



Power Java

제13장 패키지



© 2009 인피니티박스 All rights reserved



이번 장에서 학습할 내용

- 패키지의 개념
- 패키지로 묶는 방법
- 패키지 사용
- 기본 패키지
- 유틸리티 패키지

패키지는
연관된
클래스들을
묶는
기법입니다.



© 2009 인피니티박스 All rights reserved



패키지란?

- 패키지(package) : 클래스들을 묶은 것
- 자바 라이브러리도 패키지로 구성
 - (예) java.net 패키지- 네트워크 관련 라이브러리



그림 13.1 패키지의 개념

© 2009 인피니티박스 All rights reserved



패키지의 장점

- ① 관련된 클래스들을 쉽게 파악
- ② 원하는 클래스들을 쉽게 찾을 수 있다.
- ③ 패키지마다 이름 공간을 따로 갖기 때문에 같은 클래스 이름을 여러 패키지가 사용
- ④ 패키지별로 접근에 제약을 가할 수 있다.

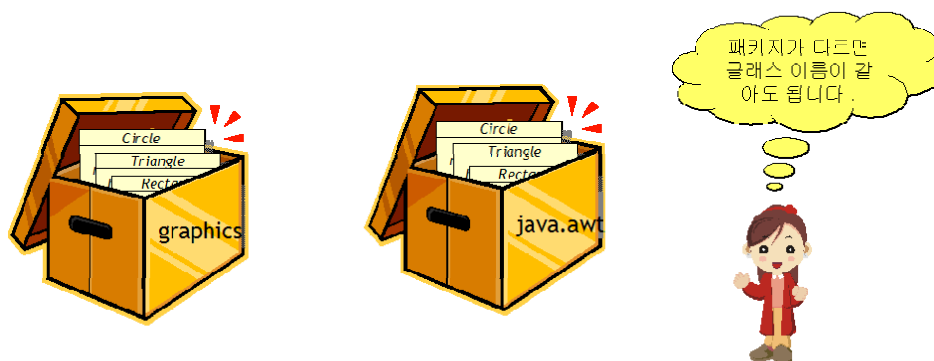


그림 13.2 패키지의 이점

© 2009 인피니티박스 All rights reserved



패키지 생성하기

```
package business; // 패키지 선언
public class Order {
    ...
}
```

Order 라는 클래스는 business 패키지에 속한다.



Q: 만약 패키지 문을 사용하지 않은 경우에는 어떻게 되는가?

A: 디폴트 패키지(default package)에 속하게 된다.

© 2009 인피니티박스 All rights reserved



패키지의 이름

- 인터넷 도메인 이름을 역순으로 사용한다. 예를 들면 com.company.test라는 패키지 이름은 도메인 이름 company.com에서의 test라는 프로젝트를 의미한다.

© 2009 인피니티박스 All rights reserved



소스 파일과 클래스 파일의 위치



Q: 만약 패키지명이 있는 경우에 소스 파일과 클래스 파일의 위치는?

A: 패키지 이름이 디렉토리가 되고 그 아래에 저장된다.

```

package business;
public class Order {
    ...
}

```

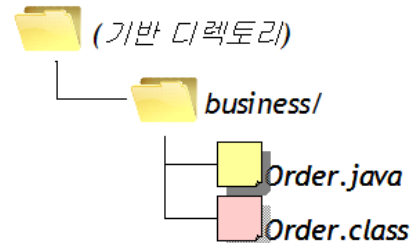


그림 13.3 소스 파일과 클래스 파일의 위치

© 2009 인피니티박스 All rights reserved



클래스 경로



Q: 우리가 실행시키고 싶은 클래스 파일이 다른 디렉토리에 있는 경우에도 실행시킬 수 있을까?

A: 실행이 가능하다. 다만 클래스 경로를 설정하여야 한다. 자바 가상 기계가 클래스 파일을 찾는 경로를 클래스 경로(class path)라고 한다.

- 환경 변수인 CLASSPATH를 설정한다.

```
C> set CLASSPATH=C:\classes;C:\lib;.
```

- 컴파일러나 자바 가상 기계를 실행할 때 옵션 -classpath를 사용한다.

```
C> java -classpath C:\classes;C:\lib;. business.Order
```

탐색 순서

- ① C:\classes\business\Order.class
- ② C:\lib\business\Order.class
- ③ .\business\Order.class

© 2009 인피니티박스 All rights reserved



중간 점검 문제

1. 패키지를 사용하는 이점은 무엇인가?
2. 패키지 이름을 짓는 일반적인 관례는 무엇인가?
3. 인터넷 도메인 이름이 `hu.ac.kr`이라면 권장되는 패키지 이름은?
4. 만약 패키지의 이름이 `company.project`와 같다면 소스 파일이 저장되는 디렉토리는?
5. 만약 `Server`라는 클래스를 `project.server` 패키지로 만들고 싶으면 어떤 문장을 `Server.java`에 추가하여야 하는가?
6. 소스 파일과 클래스 파일을 분리하려면 어떻게 하면 되는가?

© 2009 인피니티박스 All rights reserved



패키지를 사용하는 방법

1. 클래스에 패키지 이름을 붙여서 참조한다.
2. 개별 클래스를 `import`한다.
3. 전체 패키지를 `import`한다.

1. `business.Order myOrder = new business.Order();`
2. `import business.Order;` // business 패키지 안의 Order 클래스 포함
3. `import business.*;` // 패키지 전체 포함

© 2009 인피니티박스 All rights reserved



주의할점!!



Q: `java.awt.*` 문장은 `java.awt.font` 패키지를 포함하는가?

A: `java.awt.font` 패키지는 `java.awt` 패키지 안에 포함되지 않는다.
만약 `java.awt.font`의 멤버와 `java.awt`의 멤버를 동시에 사용하려면 다음과 같이 따로 따로 포함하여야 한다.

```
import java.awt.*;  
import java.awt.font.*;
```

© 2009 인피니티박스 All rights reserved



중간 점검 문제

1. `graphics` 패키지 안에 포함된 `Rectangle`이라는 클래스를 사용하기 위한 3가지의 방법을 말하라.
2. 패키지 이름은 어떻게 지어야 하는가??

© 2009 인피니티박스 All rights reserved



자바에서 지원하는 패키지

패키지	설명
java.applet	애플릿을 생성하는데 필요한 클래스
java.awt	그래픽과 이미지를 위한 클래스
java.beans	자바빈즈 구조에 기초한 컴포넌트를 개발하는데 필요한 클래스
java.io	입력과 출력 스트림을 위한 클래스
java.lang	자바 프로그래밍 언어에 필수적인 클래스
java.math	수학에 관련된 클래스
java.net	네트워킹 클래스
java.nio	새로운 네트워킹 클래스
java.rmi	원격 메소드 호출(RMI) 관련 클래스
java.security	보안 프레임워크를 위한 클래스와 인터페이스
java.sql	데이터베이스에 저장된 데이터를 접근하기 위한 클래스
java.util	날짜, 난수 생성기 등의 유틸리티 클래스
javax.imageio	자바 이미지 I/O API
javax.net	네트워킹 애플리케이션을 위한 클래스
javax.swing	스윙 컴포넌트를 위한 클래스
javax.xml	XML을 지원하는 패키지

© 2009 인피니티박스 All rights reserved



Java.lang 패키지

- Import 문을 사용할 필요가 없이 기본적으로 포함된다.
 - Object 클래스: 기초적인 메소드를 제공하는 모든 클래스의 조상 클래스
 - Math 클래스: 각종 수학 함수들을 포함하는 클래스
 - Wrapper 클래스: Integer와 같이 기초 자료형을 감싸서 제공하는 래퍼 클래스들
 - String 클래스, StringBuffer 클래스: 문자열을 다루는 클래스
 - System 클래스: 시스템 정보를 제공하거나 입출력을 제공하는 클래스
 - Thread 클래스: 스레드 기능을 제공하는 클래스
 - Class 클래스: 클래스에 대한 정보를 얻기 위한 클래스

© 2009 인피니티박스 All rights reserved



Math 클래스

필드	설명
static double E	자연 로그의 밑수(the base of the natural logarithms.)
static double PI	파이값

메소드	설명
static double abs (double a)	절대값
static double acos (double a)	arc cosine, 반환값의 범위 는 0.0에서 π .
static double asin (double a)	arc sine, 반환값의 범위 는 $-\pi/2$ 에서 $\pi/2$.
static double atan (double a)	arc tangent, 반환값의 범위 는 $-\pi/2$ 에서 $\pi/2$.
static double atan2 (double y, double x)	직교 좌표계 (x, y)를 극좌표계 (r, θ)로 변환할 때 θ 를 반환
static double cos (double a)	cosine
static double cosh (double x)	hyperbolic cosine
static double exp (double x)	e^x
static double hypot (double x, double y)	$\sqrt{x^2 + y^2}$
static double log (double a)	natural logarithm (base e)
static double log10 (double a)	base 10 logarithm

© 2009 인피니티박스 All rights reserved



Math 클래스

static double max (double a, double b)	큰 수
static double min (double a, double b)	작은 수
static double pow (double a, double b)	a^b
static double random()	0.0과 1.0사이의 난수를 반환
static double sin (double a)	sine
static double sinh (double x)	hyperbolic sine
static double sqrt (double a)	제곱근
static double tan (double a)	tangent
static double tanh (double x)	hyperbolic tangent
static double toDegrees (double angdeg)	라디안을 디그리로 변환
static double toRadians (double angdeg)	디그리를 라디안으로 변환

© 2009 인피니티박스 All rights reserved



예제

```
public class MathTest {  
    public static void main(String[] args){  
        double x = Math.PI;  
        System.out.println(Math.sin(x));  
        System.out.println(Math.random());  
    }  
}
```

실행 결과

```
1.2246467991473532E-16  
0.2454637294993175
```

© 2009 인피니티박스 All rights reserved



Class 클래스

- Class 객체는 실행 중인 클래스를 나타낸다.
- Class 객체는 자바 가상 기계에 의하여 자동적으로 생성된다.
- Class 객체를 이용하면 객체의 클래스 이름을 출력할 수 있다.

```
void printClassName(Object obj) {  
    System.out.println("The class of " + obj + " is " + obj.getClass().getName());  
}
```

© 2009 인피니티박스 All rights reserved



System 클래스

- System 클래스는 실행 시스템과 관련된 속성과 메소드를 제공

필드	설명
static PrintStream err	표준 오류 출력 스트림("standard" error output stream)
static InputStream in	표준 입력 스트림("standard" input stream)
static PrintStream out	표준 출력 스트림("standard" output stream)
메소드	설명
static void arraycopy (Object src, int srcPos, Object dest, int destPos, int length)	지정된 소스 배열을 목적지 배열로 복사한다.
static long currentTimeMillis()	밀리초 단위로 현재 시각을 반환한다.
static void exit (int status)	현재 실행중인 자바 가상 기계를 중단한다.
static String getenv (String name)	지정된 환경 변수의 값을 얻는다.
static String getProperty (String key)	키에 의해서 지정된 시스템의 특성을 얻는다.
static long nanoTime()	나노초 단위로 현재 시각을 반환한다.

© 2009 인피니티북스 All rights reserved



System 클래스

```

public class SystemTest {
    public static void main(String[] args) {
        System.out.println(System.currentTimeMillis());
        System.out.println(System.nanoTime());
        System.exit(0);
    }
}

```

실행 결과

```

1245152017843
101355767783335

```

© 2009 인피니티북스 All rights reserved



Wrapper 클래스

- 기초 자료형을 객체로 포장시켜주는 클래스
 - (예) Integer obj = **new** Integer(10);



래퍼 클래스는
기초 자료형을
객체로 포장한
것입니다.



기초 자료형	래퍼 클래스
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean
void	Void

© 2009 인피니티박스 All rights reserved



Integer 클래스가 제공하는 메소드

반환값	메소드 이름	설명
static int	intValue()	int형으로 반환한다.
static double	doubleValue()	double형으로 반환한다.
static float	floatValue()	float형으로 반환한다.
static int	parseInt(String s)	문자열을 int형으로 변환한다.
static String	toBinaryString(int i)	int형의 정수를 2진수 형태의 문자열로 변환한다.
static String	toHexString(int i)	int형의 정수를 16진수 형태의 문자열로 변환한다.
static String	toOctalString(int i)	int형의 정수를 8진수 형태의 문자열로 변환한다.
static String	toString(int i)	int형의 정수를 10진수 형태의 문자열로 변환한다.
static Integer	valueOf(String s)	문자열 s를 Integer 객체로 변환한다.
static Integer	valueOf(String s, in radix)	문자열 s를 radix진법의 Integer 객체로 변환한다.

© 2009 인피니티박스 All rights reserved



문자열 <-> 기초 자료형

기초 자료형 -> 문자열

```
String s1 = Integer.toString(10);  
String s2 = Integer.toString(10000);  
String s3 = Float.toString(3.14);  
String s4 = Double.toString(3.141592);
```

문자열 -> 기초 자료형

```
int i = Integer.parseInt("10");  
long l = Long.parseLong("10000");  
float f = Float.parseFloat("3.14");  
double d = Double.parseDouble("3.141592");
```

© 2009 인피니티박스 All rights reserved



오토 박싱(auto-boxing)

- Wrapper 객체와 기초 자료형 사이의 변환을 자동으로 수행한다.

```
Integer box = new Integer(10);  
System.out.println(box + 1);    // box는 자동으로 int형으로 변환
```

© 2009 인피니티박스 All rights reserved



StringBuffer 클래스

- String 클래스는 주로 상수 문자열, 즉 변경이 불가능한 문자열을 나타낸다.
- StringBuffer와 StringBuilder 클래스는 변경 가능한 문자열을 나타낸다.

```
StringBuffer sb = new StringBuffer(); // 기본적으로 16바이트의 공간이 할당된다.
```

```
sb.append("Hello"); // 6바이트가 사용된다.
```



StringBuffer의 메소드

메소드	설명
<code>StringBuilder append(String s)</code> <code>StringBuilder append(char[] str)</code> <code>StringBuilder append(char[] str, int offset, int len)</code> ...	인수는 먼저 문자열로 변환되어서 문자열에 붙여진다.
<code>StringBuilder delete(int start, int end)</code> <code>StringBuilder deleteCharAt(int index)</code>	지정된 문자를 삭제한다.
<code>StringBuilder insert(int offset, char[] str)</code> <code>StringBuilder insert(int index, char[] str, int offset, int len)</code> ...	첫번째 매개 변수는 데이터가 삽입될 위치의 바로 앞을 나타낸다. 두번째 매개 변수는 문자열로 변환된 후에 삽입된다.
<code>StringBuilder replace(int start, int end, String s)</code> <code>void setCharAt(int index, char c)</code>	지정된 위치의 문자를 변경한다.
<code>StringBuilder reverse()</code>	저장된 문자들의 순서를 역순으로 한다.



중간 점검 문제

1. 자바에서 정수 10과 Integer(10)은 어떻게 다른가?
2. StringBuffer와 String은 어떻게 다른가?

© 2009 인피니티북스 All rights reserved



java.util 패키지

- 여러 가지 유틸리티 클래스 들을 제공한다.

Class Summary	
AbstractCollection<E>	This class provides a skeletal implementation of the <code>Collection</code> interface, to minimize the effort required to implement this interface.
AbstractList<E>	This class provides a skeletal implementation of the <code>List</code> interface to minimize the effort required to implement this interface backed by a "random access" data store (such as an array).
AbstractMap<K,V>	This class provides a skeletal implementation of the <code>Map</code> interface, to minimize the effort required to implement this interface.
AbstractMap.SimpleEntry<K,V>	An Entry maintaining a key and a value.
AbstractMap.SimpleImmutableEntry<K,V>	An Entry maintaining an immutable key and value.
AbstractQueue<E>	This class provides skeletal implementations of some <code>Queue</code> operations.
AbstractSequentialList<E>	This class provides a skeletal implementation of the <code>List</code> interface to minimize the effort required to implement this interface backed by a "sequential access" data store (such as a linked list).
AbstractSet<E>	This class provides a skeletal implementation of the <code>Set</code> interface to minimize the effort required to implement this interface.
ArrayDeque<E>	Resizable-array implementation of the <code>Deque</code> interface.
ArrayList<E>	Resizable-array implementation of the <code>List</code> interface.
Arrays	This class contains various methods for manipulating arrays (such as sorting and searching).
BitSet	This class implements a vector of bits that grows as needed.
Calendar	The <code>Calendar</code> class is an abstract class that provides methods for converting between a specific instant in time and a set of <code>calendar fields</code> such as YEAR, MONTH, DAY_OF_MONTH, HOUR, and so on, and for manipulating the calendar fields, such as getting the date of the next week.
Collections	This class consists exclusively of static methods that operate on or return collections.
Currency	Represents a currency.
Date	The class <code>Date</code> represents a specific instant in time, with millisecond precision.
Dictionary<K,V>	The <code>Dictionary</code> class is the abstract parent of any class, such as <code>Hashtable</code> , which maps keys to values.

© 2009 인피니티북스 All rights reserved



Random 클래스



생성자
Random() 새로운 난수 발생기 객체를 생성한다.
Random(long seed) 주어진 시드를 사용하는 새로운 난수 발생기 객체를 생성한다.

메소드	설명
protected int next(int bits)	다음 난수를 발생한다.
boolean nextBoolean()	boolean 형의 난수를 발생한다.
void nextBytes(byte[] bytes)	byte 형의 난수를 발생하여서 주어진 배열을 채운다.
double nextDouble()	double 형의 0.0과 1.0 사이의 난수를 발생한다.
float nextFloat()	float 형의 0.0과 1.0 사이의 난수를 발생한다.
double nextGaussian()	평균이 0.0이고 표준 편차가 1.0인 가우시안 분포(정규 분포)에서 다음 난수를 발생한다.
int nextInt()	int 형의 난수를 발생한다.
int nextInt(int n)	0과 n 상의 int 형의 난수를 발생한다.
long nextLong()	long 형의 난수를 발생한다.
void setSeed(long seed)	시드를 설정한다.

© 2009 인피니티박스 All rights reserved



예제

```
import java.util.Random;

public class RandomTest {
    public static void main(String[] args)
    {
        Random random = new Random();
        for (int i = 0; i < 10; i++)
            System.out.println(random.nextInt(100));
    }
}
```

실행 결과

```
79
60
98
...
96
```

© 2009 인피니티박스 All rights reserved



Arrays 클래스

메소드	설명
<code>static List asList(Object[] a)</code>	주어진 배열을 고정 길이의 리스트로 변환한다.
<code>static int binarySearch(int[] a, int key)</code>	주어진 값을 <code>int</code> 형의 배열에서 이진 탐색한다.
<code>static int binarySearch(Object[] a, Object key)</code>	<code>Object</code> 타입의 배열에서 <code>key</code> 를 이진 탐색하는 메소드
<code>static int[] copyOf(int[] original, int newLength)</code>	주어진 배열을 새로운 크기의 배열에 복사한다.
<code>static int[] copyOfRange(int[] original, int from, int to)</code>	배열에서 주어진 구간의 값들을 새로운 배열로 복사한다.
<code>static boolean equals(int[] a, int[] a2)</code>	주어진 두개의 배열이 같으면 <code>true</code> 를 반환한다.
<code>static void fill(int[] a, int val)</code>	주어진 <code>val</code> 값을 가지고 배열을 채운다.
<code>static void sort(int[] a)</code>	지정된 <code>int</code> 형의 배열을 정렬한다.

© 2009 인피니티박스 All rights reserved



예제

```

import java.util.Arrays;
public class ArraysTest {
    public static void main(String[] args) {
        int[] array = { 9, 4, 5, 6, 2, 1 };
        Arrays.sort(array);
        printArray(array);
        System.out.println(Arrays.binarySearch(array, 9));
        Arrays.fill(array, 8);
        printArray(array);
    }
    private static void printArray(int[] array) {
        System.out.print("[");
        for(int i=0 ;i< array.length;i++)
            System.out.print(array[i]+" ");
        System.out.println("]");
    }
}

```

실행 결과

```

[1 2 4 5 6 9 ]
5
[8 8 8 8 8 8 ]

```

© 2009 인피니티박스 All rights reserved



Date 클래스

```
import java.util.*;
```

```
public class DateTest {
    public static void main(String[] args) {
        Date d = new Date();
        System.out.println(d);
        System.out.println(1900+d.getYear());
        System.out.println(d.getMonth()+1);
        System.out.println(d.getDate());

        d.setHours(12);
        d.setMinutes(34);
        d.setSeconds(0);
        System.out.println(d);
    }
}
```

실행 결과

```
Mon Jun 15 11:19:31 KST 2009
2009
6
15
Mon Jun 15 12:34:00 KST 2009
```

© 2009 인피니티북스 All rights reserved



Calendar 클래스

메소드	설명
abstract void add (int field, int amount)	지정된 필드에 시간을 더하거나 뺀다.
boolean after (Object when)	현재의 객체가 주어진 시간보다 뒤이면 true 를 반환한다.
boolean before (Object when)	현재의 객체가 주어진 시간보다 앞이면 true 를 반환한다.
void clear (int field)	지정된 필드를 정의되지 않은 상태로 변경한다.
int compareTo (Calendar anotherCalendar)	두개의 Calendar 객체를 비교한다.
boolean equals (Object obj)	두개의 Calendar 객체가 같으면 true 를 반환한다.
int get (int field)	주어진 필드의 값을 반환한다.
static Calendar getInstance ()	현재 시각을 나타내는 Calendar 객체를 반환한다.
Date getTime ()	Calendar 객체를 Date 객체로 반환한다.
void set (int year, int month, int date, int hourOfDay, int minute, int second)	각 필드의 값을 설정한다.
void setTime (Date date)	Date 객체의 값으로 Calendar 객체를 설정한다.

© 2009 인피니티북스 All rights reserved



예제

```
import java.util.*;

public class CalendarTest {
    public static void main(String[] args) {
        Calendar d = Calendar.getInstance();
        System.out.println(d);
        System.out.println(d.get(Calendar.YEAR)+"년");
        System.out.println(d.get(Calendar.MONTH)+1 + "월");
        System.out.println(d.get(Calendar.DATE)+"일");

        d.set(Calendar.HOUR, 12);
        d.set(Calendar.MINUTE, 34);
        d.set(Calendar.SECOND, 0);
        System.out.println(d);
    }
}
```

실행 결과

```
java.util.GregorianCalendar[... ,YEAR=2009,MONTH=5,WEEK_OF_YEAR=25,...]
2009년
6월
15일
java.util.GregorianCalendar[... ,YEAR=2009,MONTH=5,WEEK_OF_YEAR=25,...]
```



StringTokenizer 클래스

- 문자열을 분석하여서 토큰으로 분리시켜 주는 기능을 제공

생성자
StringTokenizer(String str) 주어진 문자열을 위한 StringTokenizer 객체를 생성한다.
StringTokenizer(String str, String delim) 주어진 문자열을 위한 StringTokenizer 객체를 생성한다. 분리자로 delim을 사용한다.
StringTokenizer(String str, String delim, boolean returnDelims) 주어진 문자열을 위한 StringTokenizer 객체를 생성한다. 분리자로 delim을 사용한다. returnDelims이 true이면 분리자를 포함하여 반환한다.

메소드	설명
int countTokens()	문자열에 존재하는 토큰의 개수를 반환한다.
boolean hasMoreTokens()	다음 토큰을 가지는지를 반환한다.
String nextToken()	다음 토큰을 반환한다.
String nextToken(String delim)	다음 토큰을 반환하고 분리자를 delim으로 변경



예제

```
import java.util.*;

public class StringTest {
    public static void main(String[] args) {
        StringTokenizer st = new StringTokenizer("Will Java change my life?");
        while (st.hasMoreTokens()) {
            System.out.println(st.nextToken());
        }
    }
}
```

실행 결과

Will
Java
change
my
life?

© 2009 인피니티박스 All rights reserved



중간 점검 문제

- 1에서 1000사이의 난수를 발생하기 위한 문장을 작성하라.
- 현재 날짜를 얻기 위한 문장을 작성하라.

© 2009 인피니티박스 All rights reserved



Q & A

