

Java



System 클래스

System 클래스에 대하여
System 클래스의 사용 방법



System 클래스

01. System 클래스에 대하여

System 클래스

•• System 클래스 : 운영체제와의 상호작용에 필요한 여러 기능들을 모아놓은 클래스

| System 클래스가 제공하는 기능 | 설명 |
|---------------------|--------------------------------|
| 표준 입출력 | 키보드로부터의 입력과 모니터로의 출력 기능 |
| 환경 변수 읽기 | 운영체제에 설정되어 있는 환경 변수를 읽어오는 기능 |
| 시스템 프로퍼티 읽기 | 프로그램의 환경 모드를 프로퍼티 형태로 읽고 쓰는 기능 |
| 현재 시각 측정 | 시스템 시계로부터 현재 시각을 읽어오는 기능 |
| 프로그램 실행 관련 기능 | 프로그램을 끝내는 기능과 가비지 컬렉터 관련 기능 |
| 보안 설정 기능 | 자바 프로그램의 보안 관리자 설정 기능 |
| 그 밖의 유용한 기능 | 배열을 효율적으로 복사하는 기능 등 |

System 클래스

- 02. System 클래스의 사용 방법
- System 클래스의 API 규격서

The screenshot shows the Java 2 Platform SE 5.0 API documentation for the `System` class. The browser window title is "System (Java 2 Platform SE 5.0) - Microsoft Internet Explorer". The address bar shows the URL `http://java.sun.com/j2se/1.5.0/docs/api/index.html`. The left sidebar contains a list of Java packages and classes, with `System` selected. The main content area is divided into two sections: "Field Summary" and "Method Summary".

Field Summary

| Field | Description |
|-------------------------------------|-------------------------------------|
| <code>static PrintStream err</code> | The "standard" error output stream. |
| <code>static InputStream in</code> | The "standard" input stream. |
| <code>static PrintStream out</code> | The "standard" output stream. |

Method Summary

| Method | Description |
|--|--|
| <code>static void arraycopy(Object src, int srcPos, Object dest, int destPos, int length)</code> | Copies an array from the specified source array, beginning at the specified position, to the specified position of the destination array. |
| <code>static String clearProperty(String key)</code> | Removes the system property indicated by the specified key. |
| <code>static long currentTimeMillis()</code> | Returns the current time in milliseconds. |
| <code>static void exit(int status)</code> | Terminates the currently running Java Virtual Machine. |
| <code>static void gc()</code> | Runs the garbage collector. |
| <code>static Map<String,String> getEnv()</code> | Returns an unmodifiable string map view of the current system environment. |
| <code>static String getEnv(String name)</code> | Gets the value of the specified environment variable. |
| <code>static Properties getProperties()</code> | Determines the current system properties. |
| <code>static String getProperty(String key)</code> | Gets the system property indicated by the specified key. |
| <code>static String getProperty(String key, String def)</code> | Gets the system property indicated by the specified key. |
| <code>static SecurityManager getSecurityManager()</code> | Gets the system security interface. |
| <code>static int identityHashCode(Object x)</code> | Returns the same hash code for the given object as would be returned by the default method <code>hashCode()</code> , whether or not the given object's class overrides <code>hashCode()</code> . |
| <code>static Channel inheritedChannel()</code> | Returns the channel inherited from the entity that created this Java virtual machine. |
| <code>static void load(String filename)</code> | |

System 클래스

02. System 클래스의 사용 방법

System 클래스의 사용 방법

- 정적 필드와 정적 메소드로만 구성 -> 객체를 만들지 않고 사용
 - [예]

System.exit(0);

↑ ↑

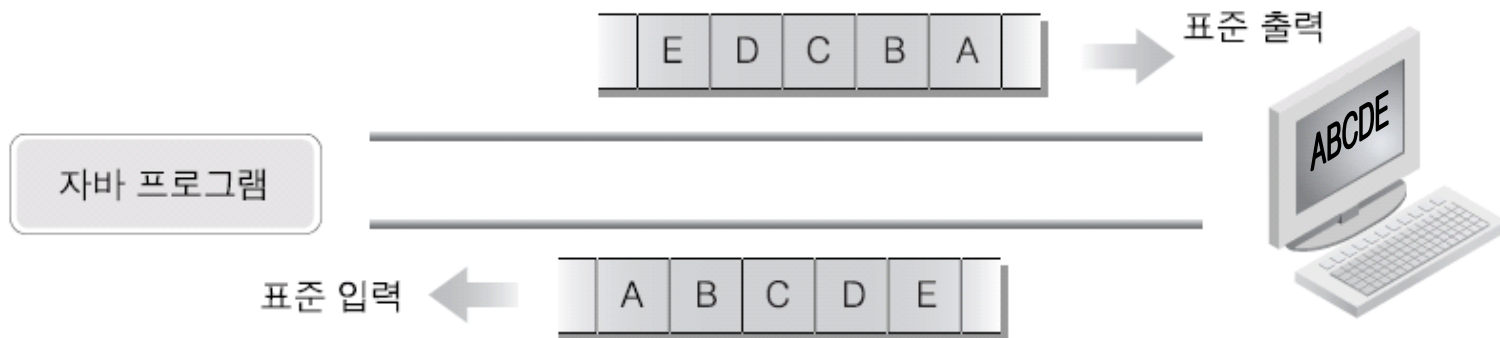
클래스 이름 메소드 이름

System 클래스

02. System 클래스의 사용 방법

표준 입출력

• 표준 입력과 표준 출력의 개념도



• System.in 필드는 표준 입력 처리, System.out 필드는 표준 출력에 사용됨

System 클래스

02. System 클래스의 사용 방법

표준 입력 필드 : in

• 키보드로부터 입력을 받는 방법

- 1) System.in 필드를 가지고 InputStreamReader 객체를 생성합니다.

```
InputStreamReader reader = new InputStreamReader(System.in);
```

System 클래스의 in 필드를 InputStreamReader 클래스의
생성자 파라미터로 넘겨주어야 합니다.

System 클래스

02. System 클래스의 사용 방법

표준 입력 필드 : in

• 키보드로부터 입력을 받는 방법

- 2) InputStreamReader 객체에 대해 read 메소드를 호출합니다.

```
char ch = (char) reader.read();
```

↑ ↑
read 메서드의 리턴 값은 int 타입이므로
char 타입으로 캐스트해야 합니다.

System 클래스

02. System 클래스의 사용 방법

표준 입력 필드 : in

- [예제 14-1] 키보드로부터 문자를 입력받는 프로그램 (1)

프로그램을 실행하면 키보드로부터
입력을 기다리는 상태가 됩니다

이 상태에서 문자열을 입력하고

System 클래스의 in 필드를 가지고
InputStreamReader 객체를 생성함

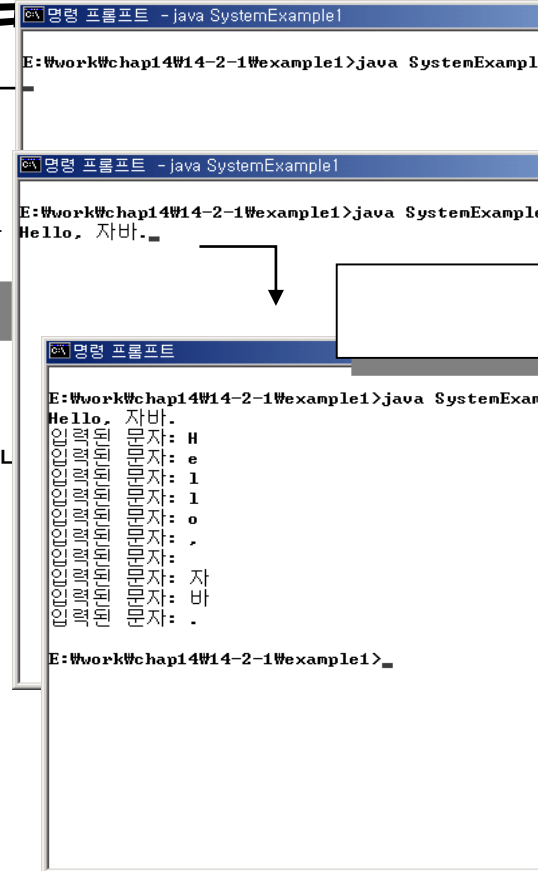
키보드로부터 한 글자씩 입력
받아서 출력하는 일을 마침표(.)가
입력될 때까지 반복합니다.

read 메서드가 발생하는
IOException을 처리합니다.

```

1  import java.io.*;
2  class SystemExample1 {
3      public static void main(String args[]) {
4          InputStreamReader reader =
5              new InputStreamReader(System.in);
6          try {
7              while(true) {
8                  char ch = (char) reader.read();
9                  System.out.println("입력된 문자: " + ch);
10                 if (ch == '.')
11                     break;
12             }
13         }
14         catch (IOException e) {
15             System.out.println("키보드 입력 에러");
16         }
17     }
18 }

```



System 클래스

02. System 클래스의 사용 방법

표준 입력 필드 : in

•• 행 단위로 표준 입력을 받는 방법

- 1) System.in 필드를 가지고 BufferedReader 객체를 생성합니다.

```
InputStreamReader reader1 = new InputStreamReader(System.in);
```

InputStreamReader 객체를 BufferedReader 생성자의
파라미터로 사용해야 합니다.

```
BufferedReader reader2 = new BufferedReader(reader1);
```

System 클래스

● 02. System 클래스의 사용 방법

● 표준 입력 필드 : in

● 키보드로부터 입력을 받는 방법

- 2) BufferedReader 객체에 대해 readLine 메소드를 호출합니다.

```
String str = reader2.readLine();
```

↑
입력 문자를 행 단위로
읽어들이는 메소드

System 클래스

02. System 클래스의 사용 방법

표준 입력 필드 : in

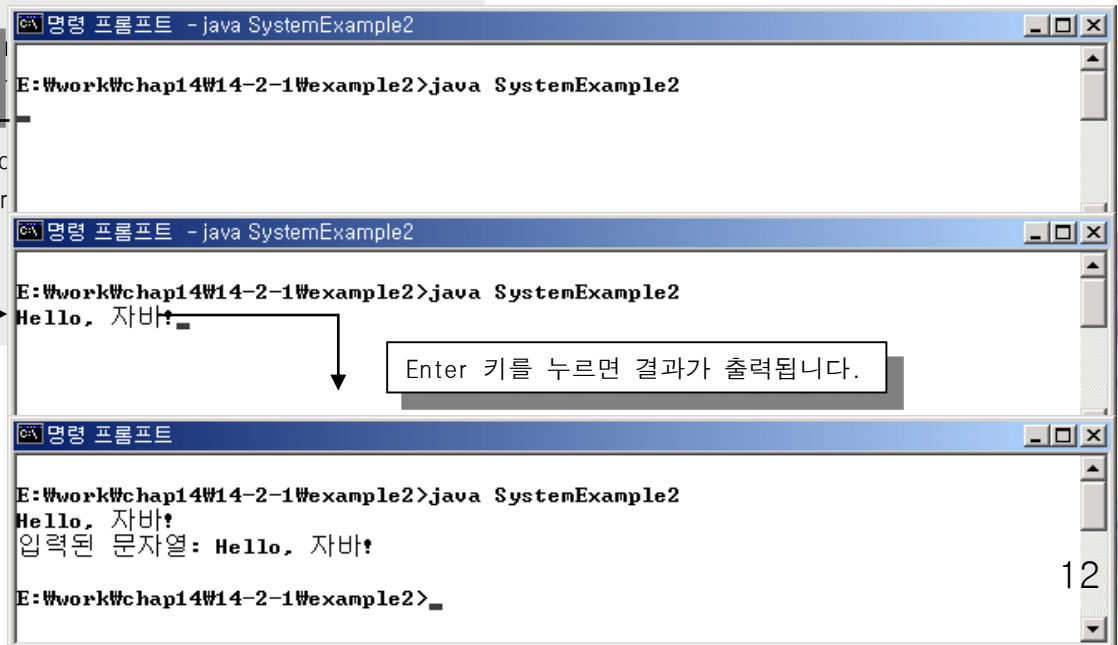
- [예제 14-2] 키보드로부터 문자를 입력받는 프로그램 (2)

```
1 import java.io.*;
2 class SystemExample2 {
3     public static void main(String args[]) {
4         BufferedReader reader = new BufferedReader(
5             new InputStreamReader(System.in));
```

System 클래스의 in 필드를 가지고
BufferedReader 객체를 생성합니다.

프로그램을 실행하면 키보드로부터
입력을 기다리는 상태가 됩니다.

이 상태에서 문자열을 입력하고



System 클래스

02. System 클래스의 사용 방법

표준 출력 필드 : out

• 모니터로 출력을 하는 방법

- System.out 필드에 대해 println, print, printf 메소드를 호출합니다.

```
System.out.println("Hello, Java");  
System.out.print(12.5);  
System.out.printf("%d", 27);
```

System 클래스의 out 필드

System 클래스

02. System 클래스의 사용 방법

표준 에러 출력 필드 : err

- err 필드 : 표준 에러 출력(모니터로의 에러 메시지 출력)에 사용되는 필드
- err 필드의 사용 방법
 - - out 필드의 사용 방법과 동일

```
System.err.println("잘못된 포맷입니다.");  
System.err.print(-2.5);  
System.err.printf("%d", 0);
```

System 클래스

02. System 클래스의 사용 방법

표준 에러 출력 필드 : err

- [예제 14-3] 표준 출력과 표준 에러 출력을 모두 하는 프로그램

```
1 class SystemExample3 {
2     public static void main(String args[]) {
3         int arr1[] = { 77, 82, 99, 100, -27, 0, 42, -35, 60, 72 };
4         int arr2[] = { 7, 0, 3, 0, -1, 2, 11, 5, 0, 9 };
5         for (int cnt = 0; cnt < arr1.length; cnt++) {
6             try {
7                 int result = arr1[cnt] / arr2[cnt];
8                 System.out.printf("%d / %d = %d %n",
9                                     arr1[cnt], arr2[cnt], result);
10            }
11            catch (java.lang.ArithmeticException e) {
12                System.err.println("잘못된 연산입니다. (index=" + cnt + ")");
13            }
14        }
15    }
```

나눗셈의 정상적인 결과를
out 필드를 통해 출력합니다.

나눗셈 중 발생한 에러에
대한 메시지를 err 필드를
통해 출력합니다.

명령 프롬프트

```
E:\work\chap14\14-2-1\example3>java SystemExample3
77 / 7 = 11
잘못된 연산입니다. <index=1>
99 / 3 = 33
잘못된 연산입니다. <index=3>
-27 / -1 = 27
0 / 2 = 0
42 / 11 = 3
-35 / 5 = -7
잘못된 연산입니다. <index=8>
72 / 9 = 8
```

System 클래스

02. System 클래스의 사용 방법

표준 입출력의 방향 전환

표준 출력의 방향 전환

```
E:\work\chap14\14-2-1\example3>java SystemExample3 > output.txt
잘못된 연산입니다. <index=1>
잘못된 연산입니다. <index=3>
잘못된 연산입니다. <index=8>

E:\work\chap14\14-2-1\example3>dir
E 드라이브의 볼륨: 로컬 디스크
볼륨 일련 번호: 305B-17D4

E:\work\chap14\14-2-1\example3 디렉터리

2006-05-07 01:49p <DIR> .
2006-05-07 01:49p <DIR> ..
2006-05-07 02:44a          1,110 SystemExample3.class
2006-05-07 02:43a           605 SystemExample3.java
2006-05-07 01:58p           98 output.txt
                3개 파일          1,813 바이트
                2 디렉터리    1,742,544,896 바이트 남음

E:\work\chap14\14-2-1\example3>
```

명령행의 뒤쪽에 >와 파일명을 쓰면
그 이름으로 파일이 생성됩니다.

```
output.txt - 메모장
파일(F)  편집(E)  서식(O)  도움말(H)

77 / 7 = 11
99 / 3 = 33
-27 / -1 = 27
0 / 2 = 0
42 / 11 = 3
-35 / 5 = -7
72 / 9 = 8
```

그 파일을 열어보면 System 클래스의
out 필드로 출력한 내용이 들어있는 것을
확인할 수 있습니다.

System 클래스

02. System 클래스의 사용 방법

표준 입출력의 방향 전환

• 표준 에러 출력의 방향 전환

```
명령 프롬프트
E:\work\chap14\14-2-1\example3>java SystemExample3 2> error.txt
77 / 7 = 11
99 / 3 = 33
-27 / -1 = 27
0 / 2 = 0
42 / 11 = 3
-35 / 5 = -7
72 / 9 = 8

E:\work\chap14\14-2-1\example3>dir
E 드라이브의 볼륨: 로컬 디스크
볼륨 일련 번호: 305B-17D4

E:\work\chap14\14-2-1\example3 디렉터리

2006-05-07 01:49p    <DIR>        .
2006-05-07 01:49p    <DIR>        ..
                1,110 SystemExample3.class
2006-05-07 02:44a             605 SystemExample3.java
2006-05-07 02:43a             90 error.txt
2006-05-07 02:02p    3개 파일             1,805 바이트
                    2 디렉터리 1,742,069,760 바이트 남음

E:\work\chap14\14-2-1\example3>
```

명령행의 뒤쪽에 2>와 파일명을 쓰면
그 이름으로 파일이 생성됩니다.

```
error.txt - 메모장
파일(F)  편집(E)  서식(O)  도움말(H)
잘못된 연산입니다. (index=1)
잘못된 연산입니다. (index=3)
잘못된 연산입니다. (index=8)
```

그 파일을 열어보면 System 클래스의
err 필드로 출력한 내용이 들어있는 것을
확인할 수 있습니다.

System 클래스

● 02. System 클래스의 사용 방법

● 표준 입출력의 방향 전환

- 표준 출력과 표준 에러 출력의 방향을 모두 전환하는 방법

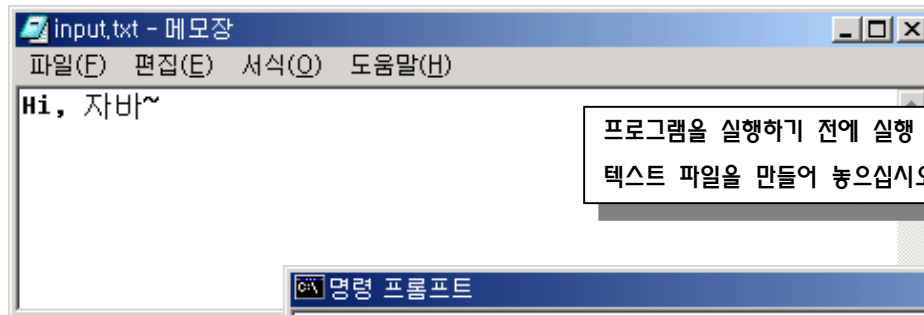
```
java SystemExample3 > output.txt 2> error.txt
```

System 클래스

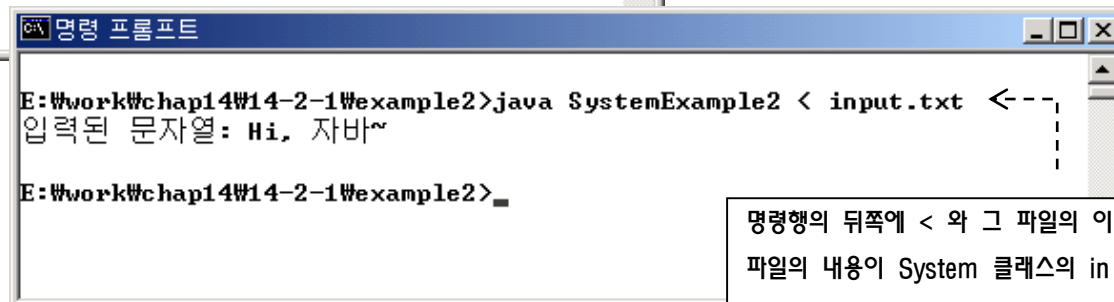
02. System 클래스의 사용 방법

표준 입출력의 방향 전환

.. 표준 입력의 방향 전환



프로그램을 실행하기 전에 실행 디렉토리에
텍스트 파일을 만들어 놓으십시오.



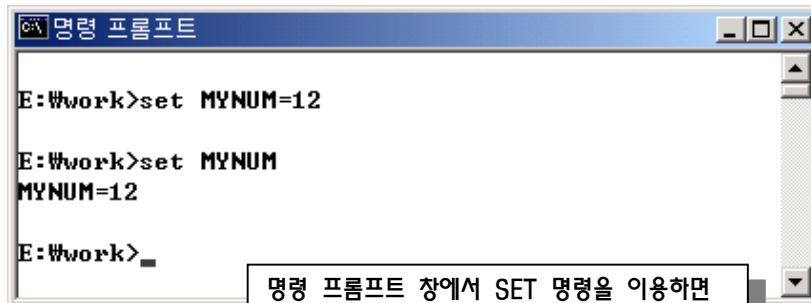
명령행의 뒤쪽에 < 와 그 파일의 이름을 쓰면
파일의 내용이 System 클래스의 in 필드를
통해 프로그램에 입력됩니다.

System 클래스

02. System 클래스의 사용 방법

환경 변수

• 환경 변수를 읽고 쓰는 방법



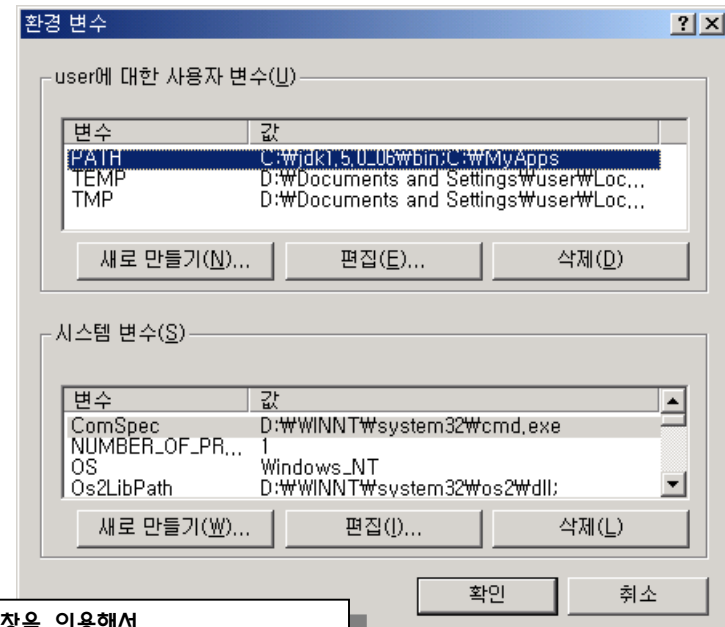
```
명령 프롬프트

E:\work>set MYNUM=12

E:\work>set MYNUM
MYNUM=12

E:\work>
```

명령 프롬프트 창에서 SET 명령을 이용하면
환경 변수의 값을 읽고 쓸 수 있습니다.



환경변수 창을 이용해서
환경 변수의 값을 읽고 쓸 수도 있습니다.

System 클래스

02. System 클래스의 사용 방법

getenv 메소드

- `getenv` 메소드 : 환경 변수를 읽는 메소드

```
String str = System.getenv("path");
```

↑
path라는 이름의 환경 변수의 값을 리턴합니다.

System 클래스

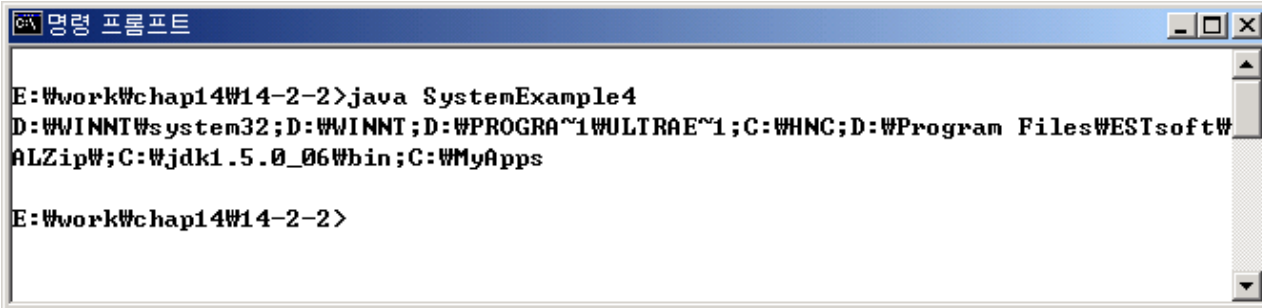
02. System 클래스의 사용 방법

환경 변수를 읽는 메소드

- [예제 14-4] path 환경 변수의 값을 읽어오는 프로그램

```
1 class SystemExample4 {  
2     public static void main(String args[]) {  
3         String str = System.getenv("path");  
4         System.out.println(str);  
5     }  
6 }
```

환경변수 path의 값을 읽어옵니다.



```
C:\>명령 프롬프트  
E:\work\chap14\14-2-2>java SystemExample4  
D:\WINNT\system32;D:\WINNT;D:\PROGRAM~1\ULTRA~1;C:\WINC;D:\Program Files\ESTsoft\ALZip;  
C:\jdk1.5.0_06\bin;C:\MyApps  
E:\work\chap14\14-2-2>
```

System 클래스

02. System 클래스의 사용 방법

시스템 프로퍼티를 읽는 메소드

•• 모든 시스템 프로퍼티를 한꺼번에 읽는 방법

- 1) `System.getProperties` 메소드를 호출해서 `Properties` 객체를 얻습니다.

```
Properties props = System.getProperties();
```

모든 시스템 프로퍼티를 읽어오는 메소드

- 2) `Properties` 객체에 대해 `list` 메소드를 호출합니다.

```
props.list(System.out);
```

이 객체에 있는

모든 프로퍼티들을
이리로 출력합니다.

System 클래스

02. System 클래스의 사용 방법

시스템 프로퍼티를 읽는 메소드

- [예제 14-5] 시스템 프로퍼티를 모두 가져와서 출력하는 프로그램

```
1 import java.util.Properties;
2 class SystemExample5 {
3     public static void main(String args[]) {
4         Properties props = System.getProperties(); ----- 시스템 프로퍼티를 모두 가져옵니다.
5         props.list(System.out); ----- 가져온 시스템 프로퍼티를 출력합니다.
6     }
7 }
```

```
E:\명령 프롬프트
E:\work\chap14\14-2-3>java SystemExample5
-- listing properties --
java.runtime.name=Java(TM) 2 Runtime Environment
sun.boot.library.path=D:\Program Files\Java\jre1
java.vm.version=1.5.0_06-b05
java.vm.vendor=Sun Microsystems Inc.
java.vendor.url=http://java.sun.com/
path.separator=;
java.vm.name=Java HotSpot(TM) Client VM
file.encoding.pkg=sun.io
user.country=KR
sun.os.patch.level=Service Pack 4
java.vm.specification.name=Java Virtual Machine
user.dir=E:\work\chap14\14-2-3
java.runtime.version=1.5.0_06-b05
java.awt.graphicsenv=sun.awt.Win32GraphicsEnviro
java.endorsed.dirs=D:\Program Files\Java\jre1.5
os.arch=x86
java.io.tmpdir=D:\DOCUME~1\user\LOCALS~1\Temp\
line.separator=

java.vm.specification.vendor=Sun Microsystems In
user.variant=
os.name=Windows 2000
sun.jnu.encoding=MS949
java.library.path=D:\WINNT\system32\.;D:\WINNT\
java.specification.name=Java Platform API Specif
java.class.version=49.0
sun.management.compiler=HotSpot Client Compiler
os.version=5.0
user.home=D:\Documents and Settings\user
user.timezone=
java.awt.printerjob=sun.awt.windows.WPrinterJob
file.encoding=MS949
java.specification.version=1.5
user.name=user
java.class.path=.
java.vm.specification.version=1.0
sun.arch.data.model=32
java.home=D:\Program Files\Java\jre1.5.0_06
java.specification.vendor=Sun Microsystems Inc.
user.language=ko
awt.toolkit=sun.awt.windows.WToolkit
java.vm.info=mixed mode, sharing
java.version=1.5.0_06
java.ext.dirs=D:\Program Files\Java\jre1.5.0_06\
sun.boot.class.path=D:\Program Files\Java\jre1.5
java.vendor=Sun Microsystems Inc.
file.separator=\
java.vendor.url.bug=http://java.sun.com/cgi-bin/
sun.cpu.endian=little
sun.io.unicode.encoding=UnicodeLittle
sun.desktop=windows
sun.cpu.isalist=pentium_pro+mmx pentium_pro pent
```


System 클래스

02. System 클래스의 사용 방법

시스템 프로퍼티를 읽는 메소드

System (Java 2 Platform SE 5.0) - Microsoft Internet Explorer

주소: <http://java.sun.com/j2se/1.5.0/docs/api/index.html>

getProperties

```
public static Properties getProperties()
```

Determines the current system properties.

First, if there is a security manager, its `checkPropertiesAccess` method is called with no arguments. This may result in a security exception.

The current set of system properties for use by the `getProperty(String)` method is returned as a `Properties` object. If there is no current set of system properties, a set of system properties is first created and initialized. This set of system properties always includes values for the following keys:

| Key | Description of Associated Value |
|--|--|
| <code>java.version</code> | Java Runtime Environment version |
| <code>java.vendor</code> | Java Runtime Environment vendor |
| <code>java.vendor.url</code> | Java vendor URL |
| <code>java.home</code> | Java installation directory |
| <code>java.vm.specification.version</code> | Java Virtual Machine specification version |
| <code>java.vm.specification.vendor</code> | Java Virtual Machine specification vendor |
| <code>java.vm.specification.name</code> | Java Virtual Machine specification name |
| <code>java.vm.version</code> | Java Virtual Machine implementation version |
| <code>java.vm.vendor</code> | Java Virtual Machine implementation vendor |
| <code>java.vm.name</code> | Java Virtual Machine implementation name |
| <code>java.specification.version</code> | Java Runtime Environment specification version |
| <code>java.specification.vendor</code> | Java Runtime Environment specification vendor |
| <code>java.specification.name</code> | Java Runtime Environment specification name |
| <code>java.class.version</code> | Java class format version number |
| <code>java.class.path</code> | Java class path |
| <code>java.library.path</code> | List of paths to search when loading libraries |
| <code>java.io.tmpdir</code> | Default temp file path |
| <code>java.compiler</code> | Name of JIT compiler to use |
| <code>java.ext.dirs</code> | Path of extension directory or directories |
| <code>os.name</code> | Operating system name |
| <code>os.arch</code> | Operating system architecture |
| <code>os.version</code> | Operating system version |
| <code>file.separator</code> | File separator ("\" on UNIX) |
| <code>path.separator</code> | Path separator (":" on UNIX) |
| <code>line.separator</code> | Line separator ("\\n" on UNIX) |
| <code>user.name</code> | User's account name |
| <code>user.home</code> | User's home directory |
| <code>user.dir</code> | User's current working directory |

Multiple paths in a system property value are separated by the path separator character of the platform.

Note that even if the security manager does not permit the `getProperties` operation, it may choose to permit the `getProperty(String)` operation.

Returns:
the system properties

System 클래스의 API 규격서에 나와 있는
시스템 프로퍼티의 이름과 설명

System 클래스

02. System 클래스의 사용 방법

시스템 프로퍼티를 읽는 메소드

•• 특정 시스템 프로퍼티를 읽는 방법

- - 시스템 프로퍼티 이름을 가지고 `System.getProperty` 메소드를 호출합니다.

```
String str = System.getProperty("user.dir");
```

↑
특정 시스템 프로퍼티를 읽어오는 메소드

System 클래스

02. System 클래스의 사용 방법

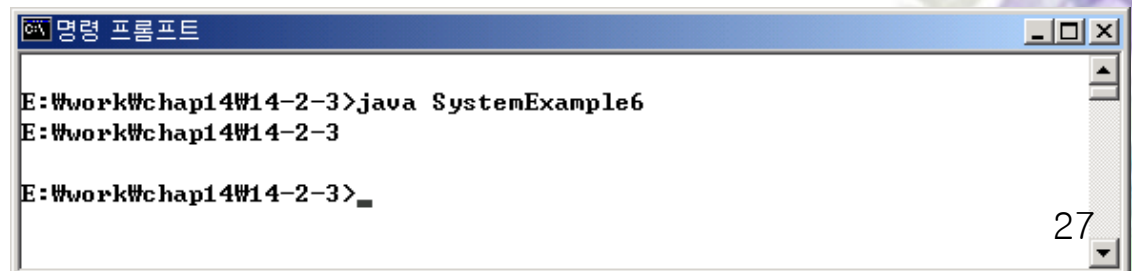
시스템 프로퍼티를 읽는 메소드

- [예제 14-6] 특정 시스템 프로퍼티 값을 가져와서 출력하는 프로그램

```
1  class SystemExample6 {  
2      public static void main(String args[]) {  
3          String str = System.getProperty("user.dir");  
4          System.out.println(str);  
5      }  
6  }
```

user.dir 시스템 프로퍼티의
값을 가져옵니다.

가져온 값을 출력합니다.



```
명령 프롬프트  
E:\work\chap14\14-2-3>java SystemExample6  
E:\work\chap14\14-2-3  
E:\work\chap14\14-2-3>
```

System 클래스

02. System 클래스의 사용 방법

exit 메소드

- exit 메소드 : 자바 프로그램의 실행을 끝내는 메소드

System.exit(0);

프로그램이 정상 종료할 때는
파라미터로 0을 넘겨줘야 합니다.

System.exit(1);

프로그램이 비정상 종료할 때는
파라미터로 0이 아닌 값을 넘겨줘야 합니다.

System 클래스

02. System 클래스의 사용 방법

exit 메소드

- [예제 14-7] exit 메소드의 사용 예

```
1  class SystemExample7 {
2      public static void main(String args[]) {
3          if (args.length != 2) {
4              System.out.println("Usage: java SystemExample7 <정수> <정수>");
5              System.exit(1); ----- 명령행 파라미터의 수가 안 맞으면 비정상 종료
6          }
7          try {
8              int num1 = Integer.parseInt(args[0]);
9              int num2 = Integer.parseInt(args[1]);
10             int sum = num1 + num2;
11             System.out.println("sum = " + sum);
12             System.exit(0); ----- 할 일을 마치고 정상 종료
13         }
14         catch (NumberFormatException e) {
15             System.err.println("잘못된 숫자입니다.");
16             System.exit(-1); ----- 익셉션이 발생하면 비정상 종료
17         }
18     }
19 }
```

명령 프롬프트

```
E:\work\chap14\14-2-4\example1>java SystemExample7
Usage: java SystemExample7 <정수> <정수>
```

```
E:\work\chap14\14-2-4\example1>java SystemExample7
잘못된 숫자입니다.
```

```
E:\work\chap14\14-2-4\example1>java SystemExample7
sum = 30
```

```
E:\work\chap14\14-2-4\example1>_
```

System 클래스

02. System 클래스의 사용 방법

gc 메소드

- gc 메소드 : 가비지 컬렉터가 더 빨리 일하도록 만드는 모듈

자바 가상 기계 안에 있는 필요없는
객체들을 자동으로 삭제하는 모듈

System.gc();

가비지 컬렉터가 가급적 빨리 일을 하도록
자바 가상 기계에게 요청하는 메소드

System 클래스

02. System 클래스의 사용 방법

gc 메소드

- [예제 14-8] gc 메소드의 사용 예

ResourceUser 클래스

```
1 class ResourceUser {
2     ResourceUser() {
3         System.out.println("시스템 자원을 할당 받습니다.");
4     }
5     void use() {
6         System.out.println("시스템 자원을 사용합니다.");
7     }
8     protected void finalize() {
9         System.out.println("시스템 자원의 할당을 해제합니다.");
10    }
11 }
```

ResourceUser 클래스를 사용하는 프로그램

```
1 class SystemExample8 {
2     public static void main(String args[]) {
3         ResourceUser obj = new ResourceUser();
4         obj.use();
5         obj = null;
6         System.gc();
7     }
8 }
```

ResourceUser 객체를 생성합니다.

ResourceUser 객체를 사용합니다.

ResourceUser 객체를 더 이상 필요치 않은 상태로 만듭니다.

System 클래스의 gc 메서드를 호출합니다.

명령 프롬프트

```
E:\work\chap14\14-2-4\example2>java SystemExample8
시스템 자원을 할당 받습니다.
시스템 자원을 사용합니다.
시스템 자원의 할당을 해제합니다.

E:\work\chap14\14-2-4\example2>
```

System 클래스

02. System 클래스의 사용 방법

runFinalization 메소드

•• runFinalization 메소드

- 불필요한 객체들의 finalize 메소드가 더 빨리 호출되도록 만드는 메소드

System.runFinalization();



가비지 컬렉터가 제거 대상 객체들의
finalize 메소드를 가급적 빨리
호출하도록 요청하는 메소드

System 클래스

02. System 클래스의 사용 방법

arraycopy 메소드

- arraycopy 메소드 : 배열의 일부/전부를 일괄 복사하는 메소드

```
for (int cnt = 0; cnt < 5; cnt++)  
    arr2[cnt + 20] = arr1[cnt + 10];
```

arr1[10]부터 arr1[14]까지의 값을
arr2[20]부터 arr2[24]까지 복사하는
반복문



```
System.arraycopy(arr1, 10, arr2, 20, 5);
```

arr1[10]부터 arr1[14]까지의 값을
arr2[20]부터 arr2[24]까지 복사하는 메소드 호출

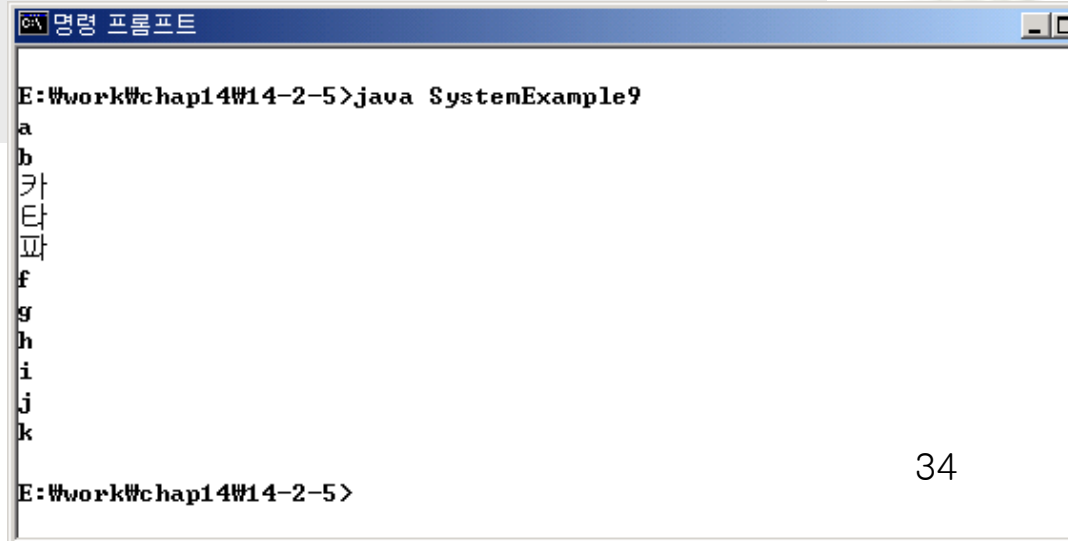
System 클래스

02. System 클래스의 사용 방법

arraycopy 메소드

- [예제 14-9] arraycopy 메소드의 사용 예 (1)

```
1  class SystemExample9 {  
2      public static void main(String args[]) {  
3          char arr1[] = { '가', '나', '다', '라', '마', '바', '사', '아', '자', '차', '카', '타', '파', '하' };  
4          char arr2[] = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k' };  
5          System.arraycopy(arr1, 10, arr2, 2, 3);  
6          for (char ch : arr2)  
7              System.out.println(ch);  
8      }  
9  }
```



```
C:\>명령 프롬프트  
E:\work\chap14\14-2-5>java SystemExample9  
a  
b  
카  
타  
파  
f  
g  
h  
i  
j  
k  
E:\work\chap14\14-2-5>
```

System 클래스

02. System 클래스의 사용 방법

arraycopy 메소드

- 같은 배열 내의 항목들을 복사하는 것도 가능

```
System.arraycopy(arr, 0, arr, 5, 10);
```

↑
arr[0]부터 arr[9]까지의 값을
arr[5]부터 arr[14]까지 복사하는 메소드 호출

```
System.arraycopy(arr, 5, arr, 0, 10);
```

↑
arr[5]부터 arr[14]까지의 값을
arr[0]부터 arr[9]까지 복사하는 메소드 호출

System 클래스

02. System 클래스의 사용 방법

arraycopy 메소드

- [예제 14-10] arraycopy 메소드의 사용 예 (2)

```
1  class SystemExample10 {  
2      public static void main(String args[]) {  
3          int arr[] = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };  
4          System.arraycopy(arr, 0, arr, 3, 5);  
5          for (int num : arr)  
6              System.out.println(num);  
7      }  
8  }
```

arr[0]~ arr[4]의 값을
arr[3] ~ arr[7]에 복사합니다.

C:\ 명령 프롬프트

E:\work\chap14\14-2-5>java SystemExample10

0
1
2
3
4
5
6
7
8
9
10

E:\work\chap14\14-2-5>_

System 클래스

02. System 클래스의 사용 방법

currentTimeMillis 메소드

- currentTimeMillis 메소드 : 시스템 시계로부터 현재 시각을 읽는 메소드

```
long time = System.currentTimeMillis();
```

↑
시스템 시계로부터
현재의 시각을 읽어오는 메소드

- * 이 메소드의 리턴 값은 밀리세컨드(1/1000초) 단위이며 long 타입임

System 클래스

02. System 클래스의 사용 방법

currentTimeMillis 메소드

- [예제 14-11] currentTimeMillis 메소드의 사용 예

```
1  class SystemExample11 {
2      public static void main(String args[]) {
3          long time1 = System.currentTimeMillis(); ----- 현재 시각 측정
4          double total = 0.0;
5          for (int cnt = 1; cnt < 1000000000; cnt += 2)
6              if (cnt / 2 % 2 == 0)
7                  total += 1.0 / cnt;
8              else
9                  total -= 1.0 / cnt;
10         double pi = total * 4;
11         long time2 = System.currentTimeMillis(); ----- 현재 시각 측정
12         System.out.println("result = " + pi);
13         System.out.printf("계산에 %d ms가 소요되었습니다.",
14                             time2 - time1); ----- 두 시각의 차를 출력
15     }
```

명령 프롬프트

```
E:\work\chap14\14-2-6>java SystemExample11
result = 3.141592651589258
계산에 29632 ms가 소요되었습니다.
E:\work\chap14\14-2-6>
```

영

어

화

시

다

^^!