

System 클래스

System 클래스

❖ System 클래스 용도

- 운영체제의 기능 일부 이용 가능
 - 프로그램 종료, 키보드로부터 입력, 모니터 출력, 메모리 정리, 현재 시간 읽기
 - 시스템 프로퍼티 읽기, 환경 변수 읽기

System 클래스

❖ 프로그램 종료(exit())

- 기능 - 강제로 JVM 종료

```
System.exit(0);
```

- int 매개값을 지정하도록 - 종료 상태 값
 - 정상 종료일 경우 0, 비정상 종료일 경우 0 이외 다른 값
 - 어떤 값 주더라도 종료
- 만약 특정 상태 값이 입력되었을 경우에만 종료하고 싶다면?
 - 자바의 보안 관리자 설정

System 클래스

❖ ExitExample.java

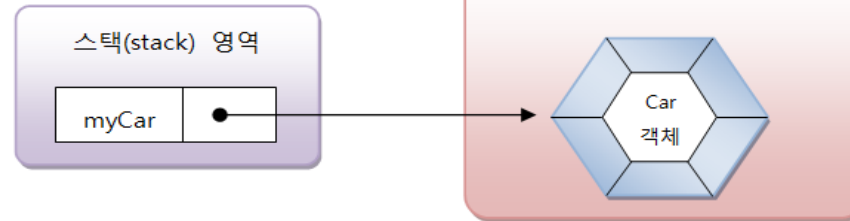
```
public class ExitExample {
    public static void main(String[] args) {
        // 보안 관리자 설정
        System.setSecurityManager(new SecurityManager() {
            @Override
            public void checkExit(int status) {
                if (status != 5) {
                    throw new SecurityException();
                }
            }
        });

        for (int i = 0; i < 10; i++) {
            // i값 출력
            System.out.println(i);
            try {
                // JVM 종료 요청
                System.exit(i);
            } catch (SecurityException e) {
            }
        }
    }
}
```

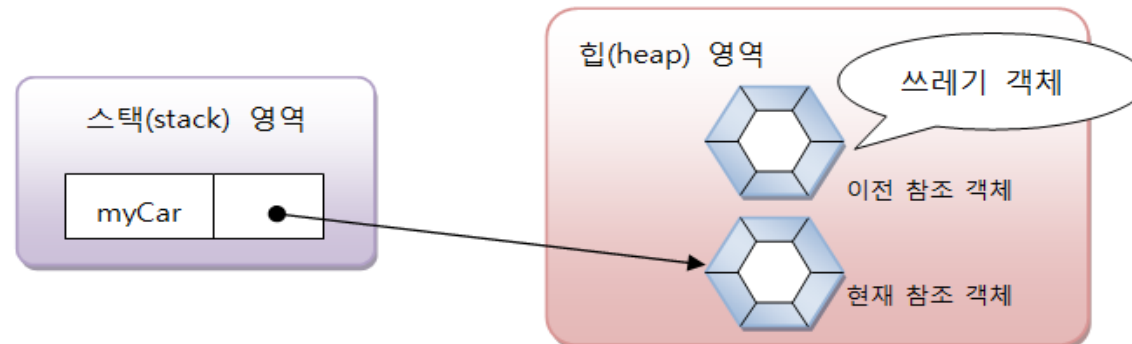
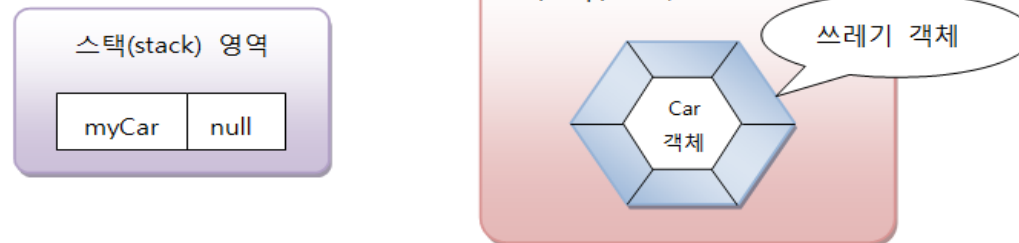
System 클래스

❖ 쓰레기 수집기 실행 요청(gc())

```
Car myCar = new Car();
```



```
myCar = null;
```



System 클래스

❖ Employee.java

```
public class Employee {  
    public int eno;  
  
    public Employee(int eno) {  
        this.eno = eno;  
        System.out.println("Employee(" + eno + ") 가 메모리에 생성됨");  
    }  
  
    public void finalize() {  
        System.out.println("Employee(" + eno + ") 이 메모리에서 제거됨");  
    }  
}
```

System 클래스

❖ GcExample.java

```
public class GcExample {  
    public static void main(String[] args) {  
        Employee emp;  
  
        emp = new Employee(1);  
        emp = null;  
        emp = new Employee(2);  
        emp = new Employee(3);  
  
        System.out.print("emp가 최종적으로 참조하는 사원번호: ");  
        System.out.println(emp.eno);  
        System.gc();  
    }  
}
```

System 클래스

❖ 현재 시각 읽기

- 현재 시간을 읽어 밀리 세컨드(`currentTimeMillis()` -> 1/1000초) 와 나노세컨드(`nanoTime()`->1/10⁹초) 단위의 long값 리턴

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

- 주로 프로그램 실행 소요 시간 구할 때 이용

System 클래스

❖ SystemTimeExample.java

```
public class SystemTimeExample {  
    public static void main(String[] args) {  
        long time1 = System.nanoTime();  
  
        int sum = 0;  
        for (int i = 1; i <= 1000000; i++) {  
            sum += i;  
        }  
  
        long time2 = System.nanoTime();  
  
        System.out.println("1~1000000까지의 합: " + sum);  
        System.out.println("계산에 " + (time2 - time1) +  
            " 나노초가 소요되었습니다.");  
    }  
}
```

System 클래스

❖ 시스템 프로퍼티 읽기(getProperty())

- 시스템 프로퍼티란?
 - JVM이 시작할 때 자동 설정되는 시스템의 속성값
- 대표적인 키와 값

| 키(key) | 설명 | 값(value) |
|----------------|------------------------------|----------------|
| java.version | 자바의 버전 | 1.7.0_25 |
| java.home | 사용하는 JRE 의 파일 경로 | <jdk 설치경로>\jre |
| os.name | Operating system name | Windows 7 |
| file.separator | File separator ("/" on UNIX) | \ |
| user.name | 사용자의 이름 | 사용자계정 |
| user.home | 사용자의 홈 디렉토리 | C:\Users\사용자계정 |
| user.dir | 사용자가 현재 작업 중인 디렉토리 경로 | 다양 |

- 시스템 프로퍼티 읽어오는 법
`System value = System.getProperty(String key);`

System 클래스

❖ GetPropertyExample.java

```
import java.util.Properties;
import java.util.Set;

public class GetPropertyExample {
    public static void main(String[] args) {
        String osName = System.getProperty("os.name");
        String userName = System.getProperty("user.name");
        String userHome = System.getProperty("user.home");

        System.out.println("운영체제 이름: " + osName);
        System.out.println("사용자 이름: " + userName);
        System.out.println("사용자 홈디렉토리: " + userHome);
    }
}
```

System 클래스

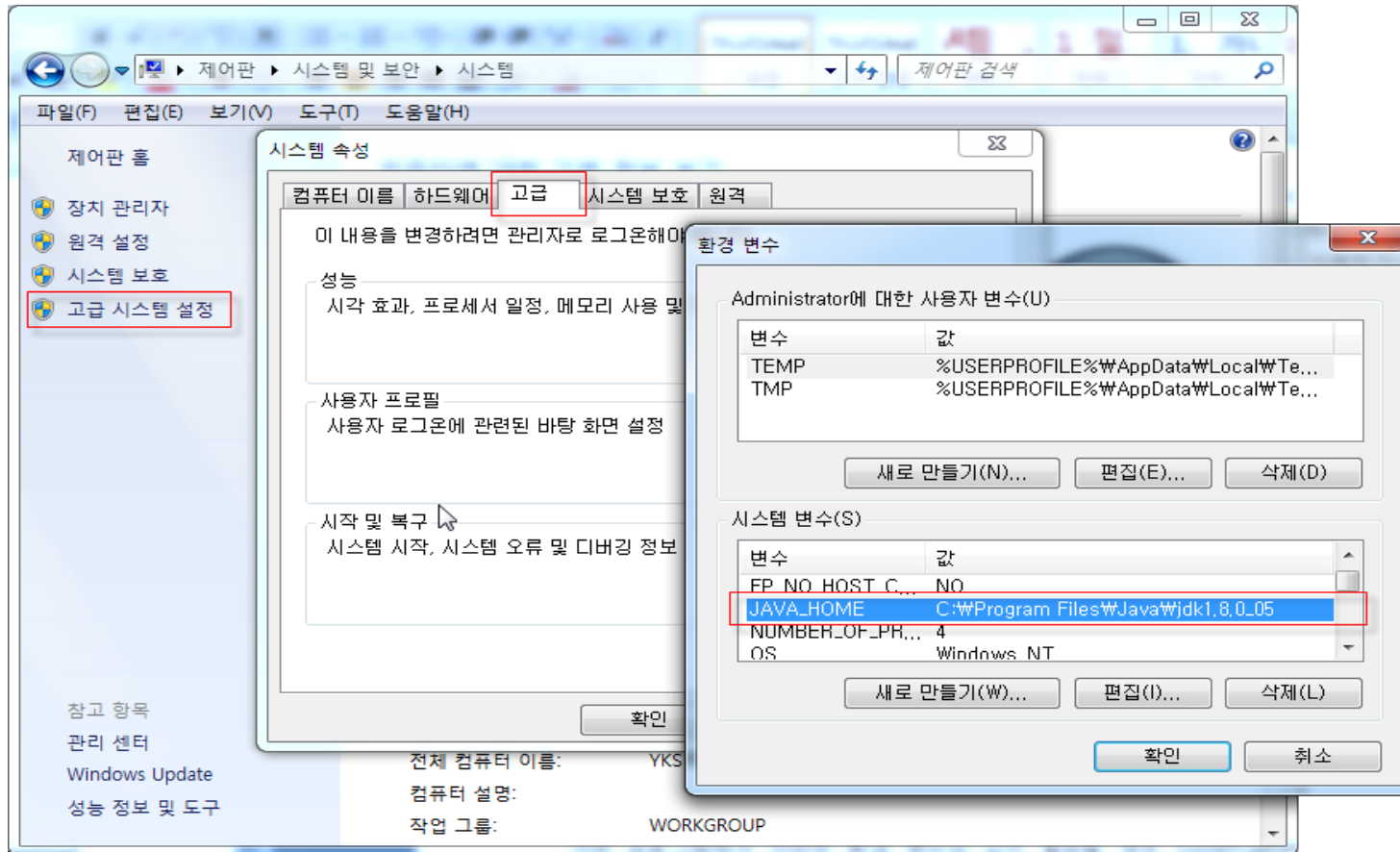
❖ GetPropertyExample.java

```
System.out.println("-----");
System.out.println(" [ key ]  value");
System.out.println("-----");
Properties props = System.getProperties();
Set keys = props.keySet();
for (Object objKey : keys) {
    String key = (String) objKey;
    String value = System.getProperty(key);
    System.out.println("[ " + key + " ]  " + value);
}
}
```

System 클래스

❖ 환경 변수 읽기(getenv())

- 운영체제가 제공하는 환경 변수 값 (문자열) 을 읽음



System 클래스

❖ SystemEnvExample.java

```
public class SystemEnvExample {  
    public static void main(String[] args) {  
        String javaHome = System.getenv("JAVA_HOME");  
        System.out.println("[ JAVA_HOME ] " + javaHome);  
    }  
}
```