Examples

Week 11

기본타입 인자 전달 (인자의 값을 매개변수에 복사)

```
public class ValuePassing
{
    public static void main( String[] args )
    {
        int n = 10;

        increase( n );

        System.out.println( n );
    }

    static void increase( int m )
    {
        m = m + 1;
    }
}
```

10

객체 인자 전달 (레퍼런스를 매개변수에 복사)

```
public class ReferencePassing
{
    public static void main( String[] args ) {
        Circle pizza = new Circle( 10 );
        increase( pizza );
        System.out.println( pizza.radius );
    }

    static void increase( Circle m ) {
        m.radius++;
    }
}

class Circle
{
    int radius;
    public Circle( int r ) { radius = r; }
    double getArea() { return 3.14*radius*radius; }
}
```

```
11
```

배열 인자 전달 (레퍼런스를 매개변수에 복사)

```
public class ArrayPassing
{
    public static void main( String[] args )
    {
        int a[] = {1, 2, 3, 4, 5};

        increase( a );

        for(int i = 0; i < a.length; i++)
            System.out.print( a[i] + " " );
    }

    static void increase( int[] array )
    {
        for( int i = 0; i < array.length; i++ )
            array[i]++;
    }
}
```

```
2 3 4 5 6
```

배열 인자 전달 (레퍼런스를 매개변수에 복사)

```
public class ArrayParameterEx
{
    static void replaceSpace(char a[]) {
        for (int i = 0; i < a.length; i++)
            if (a[i] == ' ')
                a[i] = ',';
    }
    static void printCharArray(char a[]) {
        for (int i = 0; i < a.length; i++)
            System.out.print(a[i]);
        System.out.println();
    }
    public static void main (String args[]) {
        char c[] = {'T','h','i','s',' ','i','s',' ','a',' ',
                                    'p','e','n','c','i','l','.'};
        printCharArray(c);
        replaceSpace(c);
        printCharArray(c);
    }
}
```

```
This is a pencil.
This,is,a,pencil.
```

메소드 오버로딩 (매개변수 개수와 타입으로 구분)

```
public class MethodSample
{
    public int getSum(int i, int j) { return i + j; }
    public int getSum(int i, int j , int k) { return i + j + k; }
    public double getSum(double i, double j) { return i + j; }

    public static void main( String[] args )
    {
        MethodSample a = new MethodSample();
        System.out.println( a.getSum( 1, 2 ) );
        System.out.println( a.getSum( 1, 2, 3 ) );
        System.out.println( a.getSum( 1.1, 2.2 ) );
    }
}
```

```
3
6
3.3000000000000003
```

접근지정자 (private 멤버는 외부에서 접근 불가)

```
class Sample {
    public int a;
    private int b;
    int c;
}

public class AccessEx {
    public static void main(String[] args) {
        Sample aClass = new Sample();
        aClass.a = 10;
        aClass.b = 10;
        aClass.c = 10;
    }
}
```

```
$ javac AccessEx.java
AccessEx.java:11: error: b has private access in Sample
                aClass.b = 10;
                       ^
1 error
```

TimeTest.java

```
class Time
{
    private int hour;
    private int minute;
    private int second;

    public Time() {
        this( 0, 0, 0 );
    }

    public Time(int h) {
        this(h, 0, 0);
    }

    public Time(int h, int m) {
        this(h, m, 0);
    }
```

TimeTest.java

```
    public Time(int h, int m, int s) {
        hour = ( (h >= 0 && h < 24) ? h : 0 );
        minute = ( (m >= 0 && m < 60) ? m : 0 );
        second = ( (s >= 0 && s < 60) ? s : 0 );
    }

    public Time(Time t) {
        this(t.hour, t.minute, t.second);
    }

    public String toString()     {
        return hour + ":" + minute + ":" + second;
    }
}
```

TimeTest.java

```
public class TimeTest
{
    public static void main( String[] args )
    {
        Time t1 = new Time();
        Time t2 = new Time(2);
        Time t3 = new Time(21, 45);
        Time t4 = new Time(8, 20, 25);
        Time t5 = new Time(27, 39, 82);
        Time t6 = new Time(t4);

        System.out.println( t1 );
        System.out.println( t2 );
        System.out.println( t3 );
        System.out.println( t4 );
        System.out.println( t5 );
        System.out.println( t6 );
    }
}
```

```
0:0:0
2:0:0
21:45:0
8:20:25
0:39:0
8:20:25
```

toString()

- 객체를 문자열로 표현하는 메소드
  - toString이라는 이름이 약속되어 있음

```
public String toString()     {
    return hour + ":" + minute + ":" + second;
}
```

  - 문자열이 필요한 자리에 객체가 오면, 그 객체의 toString() 메소드가 자동으로 호출됨

```
System.out.println( t1 );
→
System.out.println( t1.toString() );
```

DateTest.java

```
class Date
{
    private int month;
    private int day;
    private int year;

    public Date( int month, int day, int year )
    {
        this.month = checkMonth( month);
        this.year = year;
        this.day = checkDay( day );
    }

    private int checkMonth( int testMonth )
    {
        if ( testMonth >= 1 && testMonth <= 12 )
            return testMonth;
        else
            return 1;  // for invalid  testMonth
    }
```

DateTest.java

```
    private int checkDay( int testDay )
    {
        int[] daysPerMonth =
        { 0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

        if ( testDay > 0 && testDay <= daysPerMonth[ month ] )
            return testDay;
        if ( month==2 && testDay==29 &&
            (year%400 == 0 || (year%4 == 0 && year%100 != 0)) )
            return testDay;
        return 1;  // for invalid testDay
    }

    public String toString()
    {
        return month + "/" + day + "/" + year;
    }
}
```

DateTest.java

```
public class DateTest
{
    public static void main( String[] args )
    {
        Date date1 = new Date( 3, 1, 2018 );
        Date date2 = new Date( 2, 28, 2022 );

        System.out.println( "Entrace date:  " + date1 );
        System.out.println( "Graduation date: " + date2 );
    }
}
```

```
Entrace date:  3/1/2018
Graduation date: 2/28/2022
```

문제) Date 클래스에 increase() 구현

- DateTest.java의 Date 클래스에 아래 메소드를 추가하시오.

```
public void increase () {
        // 날짜를 하루 증가시킨다
        // 월 또는 연도가 바뀌면 적절히 처리한다.
        // 윤년인 경우에 적절히 처리한다.
}
```

- main() 메소드를 아래와 같이 수정하여 확인한다.

```
Date date2 = new Date( 1, 1, 2020 );
System.out.println( "date2:  " + date2 );
for( int i = 0; i < 800; i++) {
    date2.increase();
    System.out.println( date2 );
}
```