



Power Java

제4장 자바 프로그래밍 기초



© 2009 인피니티북스 All rights reserved



이번 장에서 학습할 내용

- 자바 프로그램에 대한 기초 사항을 학습
- 자세한 내용들은 추후에....

© 2009 인피니티북스 All rights reserved



HelloWorld.java 프로그램

HelloWorld.java

```
/**
 * 표준 출력으로 "Hello World!"를 표시하는 간단한 자바 애플리케이션의 구현이다.
 */
public class HelloWorld {
    public static void main(String args[]) {
        System.out.println("Hello World!");    // 문자열 출력
    }
}
```

실행결과

Hello World!

© 2009 인피니티박스 All rights reserved



주석

- 주석(comment): 프로그램에 대한 설명을 적어 넣은 것

```
/**
 * 표준 출력으로 "Hello World!"를 표시하는 간단한 자바 애플리케이션의 구현이다.
 */
public class Hello {
    public static void main(String args[]) {
        System.out.println("Hello World!");    // 문자열 출력
    }
}
```

주석(comment)

© 2009 인피니티박스 All rights reserved



3가지 타입의 주석

```
/* text */
```

/* 에서 */ 까지가 주석으로서 컴파일 시에 무시된다.

```
/** documentation */
```

/** 에서 */ 까지가 주석으로서 컴파일 시에 무시되는 것은 같으나 이것은 선언문 앞에만 사용할 수 있고 JDK에 포함된 javadoc 프로그램이 /**에서 */까지의 내용을 가지고 주석을 추출하여 자동적으로 소스를 설명하는 HTML 문서를 만들어준다.

```
// text
```

//에서 줄의 끝까지가 무시된다.

© 2009 인피니티박스 All rights reserved

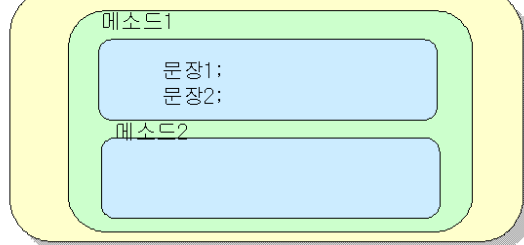


클래스

- 클래스(class): 객체를 만드는 설계도(추후에 학습)
- 자바 프로그램은 클래스들로 구성된다.

수스 파일

클래스



Hello.java

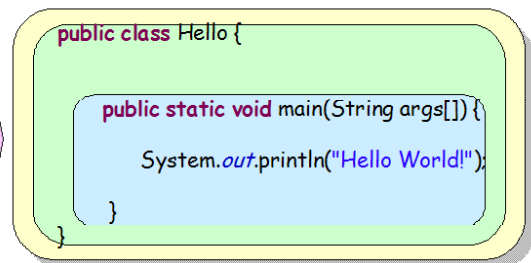


그림 4.2 자바 프로그램의 구조

© 2009 인피니티박스 All rights reserved



클래스 정의

```

/**
 * 표준 출력으로 "Hello World!"를 표시하는 간단한 자바 애플리케이션의 구현이다.
 */
public class Hello {
    public static void main(String args[]) {
        System.out.println("Hello World!"); // 문자열 출력
    }
}

```

클래스 정의

- `public` 키워드는 Hello 클래스가 다른 클래스에서도 사용가능함을 나타낸다.

© 2009 인피니티박스 All rights reserved



메소드

- 클래스 = 필드(변수) + 메소드(함수)
- 메소드(method)는 메소드는 입력을 받아서 작업을 수행하고 결과를 내보내는 작은 기계
- 메소드는 안에 들어 있는 문장들을 차례대로 실행한 후에 작업의 결과를 외부로 반환

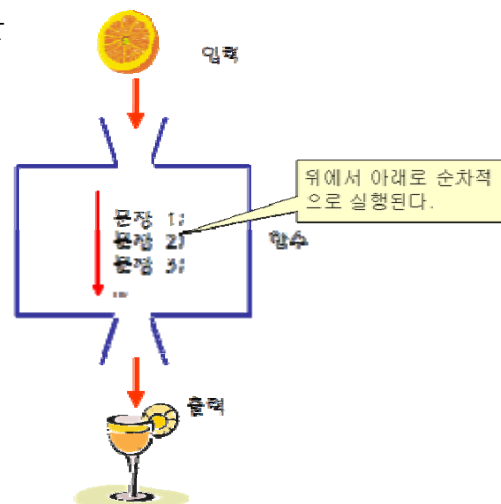


그림 4.3 메소드에는 문장들이 들어있고 이들 문장들은 위에서 아래로 차례대로 실행된다.

© 2009 인피니티박스 All rights reserved



메소드의 정의

```
/**
 * 표준 출력으로 "Hello World!"를 표시하는 간단한 자바 애플리케이션의 구현이다.
 */
public class Hello {
    public static void main(String args[]) {
        System.out.println("Hello World!"); // 문자열 출력
    }
}
```

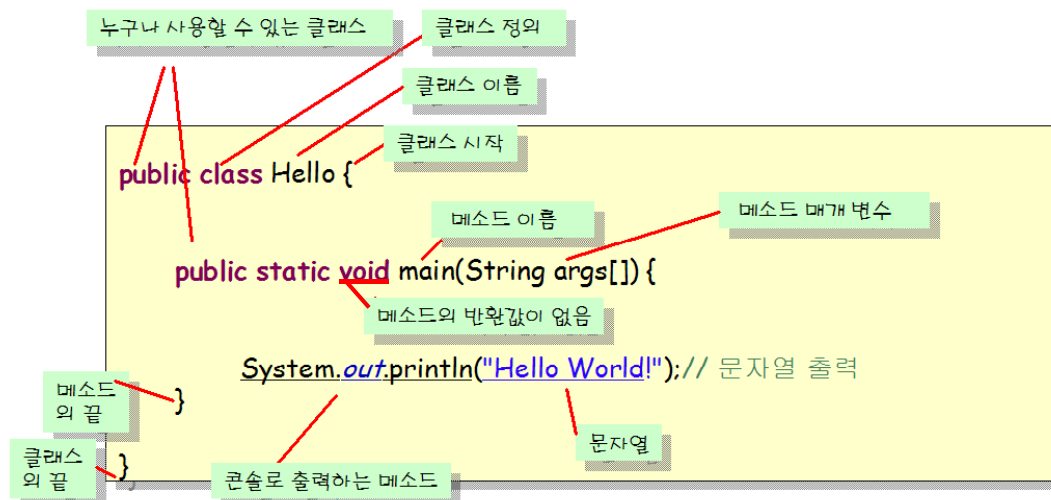
메소드 정의

- public: 누구나 이용할 수 있음
- static: 정적 메소드(9장 부근에서 학습)
- void: 반환값이 없음
- Main: 메소드 이름
- String args[]: 매개 변수(메소드가 외부에서 받는 데이터)

© 2009 인피니티박스 All rights reserved



용어 설명



© 2009 인피니티박스 All rights reserved



문장

- 문장(statement)은 사용자가 컴퓨터에게 작업을 지시하는 단위이다.
- 문장들은 메소드 안에 들어 있다.
- 보통 프로그램의 한 줄이 하나의 문장이 된다.
- 문장의 끝은 항상 세미콜론(;)으로 끝나게 된다.

```
... main(...)
{
    문장;
    문장;
    문장;
    ....
}
```

© 2009 인피니티박스 All rights reserved



문장은 순차적으로 실행

Welcome.java

// 문장의 순차적인 실행 예제

```
public class Welcome {
    public static void main(String args[]) {
        System.out.println("Welcome to");
        System.out.println("Java");
    }
}
```

문장들을 순차적으로 실행된다.

실행결과

```
Welcome to
Java
```

© 2009 인피니티박스 All rights reserved



중간 점검 문제

1. 자바 프로그램의 개발 단위는 _____이다.
2. _____은 프로그램에 대한 설명이다.
3. 입력을 받아서 작업을 수행하고 결과를 내보내는 작은 기계로 생각할 수 있는 것은 _____이다.
4. 모든 자바 소스 파일의 확장자는 _____이다.

© 2009 인피니티박스 All rights reserved



Add 예제

Hello.java

```
/**
 * 두수의 합을 계산하는 애플리케이션
 */
public class Add {
    public static void main(String args[]) {
        int x; // 첫번째 정수를 저장할 변수
        int y; // 두번째 정수를 저장할 변수
        int sum; // 두 정수의 합을 저장하는 변수

        x = 100;
        y = 200;
        sum = x + y;

        System.out.println(sum);
        return;
    }
}
```

실행결과

300

© 2009 인피니티박스 All rights reserved



변수

- 변수(variable)란 프로그램이 사용하는 데이터를 일시적으로 저장할 목적으로 사용하는 메모리 공간

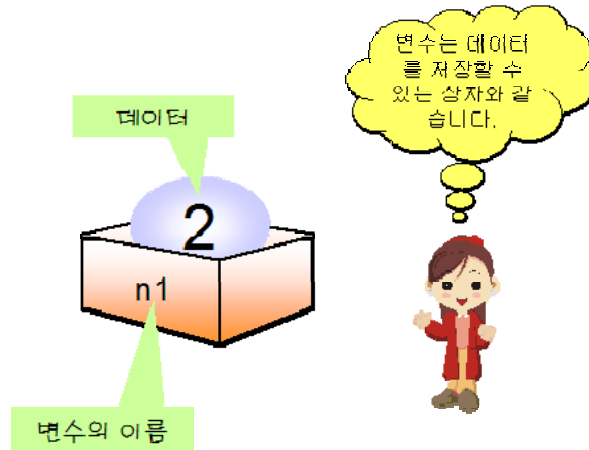


그림 4.9 변수와 상자

© 2009 인피니티박스 All rights reserved



변수에도 여러 가지 타입이 있다.

- 변수는 타입과 이름을 가지고 있다.

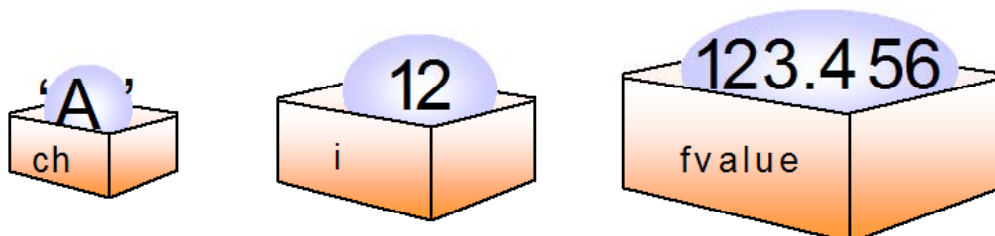


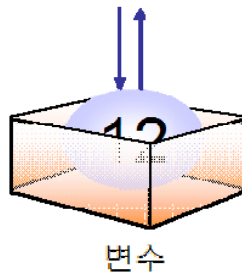
그림 4.11 변수들은 자신만의 이름을 가지고 있다.

© 2009 인피니티박스 All rights reserved

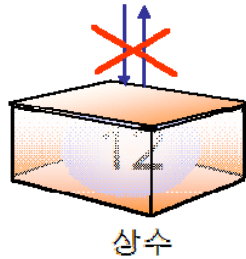


상수

- 상수(constant)란 그 값이 프로그램이 실행하는 동안 변하지 않는 수
- (예) 3.14, 1000, "Hello World!"



변수



상수

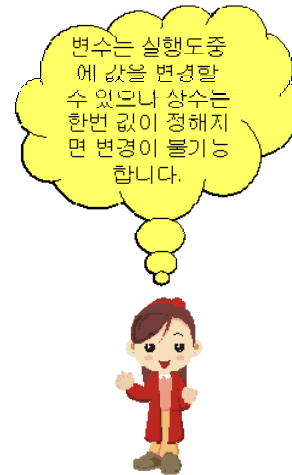


그림 4.12 변수와 상수



변수의 초기화

```
int x = 0; // 첫번째 정수를 저장하는 변수
int y = 0; // 두번째 정수를 저장하는 변수
int sum = 0; // 두 정수의 합을 저장하는 변수
```

변수는 위와 같이 = 연산자를 이용하여 초기화할 수 있다.



수식

- 수식(expression): 피연산자와 연산자로 이루어진다.
- 수식의 결과값을 가진다.

```
x = 100;  
y = 200;  
sum = x + y;
```

© 2009 인피니티박스 All rights reserved



대입 연산자

- 대입 연산자 == 할당 연산자 == 배정 연산자
- 변수에 값을 저장하는 연산자
- (예) $x = 100;$

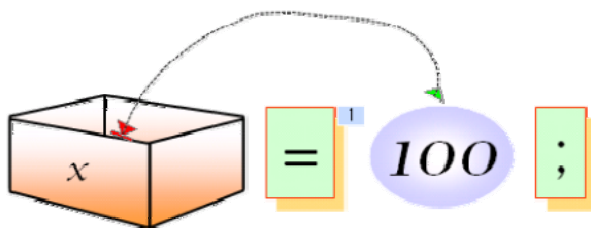


그림 4.12 = 연산자가 변수에 값을 저장하는 연산자이다.

© 2009 인피니티박스 All rights reserved



산술 연산자

연산	연산자	자바 수식	수학에서의 기호
덧셈	+	$x + y$	$x + y$
뺄셈	-	$x - y$	$x - y$
곱셈	*	$x * y$	xy
나눗셈	/	x / y	x/y 또는 $\frac{x}{y}$ 또는 $x \div y$
나머지	%	$x \% y$	$x \bmod y$

표 3.1 산술 연산자의 요약



산술 연산의 과정

- (예) `sum = x + y;`

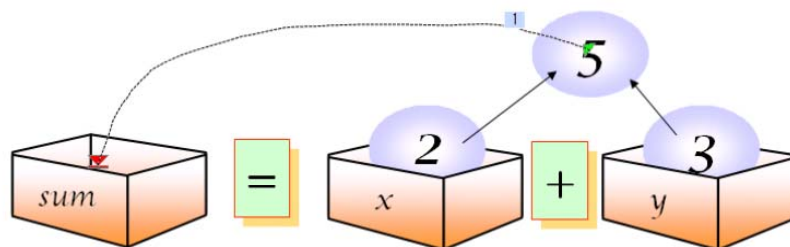


그림 3.12 산술 연산의 과정: 먼저 `x`과 `y`에서 값을 가져와서 덧셈연산이 수행되고 그 결과값이 `sum`에 저장된다.



중간 점검 문제

1. 프로그램에서 데이터를 저장하는 공간은 _____이다.
2. 변수에 값을 저장하는 연산을 _____이라고 한다.
3. 실행 도중에 값이 변하지 않는 수를 _____이라고 한다.

© 2009 인피니티박스 All rights reserved



Add2 예제

- 사용자로부터 두 개의 정수를 받아서 더하는 문제

실행결과

첫번째 숫자를 입력하시오: 10

두번째 숫자를 입력하시오: 20

30

- 사용자로부터 숫자를 받을 수 있어야 한다.

© 2009 인피니티박스 All rights reserved



Add2.java

// 사용자가 입력한 두 개의 숫자를 더해서 출력한다.

import java.util.Scanner; // Scanner 클래스 포함

public class Add2 {

// 메인 메소드에서부터 실행이 시작된다.

public static void main(String args[]) {

// 사용자로부터 입력을 받기 위해 Scanner를 생성한다.

Scanner input = **new** Scanner(System.in);

int x; // 첫 번째 숫자 저장

int y; // 두 번째 숫자 저장

int sum; // 합을 저장

System.out.print("첫 번째 숫자를 입력하십시오: "); // 입력 안내 출력

x = input.nextInt(); // 사용자로부터 첫 번째 숫자를 읽는다.

System.out.print("두 번째 숫자를 입력하십시오: "); // 입력 안내 출력

y = input.nextInt(); // 사용자로부터 두 번째 숫자를 읽는다.

sum = x + y; // 두 개의 숫자를 더한다.

System.out.println(sum); // 합을 출력한다.

} // 메인 메소드의 끝

} // Add 클래스의 끝



import 문장

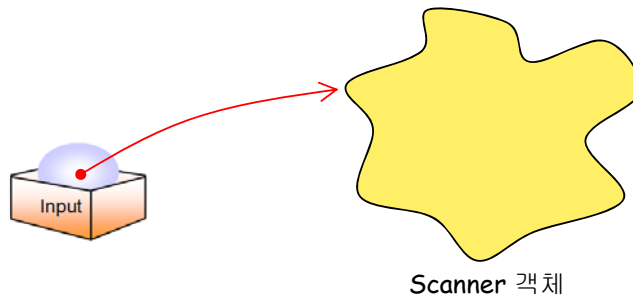
- **import** java.util.Scanner; // Scanner 클래스 포함
- Scanner 클래스를 포함시키는 문장
- Scanner는 자바 클래스 라이브러리(Java Class Library)의 일종
- Scanner는 입력을 받을 때 사용



객체 생성

```
Scanner input = new Scanner(System.in);
```

- input은 타입이 Scanner인 변수
- new Scanner(System.in)은 Scanner 클래스의 객체(object)를 생성
- input은 생성된 객체를 가리킨다.
- 상세한 설명은 차후에....
- 일단 입력을 받으려면 이 문장이 필요하다고 알아두자.



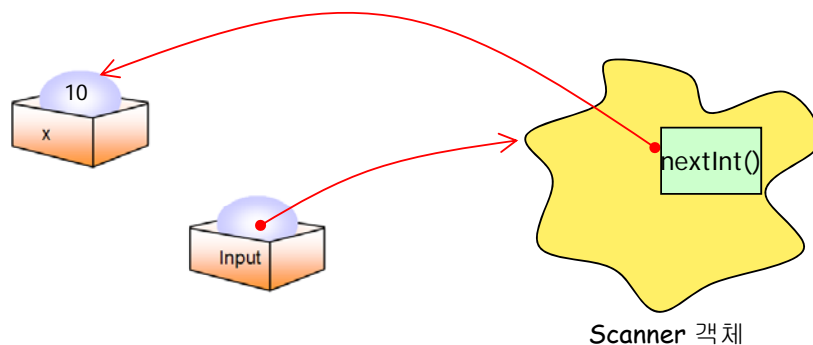
© 2009 인피니티박스 All rights reserved



사용자로부터 입력

```
x = input.nextInt(); // 사용자로부터 첫 번째 정수를 읽는다.
```

- Scanner 객체인 input을 이용하여 사용자로부터 정수를 읽어 들이는 문장
- input을 통하여 nextInt()라고 하는 메소드를 호출하게 된다.
- nextInt()에서 반환된 값은 변수 x에 대입된다.
- 추후에 자세히....



© 2009 인피니티박스 All rights reserved



print() 출력 메소드

`System.out.print("첫번째 숫자를 입력하시오: "); // 입력 안내 출력`

- println()은 문자열을 출력한 후에 줄을 바꾸는 메소드
- print()는 문자열 출력 후에 줄을 바꾸지 않는 메소드



© 2009 인피니티박스 All rights reserved



중간 점검 문제

1. 자바 프로그램의 개발 단위는 _____이다.
2. _____은 프로그램에 대한 설명이다.
3. 입력을 받아서 작업을 수행하고 결과를 내보내는 작은 기계로 생각할 수 있는 것은 _____이다.
4. 모든 자바 소스 파일의 확장자는 _____이다.

© 2009 인피니티박스 All rights reserved



연봉 계산 문제

- 10년 동안 월급을 모두 저금할 경우, 얼마나 모을 수 있을까?



실행결과

월급을 입력하시오: 200
10년 동안의 저축액: 24000

© 2009 인피니티박스 All rights reserved

Salary.java

// 저축액을 계산하는 프로그램

import java.util.Scanner; // 입력 보조 클래스

public class Salary {
 public static void main(String args[]) {

int salary; // 월급

int deposit; // 저축액

 Scanner input = **new** Scanner(System.in);

 System.out.print("월급을 입력하시오: "); // 입력 안내 출력

 salary = input.nextInt();

 deposit = 10 * 12 * salary;

 System.out.printf("10년 동안의 저축액: %d\n", deposit);

 } // end method main

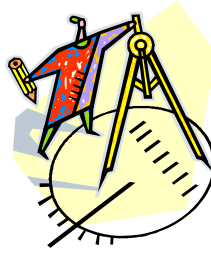
} // end class Salary

© 2009 인피니티박스 All rights reserved



원의 면적 구하기

- 사용자로부터 원의 반지름을 입력받고 이 원의 면적을 구한 다음, 화면에 출력한다.



실행결과

반지름을 입력하시오: 5
78.5

© 2009 인피니티박스 All rights reserved



원의 면적 구하기

CircleArea.java

```
import java.util.Scanner; // 프로그램은 스캐너 클래스를 사용한다.

public class CircleArea {
    public static void main(String args[]) {

        double radius; // 원의 반지름
        double area; // 원의 면적
        Scanner input = new Scanner(System.in);
        System.out.print("반지름을 입력하시오: "); // 입력 안내 출력
        radius = input.nextDouble();
        area = 3.14 * radius * radius;

        System.out.println(area);
    } // end method main
} // end class CircleArea
```

© 2009 인피니티박스 All rights reserved



중간 점검 문제

1. 사용자가 입력하는 원화를 달러화로 계산하는 프로그램은 작성하여 보자. 달러에 대한 원화 환율(예를 들어 1달러는 1392.83원)은 실수형 변수에 저장하고 사용자로부터는 받는 원화는 정수 변수에 저장한다. 원화를 환율로 나누어서 사용자가 입력한 원화가 몇 달러에 해당하는지를 계산한다.



오류

- 컴파일 오류(compile-error)
- 실행 오류(run-time error)
- 논리 오류(logical error)





컴파일 오류

- 컴파일 오류는 컴파일시에 발견되는 오류
- 문법 규칙에 따르지 않으면 구문 오류(syntax error)
- 계산식에서 호환이 되지 않는 서로 다른 타입의 데이터가 포함된 계산을 시도하는 경우에는 의미 오류(semantic error)

© 2009 인피니티박스 All rights reserved



실행 오류

- 실행 오류는 프로그램이 실행되는 도중에 발생하는 오류
 - (예) 0으로 나누려고 시도
- 자바에서는 많은 실행 오류들을 예외(exception)를 사용하여 처리



© 2009 인피니티박스 All rights reserved



논리 오류

- 논리 오류는 컴파일되고 실행도 되지만 의도하지 않는 결과를 발생하는 경우
- (예) 값이 예상과는 다르게 잘못 계산되는 경우
- 디버깅(debugging): 논리 오류를 찾고 고치는 과정



© 2009 인피니티북스 All rights reserved



중간 점검 문제

1. 컴파일시에 오류가 없는데도 원하는 결과가 나오지 않는다면 _____오류가 있는 것이다.
2. 만약 실행 도중에 0으로 나누었다면 _____오류가 발생한다.

© 2009 인피니티북스 All rights reserved



Q & A

