

Programozás III

LOGGOLÁS

NAPLÓZÁS

Egy generált try - catch:

```
BufferedReader adatBevitel =  
    new BufferedReader(new InputStreamReader(System.in));  
  
try {  
    int szam = Integer.valueOf(adatBevitel.readLine());  
} catch (IOException ex) {  
    Logger.getLogger(LoggPelda.class.getName()).log(Level.SEVERE, null, ex);  
}
```

↑
Ez mi?

Log fájl helye (win7):

C:\Users\xy\AppData\Roaming\NetBeans\8.1\var\log\messages.log

NAPLÓZÁS

Egy program futását célszerű nyomon követni.

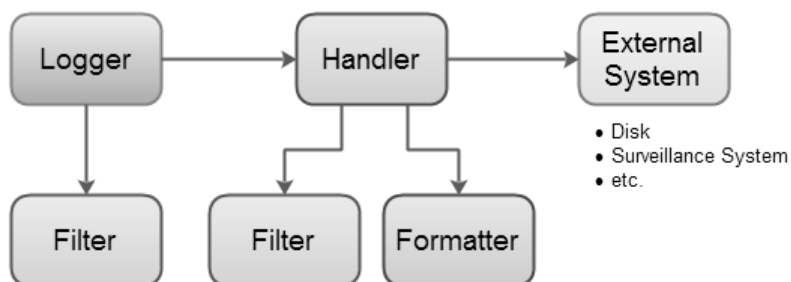
debug
naplózás

Bár időnként szükség van a felhasználónak szóló hiba-
üzenetekre is, de sokszor célszerű log fájlba írni a program
futása közben történeteket.

Erre szolgál a java.util.logging csomag.

NAPLÓZÁS

Áttekintő ábra:



NAPLÓZÁS - PÉLDA

```
class LoggPelda {  
  
    // létrehozunk egy loggert.  
    private static final Logger logger =  
        Logger.getLogger(LoggPelda.class.getName());  
  
    private FileHandler fileHandler;
```

```
void indit() {  
    try {  
        // Létrehozzuk a fájl-handlert, és megadjuk  
        // a log fájl nevét (elérési útját)  
        fileHandler = new FileHandler("logfile.txt");  
  
        // Hozzáadjuk a loggerhez a handlert  
        logger.addHandler(fileHandler);  
  
        // Beállítjuk a formázást  
        SimpleFormatter formatter = new SimpleFormatter();  
        fileHandler.setFormatter(formatter);  
  
        // Üdvözlő szöveg  
        logger.log(Level.SEVERE, "Hello, elindult a loggolás");  
  
        beolvasas();  
    } catch (IOException | SecurityException ex) {  
        logger.log(Level.SEVERE, "Hiba a handler kezelésekor");  
    }  
}
```

NAPLÓZÁS - PÉLDA

```
private void beolvasas() {  
  
    try {  
        BufferedReader adatBevitel =  
            new BufferedReader(new InputStreamReader(System.in));  
        System.out.print("Kérem a számot: ");  
        int szam = Integer.valueOf(adatBevitel.readLine());  
  
        // Csak a példa kedvéért írjuk ki.  
        logger.log(Level.SEVERE, "A szám: " + szam);  
  
        // Az előző string-konkatenáció helyett ezt javasolják  
        logger.log(Level.SEVERE, "A sz\u00e9lm: {0}", szam);  
  
    } catch (IOException | NumberFormatException ex) {  
        logger.log(Level.SEVERE, "Hiba a beolvas\u00e1skor: {0}",  
            ex.getMessage());  
    }  
}
```

NAPLÓZÁS - PÉLDA

A log-fájl tartalma:

```
nov. 21, 2017 4:04:29 DE loggolas.LoggPelda indit  
INFO: Hello, elindult a loggolás  
nov. 21, 2017 4:04:35 DE loggolas.LoggPelda beolvasas  
INFO: A szám: 5  
nov. 21, 2017 4:04:35 DE loggolas.LoggPelda beolvasas  
INFO: A szám: 5  
nov. 21, 2017 4:04:58 DE loggolas.LoggPelda beolvasas  
SEVERE: Hiba a beolvasáskor: For input string: "q"
```

A Nb saját log fájljának helye (win7):

C:\Users\...\AppData\Roaming\NetBeans\8.2\var\log\messages.log

NAPLÓZÁS - PÉLDA

Komolyabb program írásakor célszerű az Apache log4j programját használni.

```
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-core</artifactId>
  <version>2.5</version>
</dependency>
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-api</artifactId>
  <version>2.5</version>
</dependency>
```

NAPLÓZÁS - PÉLDA

Néhány jónak tűnő link:

<http://tutorials.jenkov.com/java-logging/index.html>

<http://bits.netbeans.org/dev/javadoc/org-openide-util/org/openide/util/doc-files/logging.html>

<http://www.vogella.com/articles/Logging/article.html>

<http://www.tutorialspoint.com/log4j/>

+ google ☺