

## Ingyenes és nyílt forráskódú Java megoldások az Oracle-tól

Talán még mindig sokan nem tudják, de a Java-fejlesztők már közel két éve ingyenesen használhatják az Oracle JDeveloper 10g fejlesztőrendszert.

■ Az Oracle ezzel a gesztussal a Java-fejlesztők iránti töretlen elkötelezettségét kívánta hangsúlyozni, mikor ezt a lépést megtette. A cég emellett felajánlotta, hogy az Eclipse Foundation nyílt forráskódú fejlesztői közösségben elvállalja az egységes fejlesztőeszközök kialakítását célzó JavaServer Faces (JSF) projekt vezetését, továbbá bejelentette, hogy fő támogatóként csatlakozik az Apache MyFaces kezdeményezéshez.

A nyílt forrású megoldások támogatásának további fokozására az Oracle idén márciusban újabb Java fejlesztőeszközének, az Oracle Toplink-nek a kódját nyitotta meg. Az Oracle Toplink a piacvezető objektum-relációs *perzisztens Java*-architektúra, mely az Oracle Fusion Middleware termékcsalád részeként az objektum-orientált alkalmazások és a relációs adatbázisok közötti szűk keresztmetszetet szünteti meg, így az adatlekérdezések sebessége akár a tízszeresére is növekedhet. Ennek a technológiának a kódjára építve az Eclipse JSF projekt mellett elindította az Eclipse Persistence Platform projektet is. Az Oracle természetesen továbbra is fejleszti saját Oracle JDeveloper eszközét, amelyből elérhető lesz mind a JSF, mind a Toplink. A nyílt forráskódú fejlesztők számára azonban az Eclipse-n keresztül is biztosított lesz az Oracle technológiára épülő magas szintű fejlesztési technológia.

Az alábbiakban nézzük meg, hogy a fent ismertetett Open Source technológiák fejlesztéséhez használható ingyenes Oracle eszköz, a JDeveloper 10g hogyan telepíthető Linux-ra, és mivel

jár egy egyszerű Java alkalmazás fejlesztése a segítségével. Egyszerű példánkban a Java alkalmazás nem fog tartalmazni sem JSF, sem Toplink komponenseket.

### JDeveloper telepítése

1. Mivel maga a JDeveloper is 100% Java-ban készült, ezért teljesen hordozható a különböző operációs rendszerek között. Természetesen ezért első lépésként szükséges a JDK 5.0 Update 6 telepítése, mely letölthető például a <http://www.javasoft.com> oldalról. A JDK saját könyvtárára állítsuk be a JAVA\_HOME környezeti változót.
2. A JDeveloper 10g 10.1.3.2 verziója Linuxra letölthető az OTN (Oracle Technology Network) megfelelő oldaláról. Célszerű a Studio Edition-t választani, az tartalmaz minden funkciót: <http://otn.oracle.com/software/products/jdev/index.html>
3. Csomagoljuk ki a letöltött file-t a jdeveloper könyvtárba.

```
unzip jdevstudiobase10132.zip
➤ -d /jdeveloper
```

4. Ezt követően készítsünk egy mentést a JDeveloper konfigurációs fájlokról, majd adjuk a JAVA\_HOME környezeti változó tartalmát a SetJavaHome értékéhez a konfiguráció fájlban.

```
$ cd /jdeveloper/jdev/bin
$ cp jdev.conf jdev.conf.backup
```

```
$ echo "SetJavaHome $JAVA_HOME"
➤ >> jdev.conf
```

5. Állítsuk be a végrehajtható állományok jogosultságát.

```
$ chmod +x /jdeveloper/
➤ jdev/bin/jdev
$ chmod +x /jdeveloper/
➤ jdev/bin/ojcb
```

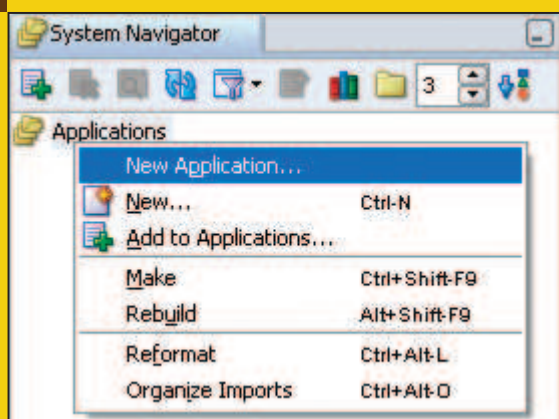
További jogosultság beállítások a [/jdeveloper/jdev/install.html](#) fájlban találhatóak

6. Ezt követően a JDeveloper már futtatható is a jdev parancs kiadásával.

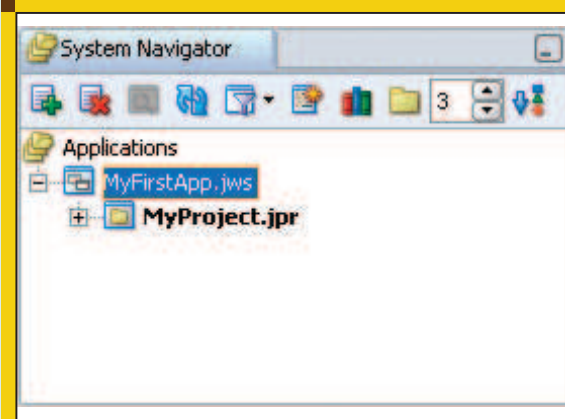
### Alkalmazás fejlesztése

Az alábbi egyszerű példával ellenőrizhetjük hogy megfelelően működik-e fejlesztő környezetünk.

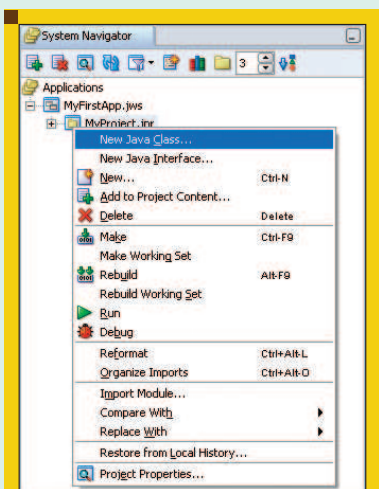
1. A JDeveloper elindítását követően az System Navigator-ban jobb klikk az Applications menüre és ott válasszuk ki a New Application-t egy alkalmazás létrehozása érdekében. Itt hagyhatunk mindent alapértelmezetten, kivéve az alkalmazás nevét módosítsuk MyFirstApp-ra, mint ahogy az automatikusan megjelenő Create Project ablakban a projekt nevét állítsuk MyProject-re.
2. A System Navigator-ban lévő hierarchia második szintjén, azaz az alkalmazás alatt jelennek meg a projektek.



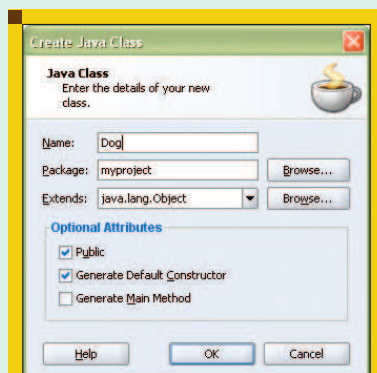
■ 1. ábra Új alkalmazás



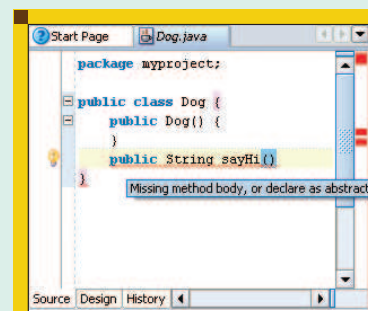
■ 2. ábra A System Navigator



■ 3. ábra Új Java osztály – első lépés



■ 4. ábra Új Java osztály – második lépés



■ 5. ábra A kódszerkesztő

- Hozzuk létre az első java osztályunkat. Jobb klikk a *MyProject*-re és válasszuk a *New Java Class...* opciót.
- A *Create Java Class* párbeszéd ablakban módosítsuk az alapértelmezett *Class1* nevet *Dog*-ra.
- Az új osztály automatikusan megjelenik a kódszerkesztőben, ahol látható az osztály kódjának váza. Adjunk egy új metódust az osztályhoz. A konstruktor után nyomjunk egy *Entert*, és írjuk be az alábbiakat:

```
public String sayHi()
```

Vegyük észre, hogy a szerkesztő ezt a sort pirossal aláhúzza, illetve jobb széles kis piros négyzetek

is jelzik, hogy valami nem stimmel. Továbbá ha az egér kurzorral a sor fölé megyünk, akkor megjelenik egy felirat, mely informál a hibáról.

- Ha a sor végén *Ctrl+Shift+Enter*-t nyomunk, akkor a kódszerkesztő automatikusan kiegészíti a kódot a hiányzó résszel, és a hibajelzések automatikusan eltűnnek, a kódszerkesztő jobb felső sarkában pedig egy zöld négyzet jelzi, hogy minden stimmel.

Az *Oracle JDeveloper*-ben számos a fentihez hasonló kódolást támogató funkció van beépítve. Az ismertett példa sem ér itt véget, a <http://www.oracle.com/technology/obe/obe1013jdev/1013/introide/introjdevide.htm> webcímen megtekinthető a folytatása, melynek legvégén például a Java kód debugolása is ismertetésre kerül.

A kiforrott, teljes funkciókészlettel rendelkező *Oracle JDeveloper* integrált fejlesztői környezet átfogó Java-funkciókat és web-szolgáltatásokat tartalmaz. A terméket korábban felhasználóként 995 dolláros fogyasztói áron vásárolhatták meg a fejlesztők. A szoftver számos egyszerűen kezelhető funkciója kiválóan alkalmas szolgáltatásorientált architektúrájú rendszerek kialakításához, mivel kiterjedten támogatja a webszolgáltatásokat, a *BPEL* alapú üzleti folyamatokat, az *UML*-modellézést és a portleteket.

A cikk elején ismertetett *JSF* és *TopLink* megoldásokról, illetve egyéb *Oracle* fejlesztői technológiákról az alábbi *URL*-n lehet bővebb információt, példákat, dokumentációkat találni:

➔ <http://www.oracle.com/technology/products/developer-tools/index.html>

Sárecz Lajos