

Java 프로그래밍

```
package com.day04;
```

```
public class Test01 {
public static void main(String[]args) {
    /* 1~45까지 난수 6개를 발생시켜서 배열에 넣어서 출력하시오.
     * (중복값은 허용X) - 로또 번호 등에 응용
     */
    int [] lotto = new int[6]; // 0 1 2 3 4 5
    int n = 0;
    while(n < lotto.length) { // n = 0
        int r = (int) (Math.random()*45) + 1; // 난수 = Double(실수)형 자료
        lotto[n]=r; // Array = 10 15 13
        for(int i=0; i < n; i++) {
            if(lotto[i] == r) { // 중복값 발생 10==10 10==15
                n--; // n = 0 1
                break;
            }
        }
        n++; // n = 1 2 2 3
    }
    //for~each문
    for(int i : lotto) {
        System.out.print(i+"\t");
    }
}
}
```

```
package com.day04;
```

```
public class Person {
    // 이름(name), 주소(addr), 전화번호(tel)
    String name;
    String addr;
    String tel;
    // study(), //play()
    public void study() {
        System.out.println(name+"는(은) "+" 공부합니다.");
    }
    public void play() {
        System.out.println(name+"는(은) "+" 운동합니다.");
    }
    public void addr() {
        System.out.println(name+"의"+" 주소는 "+addr+"입니다.");
    }
    public void tel() {
        System.out.println(name+"의"+" 전화번호는 "+tel+"입니다.");
    }
}
```

```

}
// p1 객체 생성
public static void main(String[]args) {
    Person p1=new Person();
    p1.name="홍길동";
    p1.tel="010-1111-2222";
    p1.addr="부산";
    p1.study();
    // 홍길동 운동한다 출력
    p1.play();
    // 부산에 사는 홍길동 전화번호는 ""이다. 출력되도록 만들어보세요.
    System.out.println(p1.addr + "에 사는 " + p1.name + " 전화번호는 " + p1.tel +
"입니다.");
    // 이름 : 홍길동, 전화 010-1111-2222, 홍길동은 공부한다. 출력

}
}

```

```

package com.day04;

```

```

public class Person {
    // 이름(name), 주소(addr), 전화번호(tel)
    String name;
    String addr;
    String tel;
    // study(), //play()
    public void study() {
        System.out.println(name+"는(은) "+" 공부합니다.");
    }
    public void play() {
        System.out.println(name+"는(은) "+" 운동합니다.");
    }
    public void addr() {
        System.out.println(name+"의"+" 주소는 "+addr+"입니다.");
    }
    public void tel() {
        System.out.println(name+"의"+" 전화번호는 "+tel+"입니다.");
    }
    // p1 객체 생성
    public static void main(String[]args) {
        Person p1=new Person();
        p1.name="홍길동";
        p1.tel="010-1111-2222";
        p1.addr="부산";
        p1.study();
        // 홍길동 운동한다 출력
        p1.play();
        // 부산에 사는 홍길동 전화번호는 ""이다. 출력되도록 만들어보세요.
        System.out.println(p1.addr + "에 사는 " + p1.name + "입니다." + " 전화번호는 " +

```

```

p1.tel + "입니다.");
    // 이름 : 홍길동, 전화 010-1111-2222, 홍길동은 공부한다. 출력
    System.out.println(p1.addr + "에 사는 " + p1.name + "입니다." + " 전화번호는 " +
p1.tel + "입니다.");
}

}

```

```

package com.day04;

```

```

public class Sachik {
    // 덧셈
    public void sum(int a, int b) {
        System.out.println("덧셈 : "+(a+b));
    }
    // 뺄셈
    public void sub(int a, int b) { // {} 표시는 지역변수
        System.out.println("뺄셈 : "+(a-b));
    }
    // 곱셈
    public void mul(int a, int b) {
        System.out.println("곱셈 : "+(a*b));
    }
    // 나눗셈
    public void div(int a, int b) {
        System.out.println("나누기 : "+(a/b));
    }
    public int div2(int i, int j) {
        int r = i/j;
        return r;
    }

    public static void main(String[] args) {
        Sachik s1 = new Sachik();
        s1.sum(20, 10); // 20+10
        s1.sub(20, 10);
        s1.mul(20, 10);
        s1.div(20, 10);
        int r = s1.div2(20, 4);
        System.out.println("리턴값 : " + r);
        System.out.println(s1.div2(100, 10));
        // System.out.println(s1.sum(10, 10)); 오류발생

    }
}

```

<디폴트 생성자와 멤버 변수>

```

package com.day04;

```

```

public class Car {

    int speed; // 멤버변수 초기값을 지정하지 않으면 0
    String carName;
    String carColor;
    public void carName() {

    }
    public void pub() {

    }

    public Car() { //디폴트 생성자 생략 가능(혼자 있을때) : 멤버 변수 초기화때 사용

    }

    public Car(String carName, String carColor) { // 생성자, this : 자기 자신의 생성자
        this.carName=carName;
        this.carColor=carColor;
    }

    public Car(String carColor) { // 생성자 OverLoading
        //this.carColor=carColor;
        this("미정",carColor);
    }

    public void speedUp(int speed) {
        this.speed +=speed;
    }
    public void stop() {
        speed = 0;
    }
    public void getSpeed() {
        System.out.println(carName+":"+speed);
    }
    public void infor() {
        System.out.println("차종:"+carName);
        System.out.println("색상:"+carColor);
        System.out.println("속도:"+speed);
        System.out.println();
    }

    public static void main(String[] args) {
        String carName;
        int speedUp;

        Car c1=new Car(); // 디폴트 생성자
        c1.speedUp(30);
    }
}

```

```

System.out.println(c1.speed);
c1.speedUp(50);
c1.speedUp(60);
System.out.println(c1.speed);
c1.stop();
System.out.println(c1.speed);
c1.speedUp(100);
System.out.println(c1.speed);
c1.carName="제네시스";
c1.getSpeed(); // 제네시스 : 100(현재 스피드)

```

```

Car c2=new Car("제네시스2", "흰색"); //생성자, 멤버 변수 초기화를 위해서 사용함
c2.speed=50;
c2.getSpeed();
System.out.println(c2.carColor);

```

```

Car c3=new Car("Black");
System.out.println(c3.carColor);
c1.infor();
c2.infor();
c3.infor();

```

```

}
}

```

디폴트 생성자란?

디폴트 생성자(Default Constructor)는 클래스 내에 명시적으로 정의되지 않았을 때, 자바 컴파일러에 의해 자동으로 생성되는 생성자를 말합니다. 디폴트 생성자는 매개변수가 없고, 아무런 동작도 수행하지 않는 기본 생성자입니다.

클래스에 다른 생성자가 정의되어 있지 않은 경우, 상속 시 부모 클래스에 디폴트 생성자가 존재하는 경우 사용을 한다.

디폴트 생성자는 클래스 이름과 동일한 이름을 가지고 있다.

디폴트 생성자는 클래스의 멤버 변수를 초기화하거나 다른 초기화 작업을 수행할 수 있다.

멤버 변수란?

멤버 변수(Member Variables)는 클래스 내부에 선언된 변수로, 해당 클래스의 모든 인스턴스에서 공유됩니다. 멤버 변수는 클래스의 속성이나 상태를 나타내며, 객체의 상태를 유지하고 객체 간 데이터 공유를 가능하게 합니다.

```
package com.day04;
```

```
public class Bank {
```

```

    String name; //이름
    int money; //잔액

```

```

/*
 * 한 계좌는 이름(name), 잔액(money)
 * 입금(inputMoney)
 * 출금(outMoney)
 * 잔액확인(getMoney)
 */

    public void name() {

    }

    public void money () {

    }

    public void inputMoney(int money) {
        this.money += money;
    }

    public void outMoney(int money) {
        if(money > this.money) {
            System.out.println(name+"님의 잔액이 부족합니다.");
            return; // 종료의 의미, if~else로 사용해도 가능
        }
        this.money -=money;
    }

/*
 * if(money > this.money) {
 *     System.out.println(name+"님의 잔액이 부족합니다.");
 * }
 * else this.money -=money;
 */

    public void getMoney() {
        System.out.println(name+"님의 잔액은 "+money+"원 입니다.");
    }

    public Bank() {

    }

    public Bank(String name) {
        this.name=name;
    }

    public Bank(String name, int money) {
        this.name=name;
        this.money=money;
    }

```

```

}

public static void main(String[] args) {
    Bank b1=new Bank();
    b1.name="홍길동";
    b1.inputMoney(5000);
    b1.getMoney();
    b1.outMoney(7000);
    b1.getMoney();

    //b2 객체생성, 이름 : 강감찬, 4000원 입금 3000원 출금, 잔액 확인
    Bank b2=new Bank();
    b2.name="강감찬";
    b2.inputMoney(4000);
    b2.getMoney();
    b2.outMoney(3000);
    b2.getMoney();

    Bank b3 = new Bank("이순신");
    b3.getMoney();
    Bank b4 = new Bank("홍길순", 5000);
    b4.getMoney();

}
}

```

<Method Overloading>

```

package com.day04;

public class Sachik2 {
    int a;
    int b;
    public Sachik2() {

    }
    public Sachik2(int a, int b) {
        this.a=a;
        this.b=b;
    }

    public void sum(){
        System.out.println("더하기 : " +(a+b));
    }

    // Method Overloading
    public void sum(int num1, int num2) {
        System.out.println("더하기 : " +(num1+num2));
    }
}

```

```

public void sub(){
    System.out.println("빼기 : " +(a-b));
}

public void sub(int num1, int num2) {
    System.out.println("빼기 : " +(num1-num2));
}

public void mul() {
    System.out.println("곱하기 : " +(a*b));
}

public void mul(int num1, int num2) {
    System.out.println("곱하기 : " +(num1*num2));
}

public void divide() {
    System.out.println("나누기 : " +(a/b));
}

    public void div(int num1, int num2) {
        System.out.println("나누기 : " +(num1/num2));
    }

public int div() {
    return a/b; // return 선언할때 자료형 값을 명시해줘야 함.
}

public static void main(String[] args) {
    Sachik2 sa1 = new Sachik2(20,10);
    sa1.sum();
    sa1.sub();
    sa1.mul();
    sa1.divide();

    Sachik2 sa2 = new Sachik2(56,8);
    sa2.sum();
    System.out.println("나누기 : "+ sa2.div());

    //Method Overloading 활용하여 사칙연산
    Sachik2 sa3 = new Sachik2();
    sa3.sum(5, 7);
    sa3.mul(6, 8);
    sa3.div(40, 5);
    sa3.sub(10, 9);

}
}

```


<메소드 오버로딩>

메소드 오버로딩(Method Overloading)은 같은 이름의 메소드를 여러 개 정의하고, 매개변수의 개수, 타입 또는 순서를 다르게하여 다양한 호출 방식을 지원하는 기능입니다. 메소드 오버로딩을 통해 동일한 기능을 수행하는 메소드를 하나의 이름으로 그룹화할 수 있으며, 코드의 가독성과 유지보수성을 높일 수 있습니다.

```
package com.day04;
```

```
public class SaraMain {
```

```
    public void name() {
```

```
    }
```

```
    public void addr() {
```

```
    }
```

```
    public void tel() {
```

```
    }
```

```
    public void showInfo() {
```

```
    }
```

```
    public static void main(String[] args) {
```

```
        Saram s1 = new Saram();
```

```
        s1.name="홍길동";
```

```
        s1.addr="부산";
```

```
        s1.tel="010-1111-2222";
```

```
        s1.showInfo(); // 홍길동의 주소는 부산, 전화번호는 tel이 나오도록 하세요.
```

```
        Saram s2 = new Saram("이순신","서울","010-2222-3333");
```

```
        s2.showInfo(); // 이순신의 주소는 서울, 전화번호가 나오도록 하세요.
```

```
    }
```

```
}
```

```
package com.day04;
```

```
public class Saram {
```

```
    String name;
```

```
    String addr;
```

```
    String tel;
```

```
    public void showInfo() {
```

```
        System.out.println(name+"의 주소는 "+addr+" 입니다.");
```

```
    }
```

```
public Saram() {  
  
}
```

```
public Saram(String name, String addr, String tel) {  
    this.name=name;  
    this.addr=addr;  
    this.tel=tel;  
  
}  
}
```

// 파일을 둘로 나눠서 클래스 지정을 할 수 있다.

```
package com.day04;
```

```
public class Student {  
    String name;  
    int kor;  
    int math;  
  
    public Student(String name, int kor, int math) {  
        this.math=math;  
        this.kor=kor;  
        this.name=name;  
    }  
  
    public int getTotal() {  
        return math+kor;  
    }  
  
    public float getAvg() {  
        return getTotal()/2.0f;  
    }  
  
}
```

```
public static void main(String[] args) {  
    Student s1 = new Student("홍길동", 100, 85);  
    Student s2 = new Student("이순신", 50, 70);  
    System.out.println("이름 : " +s1.name);  
    System.out.println("국어 : " +s1.kor);  
    System.out.println("수학 : " +s1.math);  
    System.out.println("총점 : " +s1.getTotal());  
    System.out.println("평균 : " +s1.getAvg());  
    System.out.println();  
    System.out.println("이름 : " +s2.name);  
    System.out.println("국어 : " +s2.kor);  
    System.out.println("수학 : " +s2.math);  
    System.out.println("총점 : " +s2.getTotal());  
    System.out.println("평균 : " +s2.getAvg());  
}
```

```
}  
}
```

```
package com.day04;
```

```
import java.util.Scanner;
```

```
class Triangle { // 같은 파일 내에 클래스 생성
```

```
    double bottom;
```

```
    double height;
```

```
    public Triangle(double bottom, double height) {
```

```
        this.bottom = bottom;
```

```
        this.height = height;
```

```
    }
```

```
    public double getArea() {
```

```
        return bottom * height * 0.5;
```

```
    }
```

```
    public void setBottom(double bottom) {
```

```
        this.bottom=bottom;
```

```
    }
```

```
    public void setHeight(double height) {
```

```
        this.height=height;
```

```
    }
```

```
}
```

```
public class TriangleMain {
```

```
    public static void main(String[] args) {
```

```
        /* Triangle tr1 = new Triangle(10.2, 17.3);
```

```
        System.out.println("삼각형의 넓이 : " + tr1.getArea());
```

```
        tr1.setBottom(7.5);
```

```
        tr1.setHeight(11.2);
```

```
        System.out.println("삼각형의 넓이 : " + tr1.getArea()); */
```

```
        // 밑변과 높이를 입력 받아서 삼각형 넓이 구하기
```

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

```
        System.out.println("밑변과 높이를 입력 하세요.");
```

```
        double b=sc.nextDouble();
```

```
        double h=sc.nextDouble();
```

```
        Triangle tr2 = new Triangle(b, h);
```

```
        /* Triangle tr2;
```

```
        * tr2=new Triangle(b, h);은 같은표현이다. */
```

```
        System.out.println("삼각형의 넓이 : " + tr2.getArea() + "입니다.");
```

```
        // System.out.println("삼각형의 넓이 : " + (b*h*0.5) + "입니다.");
```

```
    }  
}
```

<변수 set과 get의 의미>

set : 값을 바꿀 때

get : 값을 가져올 때 사용

<this에 대한 설명 - 예약어>

```
package com.day04;
```

```
// p173
```

```
class Person1{  
    String name;  
    int age;  
    public Person1() {  
        this("이름없음", 1);  
    }  
    public Person1(String name, int age) {  
        this.name = name;  
        this.age = age;  
    }  
    public Person1 returnItSelf() {  
        return this;  
    }  
}  
  
public class CallAnotherConst {  
    public static void main(String[] args) {  
        Person1 noName = new Person1();  
        System.out.println(noName.name); //이름없음  
        System.out.println(noName.age); // 1  
        Person1 p = noName.returnItSelf();  
        System.out.println("p : " + p);  
        System.out.println(noName);  
    }  
}
```

```
package com.day04;
```

```
class TV{  
    String maker;  
    int year;  
    int inch ;  
    public TV(String maker, int year,int inch) {  
        this.maker =maker;  
        this.year =year;  
    }  
}
```

```

        this.inch = inch;
    }
    public TV(String maker, int year) {
        this.maker =maker;
        this.year =year;
    }
    public void show() {
        if(inch ==0) {
            System.out.println(maker +
                               "에서 만든 " + year + "년도형 TV" );
        }else {
            System.out.println(maker +
                               "에서 만든 " + year + "년도형 " + inch + " 인치 TV");//LG에서
만든 2021 년도형 65인치 TV
        }

    }

}

public class TVMain {

    public static void main(String[] args) {
        TV tv1 = new TV("LG", 2021, 65);
        tv1.show(); //LG에서 만든 2021 년도형 65인치 TV

        TV tv2 = new TV("삼성", 2020);
        tv2.show(); //삼성에서 만든 2020 년도형 TV

    }

}

```