2024학년도 1학기

신입생 Java 교육

교육부장 20 이어진

교육 커리큘럼

1주차 변수,연산자 + 조건문(if)
2주차 조건문(switch) + 반복문
3주차 배열

<중간고사>

4주차 객체, 메소드 오버로딩과 생성자
5주차 상속과 오버라이딩
6주차 다형성과 인터페이스
7주차 예외처리

<기말고사>

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 1

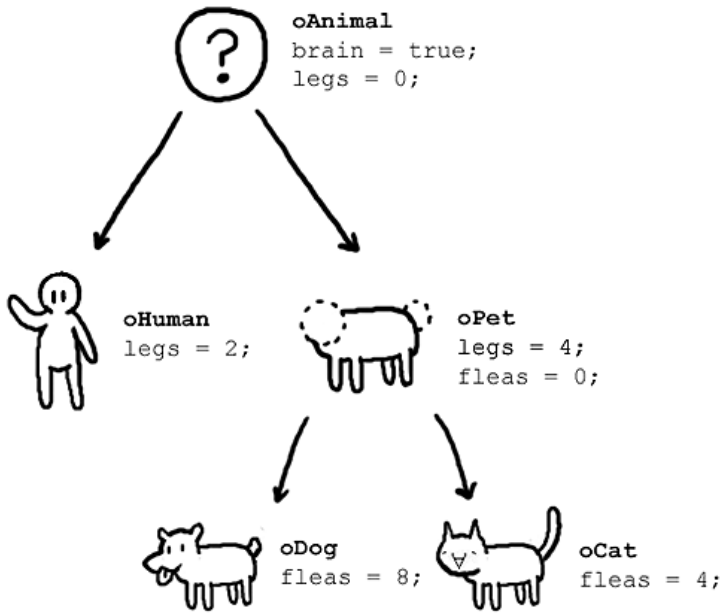
객체지향 프로그래밍

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 1 객체지향 프로그래밍?

자바는 객체지향 프로그래밍 언어다.

객체지향 프로그래밍이란?
⇒객체를 속성과 기능으로 정의한 다음
객체들 간의 상호동작 방식을 프로그램으로 표현한 것



ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 1 객체지향 프로그래밍의 특성

▶ 코드의 재사용성이 높다.
- 새로운 코드를 작성할 때 기존의 코드를 이용해서 쉽게 작성할 수 있다.

▶ 코드의 관리가 쉬워졌다.
- 코드간의 관계를 맺어줌으로써 보다 적은 노력으로 코드변경이 가능하다.

▶ 신뢰성이 높은 프로그램의 개발을 가능하게 한다.
- 제어자와 메서드를 이용해서 데이터를 보호하고, 코드의 중복을 제거하여 코드의 불일치로 인한 오류를 방지할 수 있다.

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 2

클래스와 객체

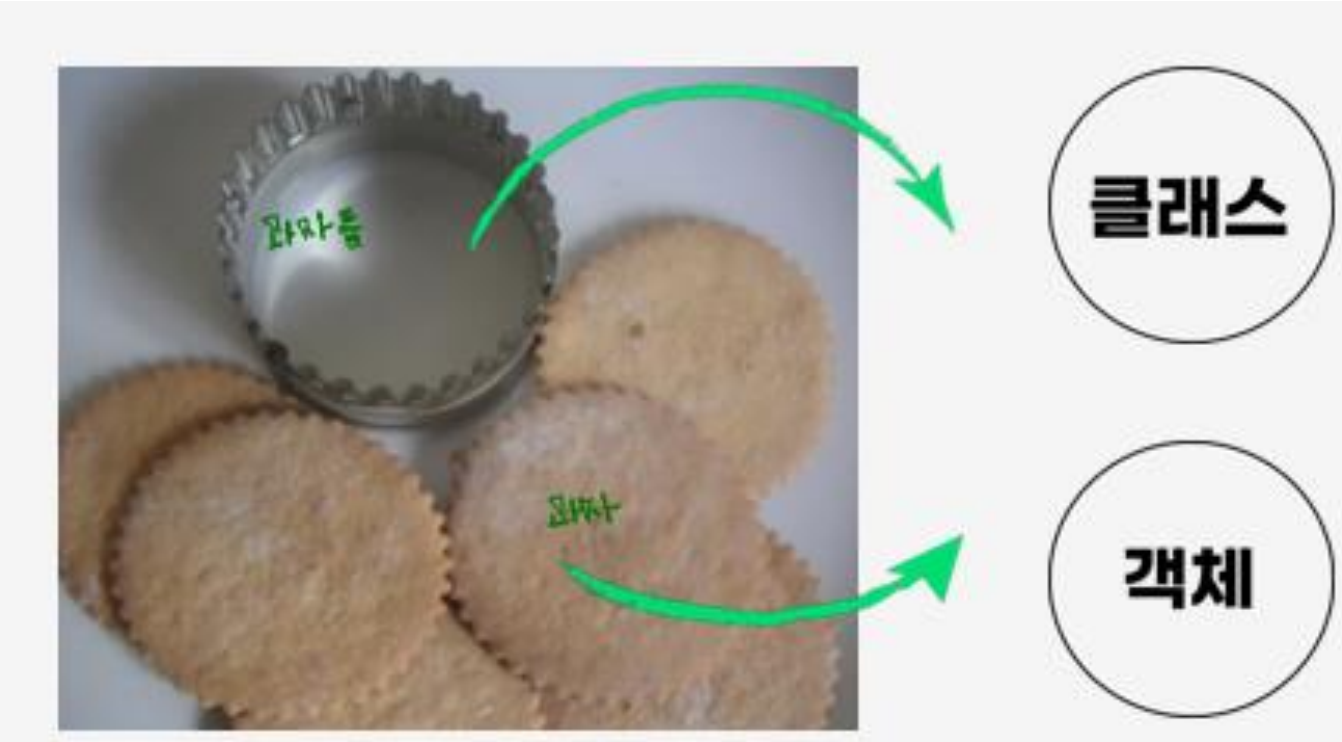
ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 2

클래스와 객체의 정의

클래스 : 객체를 정의하는 틀 또는 설계도

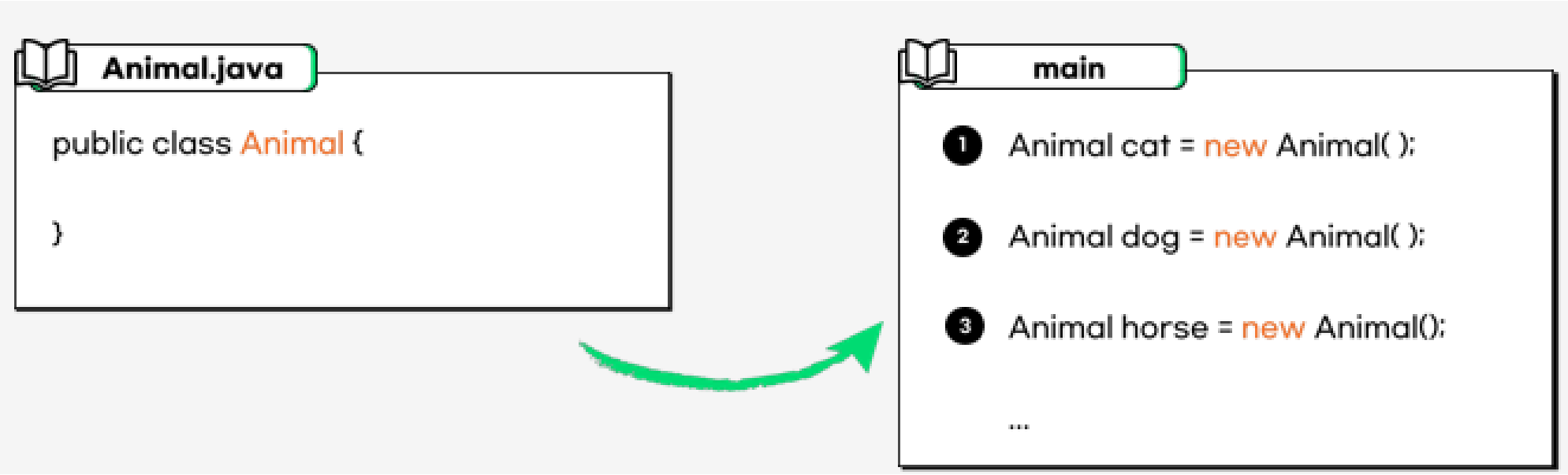
객체 : 실제로 존재하는 것. 사물 또는 개념.



ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 2 클래스와 객체의 관계

무수히 많은 동물 객체(cat, dog, horse, …)들을
Animal 클래스로 만들 수 있다.



ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 2 인스턴스

클래스에 의해서 만들어진 객체를 인스턴스라고 한다.

즉, 객체 ≒ 인스턴스

```
Animal cat = new Animal( );
```

1. cat은 객체이다.
2. cat이라는 객체는 Animal의 인스턴스(instance)이다.

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 3

객체변수와 메소드

ⓒSaebyeol Yu. Saebyeol's PowerPoint

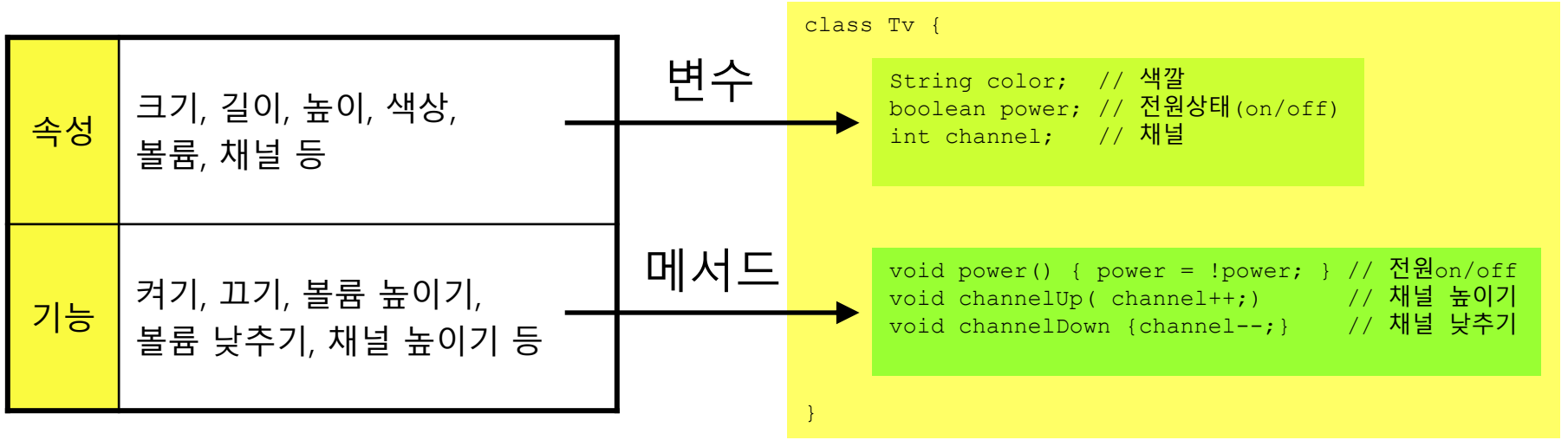
Part 3 객체의 구성요소

▶ 객체는 속성과 기능으로 이루어져 있다.

- 객체는 속성과 기능의 집합이며, 속성과 기능을 객체의 멤버(member, 구성 요소)라고 한다.

▶ 속성은 변수로, 기능은 메서드로 정의한다.

- 클래스를 정의할 때 객체의 속성은 변수로, 기능은 메서드로 정의한다.



ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 3 객 체 변 수

객체 변수?
=> 클래스에 선언된 변수

Animal.java

```
public class Animal {
    String name;

    public static void main(String[] args) {
        Animal cat = new Animal();
        System.out.println(cat.name);
    }
}
```

설명

객체 변수는 도트연산자(.)를 이용하여 접근할 수 있다.

" 객체.객체변수 "

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 3 메소드의 구조

메소드?
=> 클래스 내에 구현된 함수

```
public 리턴자료형 메소드명(입력자료형1 입력변수1, 입력자료형2 입력변수2, ...) {
    ...
    return 리턴값; // 리턴자료형이 void 인 경우에는 return 문이 필요없다.
}
```

```
public class Main {
    Run | Debug
    public static void main(String[] args) {
        Calculator c = new Calculator();
        int result = c.add(1, 2);
        System.out.println(result);
    }
}
```

```
public class Calculator {
    public int add(int a, int b) {
        return a + b;
    }
}
```

```
3
```

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 3 메소드 종류 – 입력 값O, 리턴 값O

```java
public class Test {
    public int sum(int a, int b) {
        return a + b;
    }

    Run | Debug
    public static void main(String[] args) {
        Test test = new Test();
        int result = test.sum(1, 2);
        System.out.println(result);
    }
}
```

리턴 자료형 : int
메소드 명 : sum

입력 값 : int a, int b
리턴 값 : a + b

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 3 메소드 종류 – 입력 값X, 리턴 값O

```java
public class Test {
    public String say() {
        return "Hi";
    }

    Run | Debug
    public static void main(String[] args) {
        Test test = new Test();
        String result = test.say();
        System.out.println(result);
    }
}
```

리턴 자료형 : String
메소드 명 : say

입력 값 : X
리턴 값 : “Hi”

```
Hi
```

ⓒSaebyeol Yu. Saebyeol’s PowerPoint

Part 3 메소드 종류 – 입력 값 O, 리턴 값 X

```java
public class Test {
    public void say_sum(int a, int b) {
        System.out.println(a + " + " + b + " = " + (a+b));
    }

    Run | Debug
    public static void main(String[] args) {
        Test test = new Test();
        test.say_sum(1, 2);
    }
}
```

리턴 자료형 : void
메소드 명 : say_sum

입력 값 : int a, int b
리턴 값 : X

```
1 + 2 = 3
```

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 3 메소드 종류 – 입력 값X, 리턴 값X

```java
public class Test {
    public void say_hi() {
        System.out.println("Hi");
    }

    Run | Debug
    public static void main(String[] args) {
        Test test = new Test();
        test.say_hi();
    }
}
```

리턴 자료형 : void
메소드 명 : say_hi

입력 값 : X
리턴 값 : X

```
Hi
```

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 3 void에서 return

```java
public class Test {
    public void equal(int a) {
        if (a == 0) {
            return;
        }
        System.out.println("a is not 0");
    }

    Run | Debug
    public static void main(String[] args) {
        Test test = new Test();
        test.equal(0);
    }
}
```

출력 X

```java
public class Test {
    public void equal(int a) {
        if (a == 0) {
            return;
        }
        System.out.println("a is not 0");
    }

    Run | Debug
    public static void main(String[] args) {
        Test test = new Test();
        test.equal(1);
    }
}
```

```
a is not 0
```

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 4

메소드 오버로딩

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 4

메소드 오버로딩

메소드 오버로딩?
⇒하나의 클래스에 같은 이름의 메소드를
여러 개 정의하는 것

```java
public class Test {
    public void say() {
        System.out.println("입력 값 : X");
    }
    public void say(int a) {
        System.out.println("입력 값 : " + a);
    }
    public void say(String s) {
        System.out.println("입력 값 : " + s);
    }

    Run | Debug
    public static void main(String[] args) {
        Test test = new Test();
        test.say(); // 입력 값 : X
        test.say(1); // 입력 값 : 1
        test.say("Hi"); // 입력 값 : Hi
    }
}
```

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 4

메소드 오버로딩의 조건

1. 메소드 이름이 같아야 한다.
2. 매개변수의 개수 or 타입이 달라야 한다.

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 4

메소드 오버로딩이 아닌 예시

1. 메소드 이름이 같아야 한다.
2. 매개변수의 개수 or 타입이 달라야 한다.

- 매개변수의 이름이 다른 것은 오버로딩이 아니다.

```
[보기1]
int add(int a, int b) { return a+b; }
int add(int x, int y) { return x+y; }
```

- 리턴타입은 오버로딩의 성립조건이 아니다. 그냥 다른 함수.

```
[보기2]
int add(int a, int b) { return a+b; }
long add(int a, int b) { return (long)(a + b); }
```

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 4

메소드 오버로딩의 예시

1. 메소드 이름이 같아야 한다.
2. 매개변수의 개수 or 타입이 달라야 한다.

- 매개변수의 개수가 다르므로 오버로딩이 성립한다.

```
public int add (int a, int b) {
    return a + b;
}
public int add (int a) {
    return a + 10;
}
```

- 매개변수의 타입이 다르므로 오버로딩이 성립한다.

```
[보기3]
long add(int a, long b) { return a+b; }
long add(long a, int b) { return a+b; }
```

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 5

생성자

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 5 생 성 자

생성자?
=> 인스턴스가 생성될 때마다 호출되는 인스턴스 초기화 메소드

```
public class Tv {
    String color;
    boolean power;
    int channel;

    void power() {
        power = !power;
    }
    void channelUp() {
        ++channel;
    }
    void channelDown() {
        --channel;
    }
}
```

=> 객체 변수의 초기화 필요

```
public class Tv {
    String color;
    boolean power;
    int channel;

    Tv() {
        color = "white";
        power = false;
        channel = 0;
    }
```

매개변수가 없는 생성자

```
public class Tv {
    String color;
    boolean power;
    int channel;

    Tv(String color, boolean power, int channel) {
        this.color = color;
        this.power = power;
        this.channel = channel;
    }
```

매개변수가 있는 생성자

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 5 기 본 생 성 자

기본생성자?
⇒매개변수가 없는 생성자
⇒클래스에 생성자가 하나도 없으면 컴파일러가 기본 생성자를 추가한다.

```
public class Tv {
    String color;
    boolean power;
    int channel;

    Tv() { }

    void power() {
        power = !power;
    }
    void channelUp() {
        ++channel;
    }
    void channelDown() {
        --channel;
    }
}
```

=> 기본생성자

모든 클래스에는 반드시
하나 이상의 생성자가 있어야 한다.

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 5 생성자 예시

```java
public class Tv {
    String color;
    boolean power;
    int channel;

    Tv() {
        color = "white";
        power = false;
        channel = 0;
    }

    Tv(String color, boolean power, int channel) {
        this.color = color;
        this.power = power;
        this.channel = channel;
    }

    void power() {
        power = !power;
    }
    void channelUp() {
        ++channel;
    }
    void channelDown() {
        --channel;
    }
}
```

```java
public class Main {
    Run | Debug
    public static void main(String[] args) {
        Tv tv1 = new Tv();
        Tv tv2 = new Tv("black", true, 7);

        System.out.println("tv1의 color: " + tv1.color + ", power: " + tv1.power + ", channel: " + tv1.channel);
        System.out.println("tv2의 color: " + tv2.color + ", power: " + tv2.power + ", channel: " + tv2.channel);
    }
}
```

```
tv1의 color: white, power: false, channel: 0
tv2의 color: black, power: true, channel: 7
```

ⓒSaebyeol Yu. Saebyeol's PowerPoint

실습문제

[문제]
사각형의 변의 길이에 따라
사각형의 넓이와 둘레를 다르게 출력하는 클래스를 만들기

[조건]
· 사각형의 변의 길이는 int로 1개나 2개를 입력 받는다.

오늘은 백준 사용 X
다 푼 사람은 손 들면 임원들이 가서 검사 후 귀가 가능 ^^
못 풀어도 시간 되면 집 보내 드립니다! 걱정마세요~

해설

```java
public class Quadrangle {
    int a;
    int b;

    Quadrangle(int a) {
        [redacted]
        [redacted]
    }

    Quadrangle(int a, int b) {
        [redacted]
        [redacted]
    }

    public int area() {
        [redacted]
    }

    public int perimeter() {
        [redacted]
    }
}
```

매개변수를 하나만 받는 생성자
this.a, this.b 초기화

매개변수를 두 개 받는 생성자
this.a, this.b 초기화

사각형의 넓이를 반환하는 메소드

사각형의 둘레를 반환하는 메소드

```java
public class Main {
    Run | Debug
    public static void main(String[] args) {
        Quadrangle q1 = new Quadrangle(3);
        System.out.println("사각형의 넓이 : " + q1.area());
        System.out.println("사각형의 둘레 : " + q1.perimeter());

        Quadrangle q2 = new Quadrangle(3, 4);
        System.out.println("사각형의 넓이 : " + q2.area());
        System.out.println("사각형의 둘레 : " + q2.perimeter());
    }
}
```

3을 입력받은 경우 출력 =>

```
사각형의 넓이 : 9
사각형의 둘레 : 12
```

3 4를 입력받은 경우 출력 =>

```
사각형의 넓이 : 12
사각형의 둘레 : 14
```

해설

```java
public class Quadrangle {
    int a;
    int b;

    Quadrangle(int a) {
        this.a = a;
        this.b = a;
    }

    Quadrangle(int a, int b) {
        this.a = a;
        this.b = b;
    }

    public int area() {
        return a * b;
    }

    public int perimeter() {
        return 2 * (a + b);
    }
}
```

매개변수를 하나만 받는 생성자
this.a, this.b 초기화

매개변수를 두 개 받는 생성자
this.a, this.b 초기화

사각형의 넓이를 반환하는 메소드

사각형의 둘레를 반환하는 메소드

```java
public class Main {
    Run | Debug
    public static void main(String[] args) {
        Quadrangle q1 = new Quadrangle(3);
        System.out.println("사각형의 넓이 : " + q1.area());
        System.out.println("사각형의 둘레 : " + q1.perimeter());

        Quadrangle q2 = new Quadrangle(3, 4);
        System.out.println("사각형의 넓이 : " + q2.area());
        System.out.println("사각형의 둘레 : " + q2.perimeter());
    }
}
```

3을 입력받은 경우 출력 =>

```
사각형의 넓이 : 9
사각형의 둘레 : 12
```

3 4를 입력받은 경우 출력 =>

```
사각형의 넓이 : 12
사각형의 둘레 : 14
```

Q&A

감사합니다

ⓒSaebyeol Yu. Saebyeol's PowerPoint