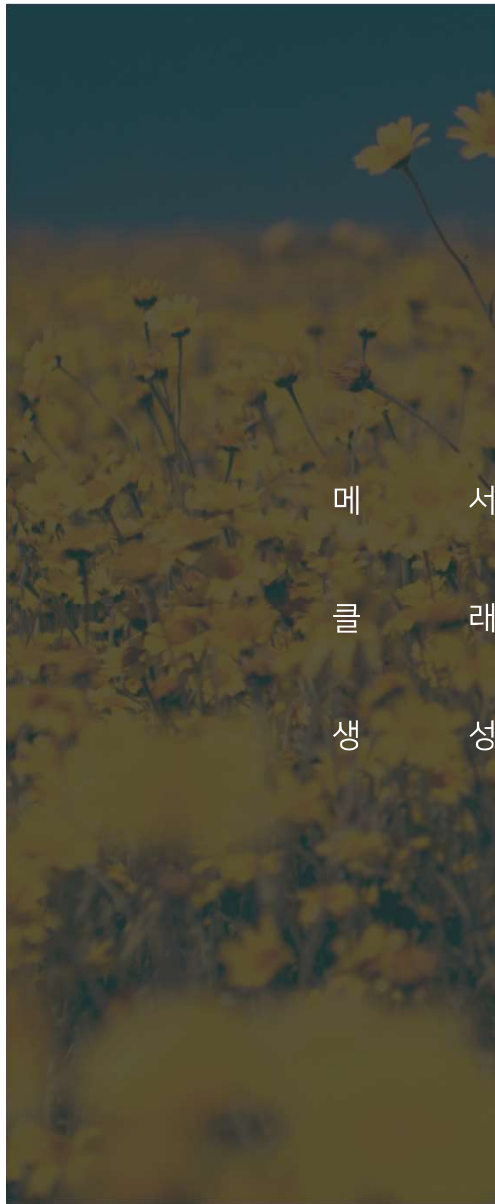




JAVA

GU, DA HAE



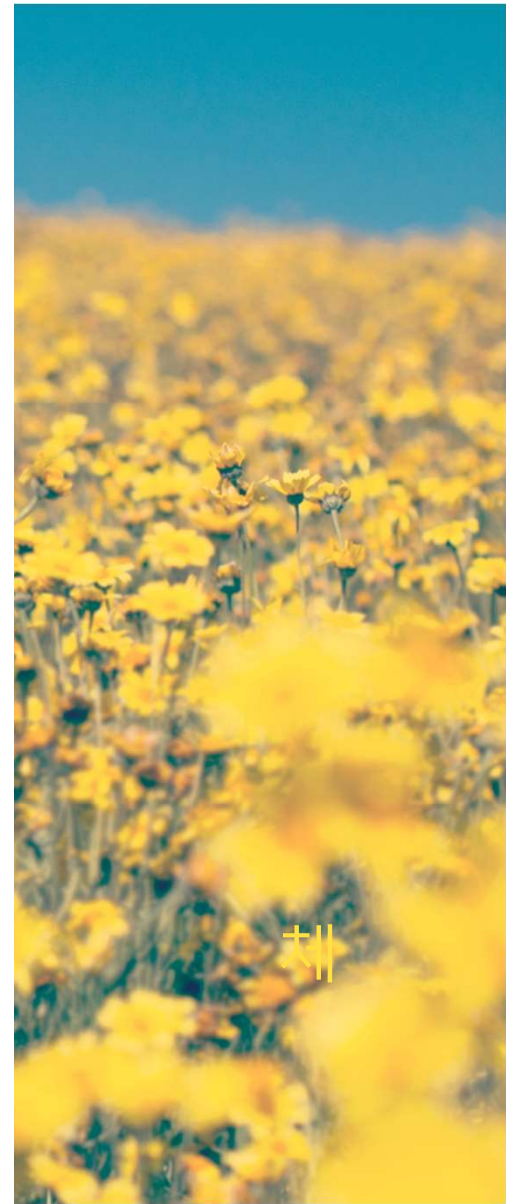
서
래
성
메
클
생

드
스
자

JAVA

객

체



메서드

Method

메서드는 C/C++의 함수와 같다.

```
public static 자료형 식별자(매개변수1, 매개변수2, ...){  
    ...  
}
```

자바 프로그램은 main 메서드 호출부터 시작한다.

프로그램 시작을 목적으로 하지 않는다면, main이라는 식별자 외에 다른 식별자로 메서드를 만들 수 있다.

메서드는 여러 번 호출할 수 있다.

J A V A

예제

```
public class Ex01{  
    public static void main(String[] args){  
        System.out.println("Good Morning!");  
        myName("James");  
        System.out.println("Bye!");  
    }  
    public static void myName(String name){  
        System.out.println("My Name is " + name);  
    }  
}
```

J A V A

예제

```
public class Ex02{  
    public static void main(String[] args){  
        System.out.println("Good Morning!");  
        myName("James");  
        myAge(30);  
        MyName("Chris");  
        myAge(15);  
        System.out.println("Bye!");  
    }  
    public static void myName(String name){  
        System.out.println("My Name is " + name);  
    }  
    public static void myAge(int age){  
        System.out.println("My age is " + age);  
    }  
}
```

J A V A

예제

```
public class Ex03{  
    public static void main(String[] args){  
        System.out.println(add(3.7, 8.7));  
        System.out.println(sub(12.87, 34.76));  
    }  
    public static double add(double x, double y){  
        return x+y;  
    }  
    public static double sub(double x, double y){  
        return x-y;  
    }  
}
```

예제

```
public class Ex04{
    public static void main(String[] args){
        divide(12, 3);
        divide(5, 0);
    }
    public static void divide(double x, double y){
        if(y == 0){
            System.out.println("DON'T DIVIDE BY ZERO");
            return;
        }
        else System.out.println("DIVIDE RESULT : " + x/y);
    }
}
```

J A V A

해보기 1

원의 반지름을 인자로 전달하여 원의 둘레를 반환하는 메서드, 원의 넓이를 반환하는 메서드를 각각 정의해보자.

J A V A

해보기 2

인자로 전달된 값이 소수(Prime Number)인지 검사하여, 소수면 true, 소수가 아니면 false를 반환하는 메서드를 정의하자. 이를 통해 1 이상 50 이하의 소수를 전부 출력해보자.

메서드 재귀 호출

Recursion

메서드 내부에서 자기자신을 호출할 수 있다. 이를 재귀 호출이라 한다.

```
public static void method(){  
    method();  
}
```

J A V A

예제

```
public class Ex01{  
    public static void main(String[] args){  
        hello();  
    }  
    public static void hello(){  
        System.out.println("HELLO!");  
        hello();  
    }  
}
```

예제

```
public class Ex02{  
    public static void main(String[] args){  
        hello(5);  
    }  
    public static void hello(int num){  
        if(num == 0) // 재귀호출 종료 조건  
            return;  
  
        System.out.println("HELLO!");  
        hello(num - 1);  
    }  
}
```

J A V A

해보기 1

팩토리얼을 계산하는 재귀 메서드를 정의해보자.

해보기 2

정수 n 을 전달받아서, 2의 n 승을 반환하는 재귀 메서드를 정의해보자.

J A V A

클래스

클래스는 C/C++의 구조체와 같은 역할을 한다. 다만 함수도 멤버로 포함하고 있을 뿐이다.

```
class 식별자{  
    멤버 변수;  
    메서드;  
}
```

```
클래스이름 식별자 = new 클래스이름();
```

새로운 타입(type)을 정의한다.

예제

```
class Person{
    String name;
    int age;
    char gender;
}

public class Ex01{
    public static void main(String[] args){
        Person p1 = new Person();
        p1.name = "Louis";
        p1.age = 55;
        p1.gender = 'm';

        printPersonInfo(p1);
    }
}
```

```
public static void printPersonInfo(Person
p){
    System.out.println("My name is " +
p.name);
    System.out.println("My age is " +
p.age);
    if(p.gender == 'm')
        System.out.println("I'm male.");
    else System.out.println("I'm female.");
}
}
```


예제

```
class Person{
    String name;
    int age;
    char gender;

    void printPersonInfo(){
        System.out.println("My name is " +
            name);
        System.out.println("My age is " + age);
        if(gender == 'm')
            System.out.println("I'm male.");
        else System.out.println("I'm female.");
    }
}
```

```
public class Ex02{
    public static void main(String[] args){
        Person p1 = new Person();
        p1.name = "Louis";
        p1.age = 55;
        p1.gender = 'm';

        p1.printPersonInfo();
    }
}
```

J A V A

Person p1 = new Person();

참조 변수

객체(인스턴스)

참조
변수
생성

객체
생성

0x100

p1

String name;
int age;
char gender;

printPersonInfo();

예제

```
class Person{
    String name;
    int age;
    char gender;

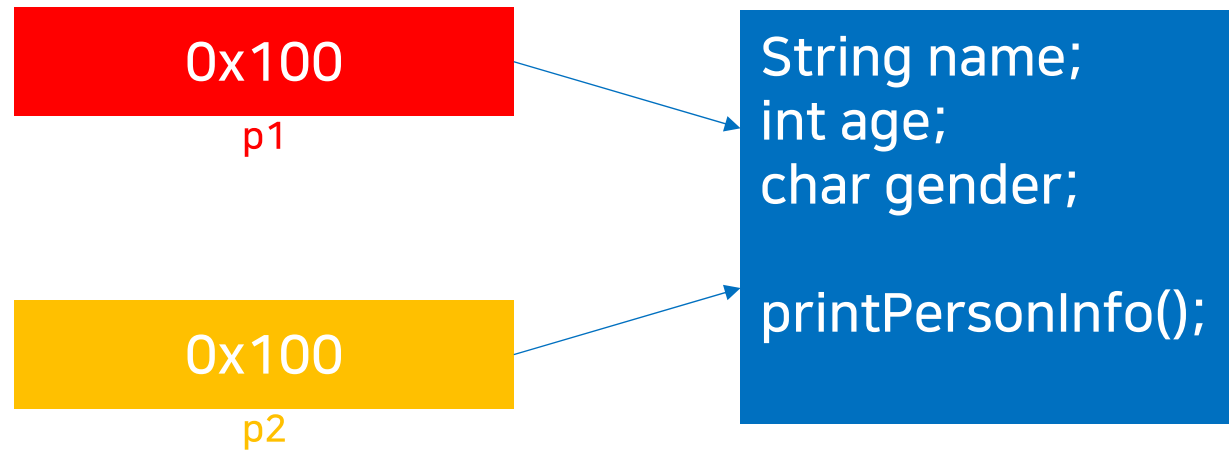
    void printPersonInfo(){
        System.out.println("My name is " +
            name);
        System.out.println("My age is " + age);
        if(gender == 'm')
            System.out.println("I'm male.");
        else System.out.println("I'm female.");
    }
}
```

```
public class Ex03{
    public static void main(String[] args){
        Person p1 = new Person();
        p1.name = "Louis";
        p1.age = 55;
        p1.gender = 'm';

        Person p2 = p1;
        p2.printPersonInfo();
    }
}
```

예제

J A V A



예제

```
class Person{
    String name;
    int age;
    char gender;

    void printPersonInfo(){
        System.out.println("My name is " +
            name);

        System.out.println("My age is " + age);
        if(gender == 'm')
            System.out.println("I'm male.");
        else System.out.println("I'm female.");
    }
}
```

```
public class Ex04{
    public static void main(String[] args){
        Person p1 = new Person();
        p1.name = "Louis";
        p1.age = 55;
        p1.gender = 'm';
    }

    public static void objectMethod(Person k){
        k.printPersonInfo();
    }
}
```

J A V A

main

0x100

p1

objectMethod

0x100

k

String name;
int age;
char gender;

printPersonInfo();

예제

```
class Person{
    String name;
    int age;
    char gender;

    void printPersonInfo(){
        System.out.println("My name is " +
            name);
        System.out.println("My age is " + age);
        if(gender == 'm')
            System.out.println("I'm male.");
        else System.out.println("I'm female.");
    }
}
```

```
public class Ex05{
    public static void main(String[] args){
        Person p1 = new Person();
        p1.name = "Louis";
        p1.age = 55;
        p1.gender = 'm';

        ModifyPerson(p1);
        p1.printPersonInfo();
    }

    public static void ModifyPerson(Person k){
        k.name = "Keneth";
        k.age = 35;
    }
}
```

예제

```
class Person{
    String name;
    int age;
    char gender;

    void printPersonInfo(){
        System.out.println("My name is " +
            name);
        System.out.println("My age is " + age);
        if(gender == 'm')
            System.out.println("I'm male.");
        else System.out.println("I'm female.");
    }
}
```

```
public class Ex06{
    public static void main(String[] args){
        Person p1 = null;
        // 특정 객체로 초기화 하지 않을 시, null로
        초기화한다.
    }
}
```


해보기

원의 반지름과 중심 좌표를 표현한 타입 circle을 정의해보자.

```
int x;  
int y;  
double radius;
```

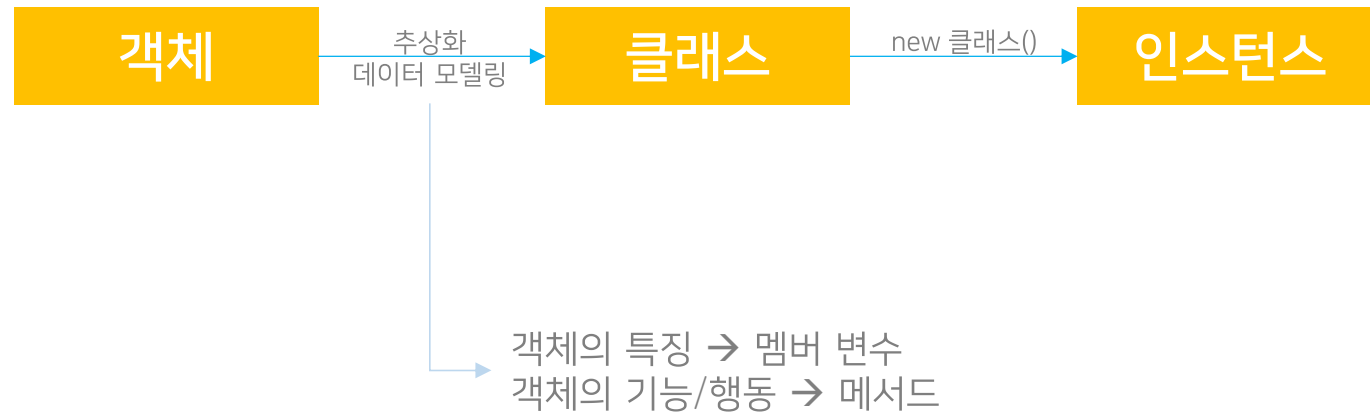
circle 타입에 아래와 같은 메서드를 정의하고, 사용해보자.

```
void showArea() - 원의 넓이 출력  
void showPeri() - 원의 둘레를 출력  
void showCircle() - 원의 정보를 출력
```

객체

Object

객체(Object; 오브젝트)는 현실 세계의 모든 사물과 대상을 가리키는 단어다.



생성자

Constructor

인스턴스를 생성할 때 단 한번 호출되는 메서드다.

```
class 클래스이름{  
    ...  
    클래스이름(매개변수, ...){  
        ...  
    }  
    ...  
}
```

생성자의 이름은 클래스이름과 같아야 한다.

생성자는 자료형이 없다.

예제

```
class Complex{  
    double real;  
    double image;  
  
    void showComplex(){  
        System.out.println(real + " + " + image  
            + "j");  
    }  
}
```

```
public class Ex01{  
    public static void main(String[] args){  
        Complex c1 = new Complex();  
        c1.real = 1;  
        c1.image = 2;  
  
        Complex c2 = new Complex();  
        c2.real = 3;  
        c2.image = 4;  
  
        Complex c3 = new Complex();  
        c3.real = 5;  
        c3.image = 6;  
    }  
}
```

J A V A

예제

```
class Complex{
    double real;
    double image;

    Complex(){
        System.out.println("Call
        Constructor!!");
    }
    void showComplex(){
        System.out.println(real + " + " + image
        + "j");
    }
}
```

```
public class Ex02{
    public static void main(String[] args){
        Complex c1 = new Complex();
        c1.real = 1;
        c1.image = 2;

        Complex c2 = new Complex();
        c2.real = 3;
        c2.image = 4;

        Complex c3 = new Complex();
        c3.real = 5;
        c3.image = 6;
    }
}
```

J A V A

Complex c1 = new Complex ();

인스턴스 생성

생성자 호출

Complex()는 인자를 전달받지 않는 생성자 호출을 뜻한다.

예제

```
class Complex{
    double real;
    double image;

    Complex(double r, double i){
        real = r;
        image = i;
    }

    void showComplex(){
        System.out.println(real + " + " + image
            + "j");
    }
}
```

```
public class Ex03{
    public static void main(String[] args){
        Complex c1 = new Complex(1, 2);
        Complex c2 = new Complex(3, 4);
        Complex c3 = new Complex(5, 6);

        c1.showComplex();
        c2.showComplex();
        c3.showComplex();
    }
}
```

디폴트 생성자

Default Constructor

클래스에 생성자가 정의되어 있지 않을 때 자동으로 정의되는 생성자.

```
클래스이름(){  
    // 비어있음  
}
```

클래스에는 반드시 생성자가 존재해야한다.

생성자가 없으면 자바 컴파일러가 생성자를 대신 생성해 준다. 이때 생성 되는 생성자를 디폴트 생성라한다.

식별자 규칙

Camel Case

OOPL에서 식별자를 작성하는 규칙을 Camel Case라 한다.

클래스 식별자 규칙

- 첫 문자는 대문자로 시작한다.
- 여러 단어를 묶어 하나의 식별자를 생성할 때, 단어의 첫 글자는 대문자로 한다.
- Person, Complex, NameCard, ...

메서드, 변수 식별자 규칙

- 클래스 식별자 규칙과 동일하나 첫 문자를 소문자로 시작한다.
- printInfomation, showMenu, ...

상수 식별자 규칙

- 모든 문자를 대문자로 한다.
- PI, COLOR, ...

해보기

밑변과 높이를 표현하는 Triangle 클래스를 정의하자. 클래스 내부에는 멤버 변수를 초기화 하는 생성자도 포함되어 있어야 한다.