cikk) {
41      cikkek.add(cikk);
42      osszHazugsag += cikk.getHazugsagSzazalekLab();
43      atlagHazugsagSzazalek = osszHazugsag / cikkek.size();
44  }
45
46  @Override
47  public String toString() {

```

Name	Type	Value
this	Ujsag	#335
nev	String	"Ki kivel"
datum	String	"2014.09.10"
cikkek	ArrayList	"size = 0"
osszHazugsag	int	0
atlagHazugsagSzazalek	double	0.0
cikk	Cikk	#346
szerzo	String	"Hazug Henrik"
cim	String	"Döbbnét"
karakterSzam	int	1200
hazugsagSzazalek	int	70



A Step Over Expression (Shift F8) átugorja a vizsgált kifejezést, ezzel pedig le lehet állítani a debuggolást:



Remélem, a képek alapján kedvet kapott arra, hogy ki is próbálja.

Egy videó tutoriál:

<https://www.youtube.com/watch?v=ReLlcbi1es4>

Nyilván sok másik is létezik, de az igazi az, ha saját maga is próbálkozik vele.