

# VII. Appletek, grafika

## 1. Bevezetés

A tantárgy elején említettük, hogy a Java alkalmazásokat két nagy csoportba sorolhatjuk. Ezek:

- **alkalmazások (applications)**
- **„alkalmazáskák” (applets)**

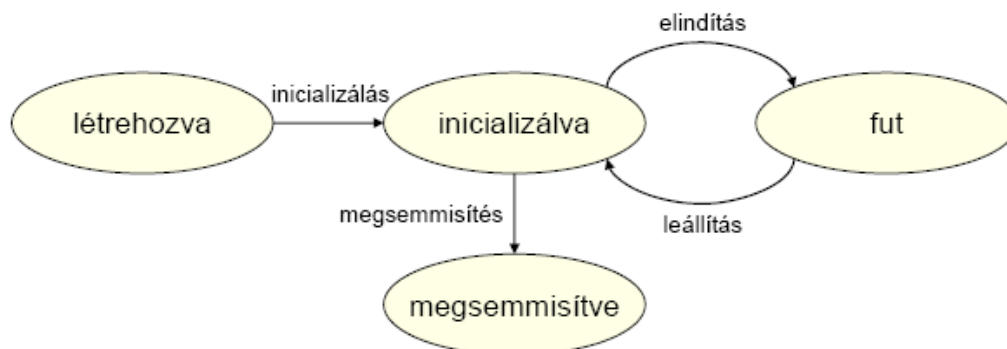
Az eddig megírt programjaink alkalmazások voltak, de a Java elterjedését és hírnevét főként az appleteknek köszönheti. Ezért ebben a fejezetben az Appletekkel fogunk foglalkozni, amelyek szintén java programok, és felhasználható bennük az összes eddig megszerzett ismeret.

Az appletek és az applikációk között az egyik legfontosabb különbség, hogy amíg az alkalmazások önállóan futtatható programok (a JRE segítségével) addig az Appletek futtatásához szükségünk van egy böngésző programra. Ebben az esetben a futtatást a böngészőbe beépülő JVM végzi. A másik futtatási lehetőség, hogy a JDK által biztosított appletviewer programot használjuk.

## 2. Appletek használata

### 2.1 Appletek életrajza

Az appletek életrajza némileg eltér az alkalmazásokétól. Itt a gazda alkalmazás, a böngésző vezérli az appletet. Az appletek állapotait a következő ábra szemlélteti:



Az állapotátmenetek a következők szerint következnek be:

- **Inicializálás:** a böngésző az oldal betöltésekor inicializálja az appletet
- **Elindítás:** az appletet a böngésző az inicializálás után indítja el

- **Leállítás:** amikor az appletet tartalmazó oldalról kilép a felhasználó, a böngésző leállítja az appletet. Ha visszatér, az applet újraindítható (inicializálás nélkül)
- **Megsemmisítés:** ha az appletre már nincs szükség, a böngésző az erőforrások felszabadítására utasítja az appletet. A böngésző a megsemmisítés előtt mindenképpen leállítja az appletet.

## 2.2 Az Applet osztály

Az appletek közös jellemzője, hogy a **java.applet.Applet** osztályból származnak. Az Applet osztály tartalmazza az applet készítéséhez szükséges legfontosabb tulajdonságokat és metódusokat. Az osztály egyes metódusait a böngésző hívja meg, és megfelelnek az életciklus állapotátmeneteinek:

- *init()* : inicializálás
- *start()* : indítás
- *stop()* : leállítás
- *destroy()* : megsemmisítés

Még egy nagyon fontos metódus a **paint()** metódus, amely a rajzolásért felelős. Applet készítés során ezt a metódust használjuk a legtöbbször.

Az applet nem ablak!!! Nincs kerete, címe, ikonjai és nem lehet ablakesemény forrása. Az Applet osztálynak a Panel a közvetlen őse.

## 2.3 Applet készítése

### 1.lépés

Láthattuk, hogy az appletek a java.applet.Applet osztály kiterjesztései, ezért első lépésben importáljuk az applet csomagot:

```
import java.applet.*;
```

### 2.lépés

Ezután definiáljunk egy saját osztályt a fenti Applet osztály kiterjesztésével:

```
public class SajatOsztaly extends Applet
{
}
}
```

(A java megköveteli, hogy a *SajatOsztaly* osztályunk publikus legyen!)

### 3.lépés

Valósítsuk meg a fent említett 4 állapotátmeneti metódusok valamelyikét vagy a **paint()** rajzoló metódust.

## 2.4 Applet elindítása

Ahhoz hogy egy applet el tudjunk indítani először be kell ágyaznunk a bájtkódot egy HTML oldalba. A beágyazáshoz a HTML nyelv alapszintű ismerete szükséges. Ha a böngésző a HTML állományban az `<applet>...</applet>` címkepárt találja, akkor az ott megadott adatok alapján a JVM meghívásával létrehoz egy applet objektumot, majd betölti a böngészőbe és futtatja. A szerkezet a következő:

```
<html>
  <head>
    <title>Példa applet</title>
  </head>
  <body bgcolor=black>

    <applet
      codebase="d:\ "
      code="Pelda.class"
      width="400"
      height="300"
      align=center
      alt="Hiba az appletben. Keresse meg a készítő!"
    >
    <param name="vezeteknev" value="Kiss"/>
    <param name="keresztnev" value="Béla"/>
  </applet>

</body>
</html>
```

### Az <applet> tag paraméterei:

- **codebase** : a java kód relatív könyvtára, ha nem adjuk meg, akkor a HTML-állomány könyvtára
- **code** : a fő osztály specifikációja class kiterjesztéssel
- **width**: az applet szélessége a böngészőben (pixelben megadva)
- **height**: az applet magassága a böngészőben (pixelben megadva)
- **align**: az applet igazítása a böngészőben
- **alt**: ha az applet nem tud futni, akkor ez a szöveg jelenik meg a lapon

### Paraméterek használata

A HTML lapon lehetőségünk van arra, hogy paramétereket adjunk át az appletnek. Ezek a paraméterek származhatnak egy HTML űrlapból, egy PHP kódból...stb.

Használatukhoz a HTML állományban az `<applet>` nyitó címke és a `</applet>` záró címke között meg kell adni a paraméter nevét (ezzel a névvel fogunk rá hivatkozni a kódban) és a paraméter értékét. A paramétert a **`<param name="..." value="..."/>`** kombinált címkével definiálhatjuk.

A Java kódban a paraméterek értékeit az Applet osztály `getParameter()` metódusával kérhetjük le.

Például:

A fent leírt HTML állományban szereplő paramétereket Java-ban a következőképpen érhetjük el:

```
String param1=getParameter("vezeteknev");  
String param2=getParameter("keresztnev");
```

## 2.5 Applet biztonsága

Az appletek általában egy szerveren tárolódnak és a felhasználók távolról kezdeményezik használatukat. Ilyenkor az applet letöltődik a felhasználó gépére és annak böngészőjében (pontosabban a böngésző virtuális gépének segítségével) futnak. Óriási biztonsági kockázatot jelentene a felhasználóknak, ha az általuk letöltött appletek bármit megtehetnének a számítógépükön. Ezért az appletek szigorú biztonsági szabályoknak megfelelően működnek. Egy félév is kevés lenne ahhoz, hogy a Java biztonsági rendszerét tanulmányozzuk, ezért csak a legfontosabb jellemzőket említjük meg.

- egy applet csak azzal a szerverrel kommunikálhat, amelyről letöltötték
- az appletek nem férhetnek hozzá a helyi gép állományrendszeréhez
- stb...stb...

## 3. Grafika használata

Egy applet a `paint()` metódus segítségével rajzol a képernyőre (vonalat, szöveget, képet, színes hátteret, stb.). A rajzolás nagyon sokszor előfordulhat egy applet életében. Ez a metódus hívódik meg, ha például egy appletet tartalmazó ablakot arrébb mozgatunk a képernyőn, illetve bármely olyan esetben, amikor az appletet újra kell rajzolni. Az applet írójának legtöbbször ezt a metódust kell felülírnia.

Figyeljük meg, hogy ennek a metódusnak - a többitől eltérően - van paramétere, a **Graphics** nevű osztály egy példánya. Mivel a Graphics osztály a *java.awt* csomag része, ezért ne feledkezzünk el az importálásról.

Példa

Az alábbi nagyon egyszerű példa egy piros téglalapot rajzol ki.

```
import java.applet.*;  
import java.awt.*;  
  
public class Pelda extends Applet  
{  
  
    public void paint(Graphics g)  
    {  
        g.setColor(Color.red);  
        g.drawRect(20,20,40,40);  
    }  
}
```

A Graphics osztály néhány hasznos metódusa:

- drawArc() – ív
- drawImage() – kép
- drawLine() - vonal
- drawOval() – ellipszis
- drawRect() – téglalap
- drawString() – szöveg
- fillOval() – kitöltött ellipszis
- fillRect() – kitöltött téglalap
- setColor() – a rajzolási szín beállítása
- ...

(További metódusok és azok használata megtalálható a HELP java.awt.Graphics osztály leírásánál.)

Megjegyzések:

- Ahhoz hogy grafikáinkat színesebbé tehesük, tanulmányozzuk a java.awt csomag Color osztályának változóit és metódusait!
- Rajzoláskor a grafikus objektum bal felső sarka tekintendő a (0,0) koordinátának.
- Az appletek egy grafikus felületet nyújtanak számunkra, amelyre közvetlenül rajzolhatunk. Ezen a felületen használhatjuk az előző fejezetben megismert grafikus komponenseket (AWT komponenseket) és az eseményvezérelt programozás kapcsán elsajátított ismereteinket.
- Másrészt az is elmondható, hogy az előző órán bemutatott grafikus felhasználói felülettel rendelkező „normál” java alkalmazásokban (applications) is használhatók azok a grafikus rajzolást biztosító eszközök, amelyeket ebben a fejezetben tekintettünk át.