



그림으로 설명하고 실생활 예제로 실습하는

자버를 부탁해

학습 로드맵

- 01 자바 시작하기
- 02 변수, 타입, 연산자

② 메소드와 조건문

03 메소드

04 조건문

Part 1 기본 문법 Part 2 객체 문법

- 4 OOP 기초
- 07 객체 지향 프로그래밍
- 08 객체 지향 프로그래밍의 활용
- 09 자바 API

- **6** OOP 심화
- 10 상속
- 11 인터페이스
- 12 예외 처리와 파일 입출력

- ❸ 반복문과 배열
- 05 반복문

06 배열

- 6 프로젝트
- 13 실전 프로젝트: 영화 예매 관리 프로그램

CHO3 CHACE

01 메소드 개요

02 메소드 정의와 호출

03 메소드의 유형

메소드의 개념을 학습한다.

메소드의 사용법을 익힌다.

메소드의 실행 흐름을 이해한다.

CHO3 CHACE

01 메소드 개요

02 메소드 정의와 호출

03 메소드의 유형

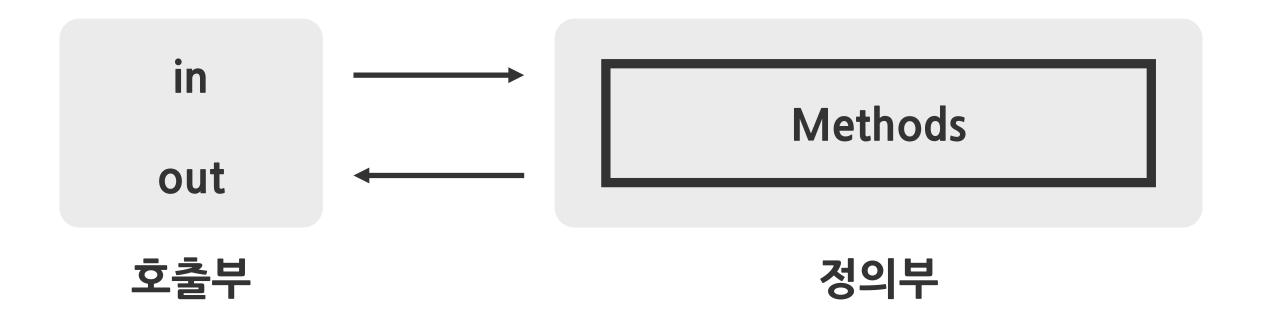
I. 메소드란

메소드(method)란 입력값에 따른 결과를 반환하는 일련의 코드 묶음 간단한 호출만으로도 여러 줄의 코드를 수행할 수 있음

Ⅱ. 메소드의 동작 원리

메소드는 호출부와 정의부로 나뉘어 동작

- 호출부: 메소드를 사용하는 영역, 입력 값을 전달하여 그 결과를 돌려받는 코드
- 정의부: 입력값을 받아 결과를 만드는 일련의 코드 묶음

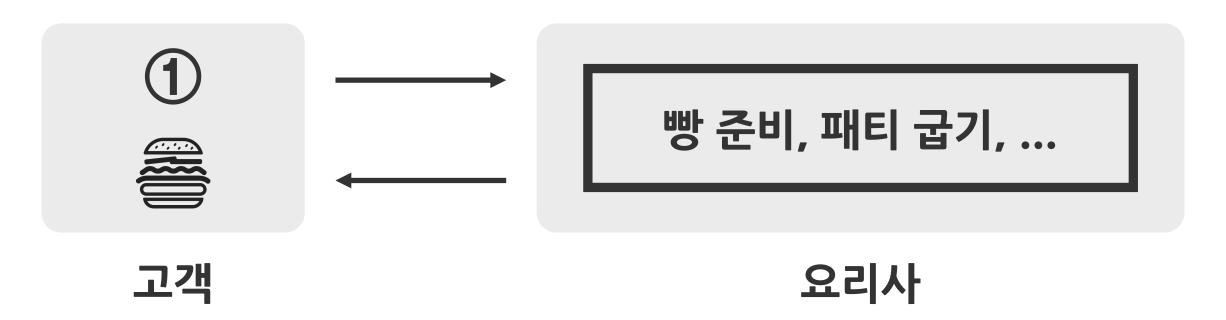


6/44

Ⅱ. 메소드의 동작 원리

햄버거 주문의 예

- 고객이 1번 메뉴를 주문하면 요리사는 이를 만들어 고객에게 전달
- 메뉴에 따른 조리법이 이미 준비되어 있기에 고객은 편리한 주문이, 요리사는 신속한 조리가 가능
- 메소드에 대입하면 고객이 호출부, 요리사는 정의부
- 메소드를 만드는 것을 **정의**(define)라 하고, 이를 사용하는 것을 **호출**(call) 이라 부름



CHO3 CHACE

01 메소드 개요

02 메소드 정의와 호출

03 메소드의 유형

1. 메소드의 구조

- 1 메소드명(method name): 메소드의 이름으로, 호출 시 사용
- ② 입력변수(parameters): 메소드 실행 시 입력값을 저장하는 변수, 매개변수 혹은 파라미터라고도 부름
- 3 반환값(return value): 메소드의 최종 결과로, 반환될 값을 의미
- 4 반환 타입(return type): 반환값의 타입을 의미

Ⅱ. 메소드 정의하기

정수를 입력받아 그 제곱값을 반환하는 메소드 만들기

- ① 메소드명 작성
- ②입력변수생성
- ③ 반환값 생성
- ④ 반환 타입 명시

```
public static int square (int n) {
  return n * n; 반환 타입
}

반환 타입을 명시
(반환 타입의 종류는 String, int, double, boolean, void 등이 있음)
```

Ⅲ. 메소드 호출하기

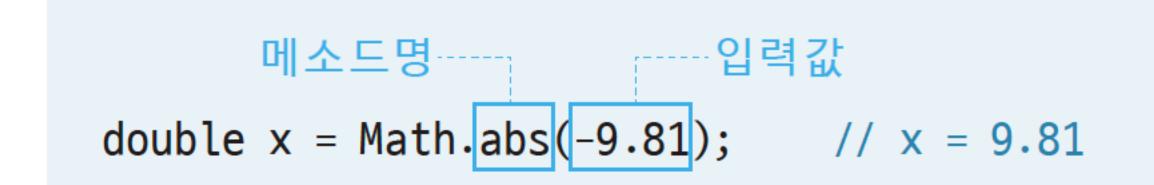
메소드 호출은 메소드명을 적고, 소괄호 사이에 입력값을 전달하여 사용

```
메소드명(입력값1, 입력값2, ・・・)
```

square() 메소드의 호출 예

Ⅲ. 메소드 호출하기

Math.abs() 메소드 호출 예



입력값 -9.81의 절댓값 9.81이 반환되어, 그 결과가 x에 대입됨

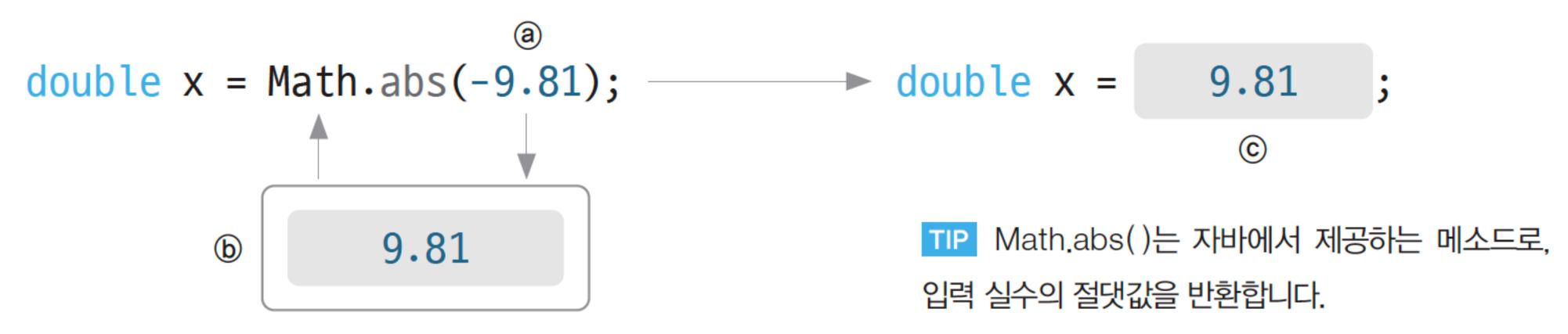


그림 3-3 Math.abs() 메소드의 실행 흐름

Ⅲ. 메소드 호출하기

Math.max() 메소드 호출 예



그림 3-4 Math.max() 메소드의 실행 흐름

TIP Math.max()는 자바에서 제공하는 메소드로, 입력된 두 수 중 더 큰 값을 반환합니다.

Quiz

01 매개변수와 파라미터는 ___의 또다른 이름이다.

02 다음 코드의 출력 결과는?

```
System.out.println(Math.abs(3.14));
System.out.println(Math.max(10, 100));
System.out.println(Math.min(10, 100));
```

CHO3 CHACE

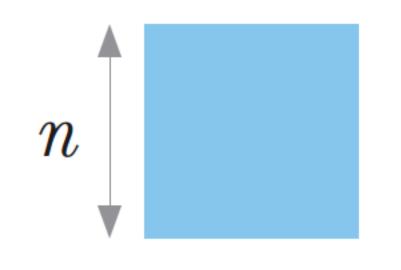
01 메소드 개요

02 메소드 정의와 호출

03 메소드의 유형

1. 단일 입력변수 메소드

한 변의 길이가 n인 정사각형의 넓이 S를 구하는 프로그램 square() 메소드를 사용해 정사각형의 넓이를 계산



$$S = n^2$$

S: 정사각형의 넓이

n: 정사각형의 한 변의 길이

그림 3-5 정사각형의 넓이 공식

```
실습 3-1
         단일 입력변수 메소드: 정사각형의 넓이
                                              ch03/ex3_1/Square.java
01 public class Square {
     public static void main(String[] args) {
02
03
        int n = 4;
        04
        System.out.printf("한 변의 길이가 ‰인 정사각형의 넓이: ‰", n, s);
05
06
07
     // 정수를 입력받아 그 제곱을 반환하는 메소드
80
     public static int square(int length) {------메소드 정의부
09
10
        return length * length;
11
12 }
```

한 변의 길이가 4인 정사각형의 넓이: 16

🗕 실습 코드

https://github.com/cloudstudyingkr/JavaPlz/blob/master/src/part1/ch03/e x3_1/Square_iava

```
실습 3-1
          단일 입력변수 메소드: 정사각형의 넓이
                                                       ch03/ex3_1/Square.java
01 public class Square {
02
      public static void main(String[] args) {
                                 03행의 수행으로 지역변수 n의 값이 4가 됨
03
          int n = 4;
                                 여기서 지역변수(local variable)란 메소드 내부에 선언
04
          int s = |square(n);
                                 된 변수로, 해당 메소드 안에서만 사용할 수 있음
05
          System.out.printf("한 변역
                                  main()
06
07
                                    int n 4
80
      public static int square(int
09
                                  그림 3-6 지역변수 n의 생성
10
          return length * length;
                                 입력변수 또한 선언된 메소드 내부에서만 사용할 수 있
11
                                 <u>는데, 이러한 변수의 활동영역을 스코프(scope)라 할</u>
12 }
```

한 변의 길이가 4인 정사각형의 넓이: 16

실습 3-1 단일 입력변수 메소드: 정사각형의 넓이 ch03/ex3_1/Square.java 01 public class Square { public static void main(String[] args) { 02 03 int n = 4; --- 04행에서 수행으로 대입 연산자 우측의 square(n) 메소드가 먼저 호출 04 int s = square(n); System.out.printf("한 변의 길이가 %d인 집<mark>설행 흐름은 09행의 square() 메소트로 넘여가</mark> 05 ,입력변수 length에 4가 대입 06 07 main() // 정수를 입력받아 그 제곱을 반환하는 메소드 80 int n 4 public static int square(int length) { 09 10 return length * length; square(int length 4) 11 12 } 실행 결과 그림 3-7 square() 메소드 호출과 입력값의 전달 한 변의 길이가 4인 정사각형의 넓이: 16 |메소드 호출 시 입력값은 입력변수로 대입되므 로 타입 불일치를 주의해야 한

19/44

```
실습 3-1
           단일 입력변수 메소드: 정사각형의 넓이
                                                        ch03/ex3_1/Square.java
01 public class Square {
                                                 다시 실행 흐름은 메인 메소드의 04행으로
      public static void main(String[] args) {
02
                                                 돌아와, 04행의 수행을 마무리
03
          int n = 4;
04
          int s = square(n);
                                                 지역변수 s에 반환값 16이 대입
                                                 main()
05
          System.out.printf("한 변의 길이가 %d인 정
06
                                                          int s 16 ◀
                                                  int n 4
07
      // 정수를 입력받아 그 제곱을 반환하는 메소드
80
                                                 square(int length 4)
      public static int square(int length) {
09
                                                  return length * length
10
          return length * length;
11
                                                 그림 3-9 square() 메소드의 반환값 전달과 대입
12 }
```

한 변의 길이가 4인 정사각형의 넓이: 16

```
실습 3-1
         단일 입력변수 메소드: 정사각형의 넓이
                                                 ch03/ex3_1/Square.java
01 public class Square {
      public static void main(String[] args) {
02
03
         int n = 4;
                                int s = square(n);
04
         System.out.printf("한 변의 길이가 %d인 정사각형의 넓이: %d", n, s);
05
06
                                           끝으로 05행의 형식 문자열을 출력함으로
07
                                           써 프로그램이 종료
     // 정수를 입력받아 그 제곱을 반환하는 메소드
80
     public static int square(int length) {------메소드 정의부
09
10
         return length * length;
11
12 }
```

한 변의 길이가 4인 정사각형의 넓이: 16

Ⅱ. 다중 입력변수 메소드

반지름의 길이가 r, 높이가 h인 원기둥의 부피를 구하는 프로그램 volume() 메소드를 사용해 부피를 계산

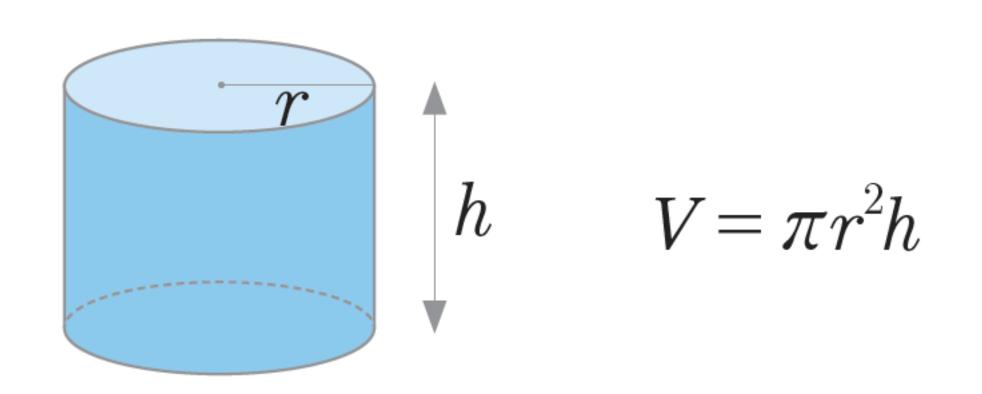


그림 3-10 원기둥의 부피 공식

V: 원기둥의 부피

r: 원기둥의 반지름

h: 원기둥의 높이

 π : 원주율(3.141592···)

```
실습 3-2
          다중 입력변수 메소드: 원기둥의 부피
                                                      ch03/ex3_2/Cylinder.java
01 public class Cylinder {
      public static void main(String[] args) {
02
03
          double r = 7;
          double h = 5;
04
          double v = volume(r, h); -----메소드 호출부
05
06
07
          System.out.printf("반지름이 %.1f, 높이가 %.1f인 원기둥의 부피: %.1f",
                            r, h, v);
08
09
      // 반지름의 길이와 높이를 입력받아, 원기둥의 부피를 반환하는 메소드
10
       public static double volume(double radius, double height) {
11
          return Math.PI * radius * radius * height;
12
13
14 }
                                              ·-----메소드 정의부
                                               🗕 실습 코드
 실행 결과
                                               https://github.com/cloudstudying-
                                               kr/JavaPlz/blob/master/src/part1/ch03/
반지름이 7.0, 높이가 5.0인 원기둥의 부피: 769.7
                                               ex3 2/Cylinder java
```

```
다중 입력변수 메소드: 원기둥의 부피
실습 3-2
                                                   ch03/ex3_2/Cylinder.java
01 public class Cylinder {
      public static void main(String[] args) {
02
03
         double r = 7;
                                   03행의 수행으로 메인 메소드 영역에 지역변수 r이 생성
         double h = 5:
04
                                    main()
05
                                      double r 7.0
06
07
         System.out.printf("반지름이
                                    그림 3-11 지역변수 r의 생성
                          r, h, v);
80
09
                                   04행이 수행되어 지역변수 h가 추가
                                    main()
10
                                      double r 7.0
12
         return Math.PI * radius * ra
13
                                    그림 3-11 지역변수 r의 생성
14 }
```

```
다중 입력변수 메소드: 원기둥의 부피
실습 3-2
                                                     ch03/ex3_2/Cylinder.java
01 public class Cylinder {
02
      public static void main(String[] args) {
03
          double r = 7;
         double h = 5;
04
                                   05행의 수행으로 volume(r, h) 메소드가 호출되면서
          double v = volume(r, h);
05
                                   실행 흐름은 11행으로 넘어감
06
07
          System.out.printf("반지름이 %.1f, 높이가 %.1f인 원기둥의 부피: %.1f",
                           r, h, v);
80
09
10
11
12
          return Math.PI * radius * radius * height;
13
14 }
```

```
다중 입력변수 메소드: 원기둥의 부피
실습 3-2
                                                    ch03/ex3_2/Cylinder.java
                                  메소드 호출 시 전달값 r과 h가 입력변수 radius와 height로 대입
01 public class Cylinder {
                                    main()
02
      public static void main(String[
         double r = 7;
03
                                                7.0
                                                                  5.0
                                      double r
                                                        double h
         double h = 5;
04
         double v = volume(r, h);
05
                                    volume(double radius 7.0, double height
                                                                           5.0
06
07
         System.out.printf("반지름이
80
                                    그림 3-13 volume() 메소드 호출과 입력값의 전달
09
      // 반지름의 길이와 높이를 입력받아, 원기둥의 부피를 반환하는 메소드
10
      public static double volume(double radius, double height) {
11
12
         return Math.PI * radius * radius * height;
13
14 }
                                            ·-----메소드 정의부
```

```
다중 입력변수 메소드: 원기둥의 부피
실습 3-2
                                                        ch03/ex3_2/Cylinder.java
01 public class Cylinder {
      public static void main(String 12행의 반환값이 계산되고, 이를 double 타입으로 해석하여 메소드의
02
          double r = 7;
03
                                    호출 위치로 반환
          double h = 5;
04
                                     main()
                                                                                  (double) 769,6902...
          double v = volume(r, h);
05
                                       double r 7.0
                                                     double h 5.0
06
          System.out.printf("반지름이
07
                                      volume(double radius 7.0 , double height 5.0 )
                                              return Math.PI * radius * radius * height
80
                                      그림 3-14 volume() 메소드의 반환값
09
       // 반지름의 길이와 높이를 입력받아, 원기둥의 부피를 반환하는 메소드
10
       public static double volume(double radius, double height) {
11
12
          return Math.PI * radius * radius * height;
13
14 }
                                               ·------메소드 정의부
실행 결과
```

```
다중 입력변수 메소드: 원기둥의 부피
실습 3-2
                                                        ch03/ex3_2/Cylinder.java
01 public class Cylinder {
      public static void main(Strin 실행 흐름은 05행으로 돌아와 반환값을 지역변수 v에 대입
02
                                     main()
          double r = 7;
03
                                              7.0
                                                             5.0
                                                                           769.7
                                                     double h
                                                                    double v
                                      double r
          double h = 5;
04
          double v = volume(r, h);
05
                                     volume(double radius 7.0 , double height 5.0 )
06
07
          System.out.printf("반지름(
                                             return Math.PI * radius * radius * height
                                    그림 3-15 volume() 메소드의 반환값 전달과 대입
08
09
       // 반지름의 길이와 높이를 입력받아, 원기둥의 부피를 반환하는 메소드
10
       public static double volume(double radius, double height) {
11
12
          return Math.PI * radius * radius * height;
13
14 }
                                                ·------메소드 정의부
```

```
다중 입력변수 메소드: 원기둥의 부피
실습 3-2
                                                   ch03/ex3_2/Cylinder.java
01 public class Cylinder {
      public static void main(String[] args) {
02
03
         double r = 7;
         double h = 5;
04
         double v = volume(r, h); -----메소드 호출부
05
06
07
         System.out.printf("반지름이 %.1f, 높이가 %.1f인 원기둥의 부피: %.1f",
                          r, h, v);
                                                 끝으로 07행의 형식 문자열을 출력함으로써
08
                                                 프로그램이 종료
09
      // 반지름의 길이와 높이를 입력받아, 원기둥의 부피를 반환하는 메소드
10
      public static double volume(double radius, double height) {
11
         return Math.PI * radius * radius * height;
12
13
14 }
                                           ·-----메소드 정의부
```

Ⅲ. 입력변수가 없는 메소드

Math.random() 메소드는 0 이상 1 미만의 임의의 실수를 반환이를 활용하면 다양한 범위의 난수(임의의 수)를 생성할 수 있음

```
double x = Math.random(); // 0.0 <= x < 1.0
```

TIP **난수** random number란 임의의 수입니다.

```
실습 3-3
          입력변수가 없는 메소드: 주사위의 눈값 출력하기 ch03/ex3_3/RandomNumber.java
01 public class RandomNumber {
02
      public static void main(String[] args) {
          int n = rollDie(); -------메소드 호출부
03
          System.out.printf("주사위의 눈: %d", n);
04
05
06
07
      // 1 이상 6 이하의 임의의 정수를 반환하는 메소드
      public static int rollDie() {
80
09
          double x = 6 * Math.random();
                                     ···········메소드 정의부
10
          int temp = (int) x;
11
          return temp + 1;
12
13
```

주사위의 눈: N ------1~6 사이의 임의의 정수

🖵 실습 코드

https://github.com/cloudstudyingkr/JavaPlz/blob/master/src/part1/ch03/ex3 3/RandomNumber_iava

```
실습 3-3
          입력변수가 없는 메소드: 주사위의 눈값 출력하기 ch03/ex3_3/RandomNumber.java
01 public class RandomNumber {
      public static void main(String[] args) {
02
                                           03행의 수행으로 rollDie() 메소드가 호출
          int n = rollDie();
03
                                           실행 흐름은 해당 메소드로 넘어감
          System.out.printf("주사위의 눈: %d'
04
                                             main()
05
06
07
08
                                             rollDie()
09
          double x = 6 * Math.random();
10
          int temp = (int) x;
11
          return temp + 1;
12
                                            그림 3-16 rollDie() 메소드의 호출
13
```

```
실습 3-3
         입력변수가 없는 메소드: 주사위의 눈값 출력하기 ch03/ex3_3/RandomNumber.java
01 public class RandomNumber {
02
      public static void main(String[] args) {
                                                 09행의 수행으로 rollDie() 메소드의 지역
         변수 x의 범위는 0.0 이상 6.0 미만의 실수
03
                                                  main()
         System.out.printf("주사위의 눈: %d", n);
04
05
06
      // 1 이상 6 이하의 임의의 정수를 반환하는 메소드
07
     public static int rollDie() {
80
                                                  rollDie()
09
         double x = 6 * Math.random();
                                                   double x 0.0 이상 6.0 미만의 실수
                                  ----대소드 정의부
10
         int temp = (int) x;
11
         return temp + 1;
                                                 그림 3-17 지역변수 x의 생성
12
```

13 }

```
실습 3-3
         입력변수가 없는 메소드: 주사위의 눈값 출력하기 ch03/ex3_3/RandomNumber.java
01 public class RandomNumber {
                                                  10행의 수행으로 temp의 범위는 0 이상
      public static void main(String[] args) {
02
                                                  6 미만의 정수
         03
                                                  main()
04
         System.out.printf("주사위의 눈: %d", n);
05
06
      // 1 이상 6 이하의 임의의 정수를 반환하는 메소드
07
                                                  rollDie()
      public static int rollDie() {
80
                                                   double x 0.0 이상 6.0 미만의 실수
09
         double x = 6 * Math.random();
                                  ------메소드 정의부
         int temp = (int) x;
10
                                                             0 이상 6 미만의 정수
                                                    int temp
11
         return temp + 1;
                                                  그림 3-18 지역변수 temp의 추가 생성
12
13 }
```

```
실습 3-3
          입력변수가 없는 메소드: 주사위의 눈값 출력하기 ch03/ex3_3/RandomNumber.java
01 public class RandomNumber {
                                            11행의 수행으로 반환값은 1 이상 6 이하의 임의의 정수
02
      public static void main(String[] args)
                                            즉, 주사위 눈값을 반환
          int n = rollDie();
03
                                            main()
          System.out.printf("주사위의 눈: %d",
04
                                                                  1~6 사이의 임의 정수 N
05
06
      // 1 이상 6 이하의 임의의 정수를 반환하는
07
                                            rollDie()
      public static int rollDie() {
80
                                                      0.0 이상 6.0 미만의 실수
                                              double x
09
          double x = 6 * Math.random();
                                                                          return temp + 1;
10
          int temp = (int) x;
                                                        0 이상 6 미만의 정수
                                              int temp
11
          return temp + 1;
                                            그림 3-19 rollDie() 메소드의 반환값
12
13
```

```
실습 3-3
          입력변수가 없는 메소드: 주사위의 눈값 출력하기 ch03/ex3_3/RandomNumber.java
01 public class RandomNumber {
      public static void main(String[] args)실행 흐름은 다시 03행으로 돌아와 메인 메소드의 지역변수
02
          int n = rollDie();
                                          n에 반환값을 대입
03
                                           main()
          System.out.printf("주사위의 눈: %d"
04
                                                                 1∼6 사이의 임의 정수 N
05
                                                      N -
                                                int n
06
07
      // 1 이상 6 이하의 임의의 정수를 반환하는
                                           rollDie()
      public static int rollDie() {
80
09
          double x = 6 * Math.random();
                                             double x 0.0 이상 6.0 미만의 실수
                                                                         return temp + 1;
10
          int temp = (int) x;
                                                       0 이상 6 미만의 정수
                                             int temp
11
          return temp + 1;
12
                                           그림 3-20 rollDie() 메소드의 반환값과 대입
13
```

```
실습 3-3
         입력변수가 없는 메소드: 주사위의 눈값 출력하기 ch03/ex3_3/RandomNumber.java
01 public class RandomNumber {
02
     public static void main(String[] args) {
        03
        System.out.printf("주사위의 눈: %d", n); 끝으로 04행의 형식 문자열을 출력함으로써
04
                                          프로그램이 종료
05
06
07
     // 1 이상 6 이하의 임의의 정수를 반환하는 메소드
     public static int rollDie() {
80
09
        double x = 6 * Math.random();
                                 ···········메소드 정의부
        int temp = (int) x;
10
11
        return temp + 1;
12
13 }
```

IV. 반환값이 없는 메소드

void 메소드: 메소드의 반환값은 존재하지 않는 메소드

- 이 경우 반환 타입은 void가 되고, return 키워드는 생략 가능

```
public static void printHello() {
    System.out.println("Hello");
    return;
    생략 가능
}
```

```
반환값이 없는 메소드: 라면 먹기
실습 3-4
                                                  ch03/ex3_4/Ramyun.java
01 public class Ramyun {
      public static void main(String[] args) {
02
         buy();----1
03
         boil();--2
04
         put();---3
05
         eat();---4
06
07
80
      public static void boil() {
09
                                                          실행 결과
         System.out.println("물을 끓인다.");
10
11
                                                        라면을 사온다.
12
13
      public static void buy() {
                                                        물을 끓인다.
         System.out.println("라면을 사온다.");---- @
14
                                                        라면을 넣는다.
15
16
                                                         맛있게 먹는다.
      public static void eat() {
17
         System.out.println("맛있게 먹는다.");---- @
18
19
20
                                                            □ 실습 코드
      public static void put() {
21
                                                             https://github.com/cloudstudyi
         System.out.println("라면을 넣는다.");---ⓒ
22
                                                            kr/JavaPlz/blob/master/src/part
23
                                                            1/ch03/ex3 4/Ramyun jaya
24 }
```

```
반환값이 없는 메소드: 라면 먹기
실습 3-4
                                                   ch03/ex3_4/Ramyun.java
01 public class Ramyun {
      public static void main(String[] args) {
02
                                                 buy() 메소드의 호출로 "라면을 사온다."가 출력
         buy();----1
03
04
05
06
07
80
09
10
11
12
13
      public static void buy() {
         System.out.println("라면을 사온다.");---- @
14
15
16
17
18
19
21
22
23
24 }
```

```
반환값이 없는 메소드: 라면 먹기
실습 3-4
                                                    ch03/ex3_4/Ramyun.java
01 public class Ramyun {
      public static void main(String[] args) {
02
         buy();----1
03
                                                 boil() 메소드의 호출로 "물을 끓인다."가 출력
         boil();---2
04
05
06
07
80
      public static void boil() {
09
         System.out.println("물을 끓인다."); b
10
11
12
13
      public static void buy() {
         System.out.println("라면을 사온다.");---- @
14
15
16
17
18
19
21
22
23
24 }
```

```
반환값이 없는 메소드: 라면 먹기
실습 3-4
                                                    ch03/ex3_4/Ramyun.java
01 public class Ramyun {
      public static void main(String[] args) {
02
         buy();----1
03
         boil();--2
04
                                                 put() 메소드의 호출로 "라면을 넣는다."가 출력
         put();---3
05
06
07
80
      public static void boil() {
09
         System.out.println("물을 끓인다."); b
10
11
12
13
      public static void buy() {
         System.out.println("라면을 사온다.");---- @
14
15
16
17
18
19
20
      public static void put() {
21
         System.out.println("라면을 넣는다.");---ⓒ
22
23
24 }
```

```
반환값이 없는 메소드: 라면 먹기
실습 3-4
                                                ch03/ex3_4/Ramyun.java
01 public class Ramyun {
      public static void main(String[] args) {
02
         buy();----1
03
         boil();--2
04
         put();---3
05
                                              eat() 메소드의 호출로 "맛있게 먹는다."가 출력
         eat(); 4
06
07
80
      public static void boil() {
09
                                                        실행 결과
         System.out.println("물을 끓인다.");
10
11
                                                      라면을 사온다.
12
13
      public static void buy() {
                                                      물을 끓인다.
         System.out.println("라면을 사온다.");---- @
14
                                                      라면을 넣는다.
15
16
                                                      맛있게 먹는다.
      public static void eat() {
17
         System.out.println("맛있게 먹는다.");---- @
18
19
20
      public static void put() {
21
         System.out.println("라면을 넣는다.");---ⓒ
22
23
24 }
```

Quiz

03 다음 설명 중 옳은 것을 모두 고르시오.

- ① 메소드의 입력변수는 없을 수 있다.
- ② 모든 메소드는 값을 반환해야 한다.
- ③ void 메소드는 return 키워드를 생략할 수 있다.
- ④ 메소드 호출이 종료되면 실행 흐름은 호출된 위치로 돌아간다.

Thank You!

