

제 25 강 : Runtime, System

※ 학습목표

- ✓ Runtime 클래스를 활용할 수 있다.
- ✓ 외부 명령어를 실행 시킬 수 있다.
- ✓ System 클래스를 활용할 수 있다.
- ✓ 프로퍼티를 활용해서 시스템 정보를 알아낼 수 있다.

1. Runtime Class

- ✓ 자바에서 기본적으로 제공해주는 클래스
- ✓ 실행시간에 실행환경에 대한정보를 얻거나, 명령을 수행

① 사용방법

✓ `Runtime rt = Runtime.getRuntime();`

② 주요메소드

- ✓ `freeMemory()`: 현재 자바가상머신(JVM)에서 사용가능 메모리량 byte 단위
- ✓ `totalMemory()`: 전체 메모리량을 byte 단위
- ✓ 인수로 받는 문자열을 명령으로 인식, 외부 프로그램을 실행
 - ✓ `exec("c:\\windows\\system32\\calc.exe");`

[실습]

```
1 package tommy.java.exam01;
2
3 public class RuntimeEx {
4     public static void main(String args[]) {
5         Runtime rt = Runtime.getRuntime();
6         long fr = rt.freeMemory();
7         System.out.println("freeMemory=" + fr / 1024 + "KB");
8         long fr2 = rt.totalMemory();
9         System.out.println("totalMemory =" + fr2 / 1024 + "KB");
```

10	try {
11	rt.exec("C:\\windows\\system32\\calc.exe");
12	} catch (Exception e) {
13	System.out.println(e.toString());
14	}
15	}
16	}

2. System Class

✓ 표준입출력을 포함 시스템 전반에 걸친 제어

① 사용방법

✓ System.메서드

② 주요메소드

✓ System.in.read();

✓ System.out.print();,

✓ System.currentTimeMillis();

✓ System.exit(int status);

✓ System.gc();

③ 현재시스템의 환경속성

✓ System.getProperty(String key);

✓ Key : "os.name" , "java.version" , ...

[실습]

1	package tommy.java.exam02;
2	
3	public class SystemEx {
4	public static void main(String args[]) {
5	long startLoopTime = System.currentTimeMillis();
6	for (int i = 0; i < 1000000; i++) {
7	for (int j = 0; j < 100; j++) {
8	int k = i * j;
9	}
10	}
11	long ttime = System.currentTimeMillis() - startLoopTime;

```

12      System.out.println(" 소요시간=" + ttime + "m sec");
13      byte buffer[] = new byte[255];
14      System.out.println("\nType a line of text: ");
15      try {
16          System.in.read(buffer, 0, 255);
17      } catch (Exception e) {
18          System.out.println(e.toString());
19      }
20      System.out.println(new String(buffer)); // byte[] -> String
21  }
22  }

```

[실습] : 시스템 정보 알아내기

```

1  package tommy.java.exam03;
2
3  public class SystemPropertyEx {
4      private static String[][] properties = {
5          { "java.version", "Java Runtime Environment version" },
6          { "java.vendor", "Java Runtime Environment vendor" },
7          { "java.vendor.url", "Java Vendor's URL" },
8          { "java.home", "Java installation directory" },
9          { "java.vm.specification.version", "Java Virtual Machine specification version" },
10         { "java.vm.specification.vendor", "Java Virtual Machine specification vendor" },
11         { "java.vm.specification.name", "Java Virtual Machine specification name" },
12         { "java.vm.version", "Java Virtual Machine implementation version" },
13         { "java.vm.vendor", "Java Virtual Machine implementation vendor" },
14         { "java.vm.name", "Java Virtual Machine implementation name" },
15         { "java.specification.version", "Java Runtime Environment specification version" },
16         { "java.specification.vendor", "Java Runtime Environment specification vendor" },
17         { "java.specification.name", "Java Runtime Environment specification name" },
18         { "java.class.version", "Java class format version number" },
19         { "java.class.path", "Java class path" },
20         { "os.name", "Operating system name" },
21         { "os.arch", "Operating system architecture" },
22         { "os.version", "Operating system version" },
23         { "file.separator", "File separator" },
24         { "path.separator", "Path separator" }, { "line.separator", "Line separator" },
25         { "user.name", "User's account name" },
26         { "user.home", "User's home directory" },
27         { "user.dir", "User's current working directory" }
28     };
29
30     public static void main(String[] arg) {
31         String propValue;
32         System.out.println("Current Java properties:");

```

```

33      System.out.println("-----");
34      for (int i = 0; i < properties.length; i++) {
35          try {
36              propValue = System.getProperty(properties[i][0]);
37              System.out.print(properties[i][1] + " = ");
38              if (properties[i][0].equals("line.separator")) {
39                  for (int j = 0; j < propValue.length(); j++) {
40                      if (propValue.charAt(j) == '\n')
41                          System.out.print("\n");
42                      if (propValue.charAt(j) == '\r')
43                          System.out.print("\r");
44                  }
45                  System.out.println("");
46              } else {
47                  if (propValue != null)
48                      System.out.println(propValue);
49                  else
50                      System.out.println("--NOT DEFINED--");
51              }
52          } catch (SecurityException se) {
53              System.err.println("Could not get value of: " +
54                               properties[i][0]);
55              System.err.println(se.toString());
56          }
57      }
58  }

```