

Szakköri segédanyag

Írta: Bátfai Norbert

2003. október 26.

Jávácska – Internet
csak gyerekeknek: avagy hogyan
láttam tizenegy évesen a már min-
denütt burjánzó számítógépeket,
a mindent behálózó Internetet,
a mindig magunknál hor-
dott mobiltelefonokat és a
mindezeket összekötő
programozást?



Általános Nyilvános Engedély

SZAKKÖRI SEGÉDANYAG A FANTASZTIKUS PROGRAMOZÁSHOZ
1.0.0 verzió – „LÁNCOK”

A dokumentumot szakkörvezető tanároknak ajánljuk. A foglalkozás számítógépet igényel, illetve, hogy már telepítve legyen a SUN J2ME Wireless Toolkit (a toolkit a <http://java.sun.com/products/j2mewtoolkit> címről tölthető le). A példákban használt osztályok és képek a Jávácska portál (<http://www.clib.dote.hu/javacska>) Szakkör menü alatt található.

Szerzői jog © 2003 Bátfai Norbert <norbi@javacska.hu>. Minden jog fenntartva.
Jávácska projekt, <http://www.clib.dote.hu/javacska>

Ez a dokumentum szabad szoftver; továbbadhatja és/vagy módosíthatja a GNU General Public Licence feltételei szerint, ahogy a Free Software Foundation közli; vagy a Licenc 2. verziója, vagy pedig (saját belátásunk szerint) bármely későbbi verzió.

Ezt a dokumentumot azzal a reménnyel terjesztjük, hogy hasznos lesz, de MINDENFÉLE GARANCIA NÉLKÜL; az ELADHATÓSÁG vagy ADOTT CÉLRAVALÓ ALKALMASSÁG garantálása nélkül. A részletek a General Public Licence-ben található.

1. Az első saját MIDletjeink

1.1. MIDletek: letölthető mobil Java alkalmazások

A MIDlet osztályból származó objektumaink képesek élni a mobiltelefonokban. Pontosabban a MIDletek a mobiltelefonok Java virtuális gépében képesek élni. A Java virtuális gép egy értelmezőprogram (interpreter), bemenete Java bájtkód (ezek a .class fájlok.)

1.2. A javacska könyvtár tartalma

Ebben a leckében négy láncsal dolgozunk. Mind a négy esetben másoljuk be a javacska könyvtárat az éppen létrehozott projekt src könyvtárába.

A javacska könyvtárban a következő osztályokat találjuk:

`Lanc.java`, `Lancolhato.java`, `VaszonGyongy.java`

Ezeket majd felhasználjuk saját programjainkban, de forrásban egyelőre nem kell őket megismerni. Most elég az alábbiakat tudnunk:

- `Lanc.java`, a `Lanc` osztály a `MIDlet` osztály gyermeke. A `Lanc` osztályból származó objektumok is képesek élni a mobilokban. Saját láncunkat úgy készítjük el, hogy kiterjesztjük a `Lanc` osztályt, azaz a saját láncunk a `Lanc` osztály gyermeke lesz.

Használt viselkedési mintái:

- `startLanc(Gyongy gyongy)`, beállítja, hogy melyik gyöngy legyen a lánc első gyöngye.
- `stopLanc(Gyongy gyongy)`, beállítja, hogy melyik gyöngy legyen a lánc utolsó gyöngye.

- `VaszonGyongy.java`, a `VászonGyöngy` osztály a `Canvas` (azaz vászon) osztály gyermeke. Az olyan gyöngyeinket, amikre rajzolni akarunk, a `VászonGyöngy` osztály gyermekeként írjuk meg.

Használt viselkedési mintái:

- `Image kepetBetol(String kepNev)`, visszaad egy `Image` (azaz kép) objektumot.

2. Képek MIDlet

Hozzuk létre a projektet (Project Name: `Kepek`, MIDlet Class Name: `KepekLanc`, MIDlet-Vendor: a tanuló neve).

Létrejött a projekt könyvtára: WTK2.0/apps/Kepek Ezen belül a res könyvtárba kell tennünk a program képerőforrásait. Másoljuk ide az `első.png`, a `második.png` és a `harmadik.png` képeket. Az `src` könyvtárba pedig másoljuk be a `KepekLanc.java` és a `KepGyongy.java` osztályokat.

Futtassuk a projektet (`Build`, `Run`)

2.1. Feladat

Helyettesítsük a képeket saját képekkel! A saját képek méretének megválasztásánál arra is figyelhetünk, hogy adott készülék mekkora grafikus kijelző (vászon) mérettel rendelkezik.

2.1.1. Néhány vászon mérete

- MOTOROLA T720/T720i, 120x146 pixel (szélesség x magasság), színes
- NOKIA 3510i, 96x54 pixel, színes
- NOKIA 3410, 96x54 pixel, fekete-fehér
- NOKIA 6610, 128x96 pixel, színes
- NOKIA N-GAGE, 176x144 pixel, színes
- SIEMENS M50, 101x64 pixel, fekete-fehér

2.2. Feladat

Láncunk most három gyöngyöt tartalmaz. Bővítsük legalább még egy gyönggyel! A `KepekLanc.java` osztályt kell szerkesztenünk:

```
...
    KepGyongy kepGyongy4;
...
    kepGyongy4 = new KepGyongy("/negyedik.png");
...
    kepGyongy3.kovetkezoGyongy(kepGyongy4);
    kepGyongy4.elozoGyongy(kepGyongy3);
...
    stopLanc(kepGyongy4);
...
```

mentsük a fájlt, majd futtassuk a projektet (`Build`, `Run`)

3. Rajzoló MIDlet

Hozzuk létre a projektet (Project Name: Rajzolo, MIDlet Class Name: RajzoloLanc, MIDlet-Vendor: a tanuló neve).

Létrejött a projekt könyvtára: WTK2.0/apps/Rajzolo Az src könyvtárba másoljuk be a RajzoloLanc.java és a RajzoloGyongy.java osztályokat.

Futtassuk a projektet (,)

3.1. Feladat

Rajzoljunk ki köröket, húzzunk vonalakat is! A RajzoloGyongy.java osztályt kell szerkesztenünk:

```
...  
g.drawArc(40, 5, 30, 30, 0, 360);  
g.drawLine(55, 35, 55, 40);  
...
```

4. Pálcika MIDlet

Hozzuk létre a projektet (Project Name: Palcika, MIDlet Class Name: PalcikaLanc, MIDlet-Vendor: a tanuló neve).

Létrejött a projekt könyvtára: WTK2.0/apps/Palcika Az src könyvtárba másoljuk be a PalcikaLanc.java és a PalcikaGyongy.java osztályokat.

Futtassuk a projektet (,)

4.1. Feladat

Rajzoljuk meg saját pálcika-emberünket kockás papíron, majd programozzuk be! A PalcikaGyongy.java osztályt kell szerkesztenünk.

5. Labirintus MIDlet

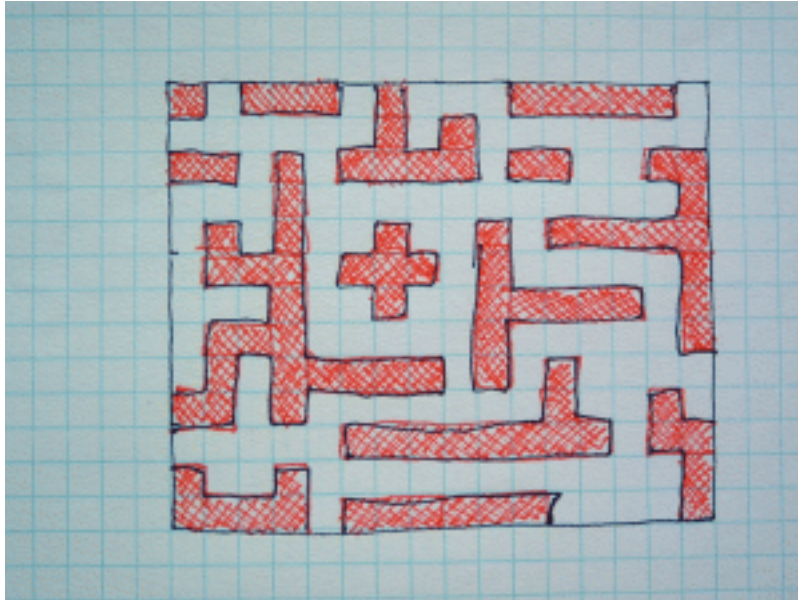
Hozzuk létre a projektet (Project Name: Labirintus, MIDlet Class Name: LabirintusLanc, MIDlet-Vendor: a tanuló neve).

Létrejött a projekt könyvtára: WTK2.0/apps/Labirintus Az src könyvtárba másoljuk be a LabirintusLanc.java és a LabirintusGyongy.java osztályokat. Az res könyvtárba pedig másoljuk be a tegla.png képet.

Futtassuk a projektet (,)

5.1. Feladat

Rajzoljuk meg saját labirintusunkat kockás papíron, majd programozzuk be!



A `LabirintusGyongy.java` osztályt kell szerkeszteni. Ahol a labirintus tervében fal van, az annak megfelelő helyre írjunk egyest, a járat helyére nullást.

Hivatkozások

- [1] Fantasztikus programozás pdf könyv. Jávacska portál, <http://www.clib.dote.hu/javacska> (2003)