

5/2 스터디노트

➤ P325 MathClass

- ✓ 수학기산 관련 클래스
- ✓ 생성자가 private 이라서 다른 클래스에서 객체생성 불가, 모든 멤버가 static 이라서 다른 클래스에서 Math.멤버명 으로 접근 가능
- ✓ final 클래스이므로 다른 클래스에서 Math클래스를 상속받을 수 없고 상수멤버는 PI(원주율), E(자연로그)가 있음
- ✓ 주요 메서드

메서드	기능
abs(int a) abs(double b) abs(float f) abs(float l)	변수의 절대값 리턴
ceil(double 변수)	매개변수 보다 큰 가장 작은 수(값 올림)
floor(double 변수)	매개변수 보다 작은 가장 큰 수(값 내림)
max(int a, int b) max(double a, double b) max(float a, float b) max(long a, long b)	매개변수 두 개 중 큰 수 리턴
min(int a, int b) min(double a, double b) min(float a, float b) min(long a, long b)	매개변수 두 개 중 작은 수 리턴
random()	0.0 이상 1.0 미만의 랜덤 수 리턴 (1은 포함되지 않음)
rint(double d)	실수 d와 가장 가까운 정수값을 double로 리턴
round(double d)	반올림

✓ 예제

```
//abs 절대값 리턴
System.out.println("Math.abs(10)=" + Math.abs(10));
System.out.println("Math.abs(-10)=" + Math.abs(-10));
System.out.println("Math.abs(3.1415)=" + Math.abs(3.1415));
System.out.println("Math.abs(-3.1415)=" + Math.abs(-3.1415));

//ceil 올림
System.out.println("Math.ceil(5.4)=" + Math.ceil(5.4));
System.out.println("Math.ceil(-5.4)=" + Math.ceil(-5.4));

//floor 내림
System.out.println("Math.floor(5.4)=" + Math.floor(5.4));
System.out.println("Math.floor(-5.4)=" + Math.floor(-5.4));

//round 반올림
System.out.println("Math.round(5.4)=" + Math.round(5.4));
System.out.println("Math.round(5.5)=" + Math.round(5.5));

//rint 입력받은 실수와 가장 가까운 정수값 리턴
System.out.println("Math.rint(5.4)=" + Math.rint(5.4));
System.out.println("Math.rint(-5.4)=" + Math.rint(-5.4));

//max, min 큰/작은 수 리턴
System.out.println("Math.max(5, 4)=" + Math.max(5, 4));
System.out.println("Math.max(5.4, 5.3)=" + Math.max(5.4, 5.3));
System.out.println("Math.min(5, 4)=" + Math.min(5, 4));
System.out.println("Math.min(5.4, 5.3)=" + Math.min(5.4, 5.3));

//random 0.0 이상 1.0 미만 랜덤수 리턴
System.out.println("Math.random()" + Math.random());
```

```
Math.abs(10)=10
Math.abs(-10)=10
Math.abs(3.1415)=3.1415
Math.abs(-3.1415)=3.1415
```

```
Math.ceil(5.4)=6.0
Math.ceil(-5.4)=-5.0
```

```
Math.floor(5.4)=5.0
Math.floor(-5.4)=-6.0
```

```
Math.round(5.4)=5
Math.round(5.5)=6
```

```
Math.rint(5.4)=5.0
Math.rint(-5.4)=-5.0
```

```
Math.max(5, 4)=5
Math.max(5.4, 5.3)=5.4
Math.min(5, 4)=4
Math.min(5.4, 5.3)=5.3
```

```
Math.random()=0.14243049219078674
```

rint 와 round 의 차이 ? rint 는 반올림하되, 소수점이 0.5 일때는 짝수로 리턴.

round(1.5) = 2

rint(1.5) = 2

round(2.5) = 3

rint(2.5) = 2.

round(3.5) = 4

rint(3.5) = 4

✓ 소수점 아래자리 반올림 예제

```
/** 소수점 아래 원하는 자리에서 반올림하는 방법
 * 12.473 을 소수점 두번째자리에서 반올림 -> 12.5
 *
 * round 메서드는 1의 자리에서 반올림하므로,
 * 반올림하고싶은 소수점자리를 1의 자리로 만들어주고 round 하고 나서 다시 소수점자리 이동
 *
 * ① 반올림하고 싶은 소수점 둘째자리를 1의 자리로 만들기 ( x 100 )
 *    12.473 x 100 = 1247.3
 * ② 반올림하기 ( round )
 *    round(1247.3) = 1250
 * ③ 1의 자리를 다시 소수점 둘째자리로 만들기 ( ÷ 100 )
 *    1250 ÷ 100 = 12.5
 * */

//PI
System.out.println(Math.PI);
// 차례대로 시행
double v1 = Math.PI * 1000;
double v2 = Math.round(v1);
double v3 = v2 / 1000.0;
System.out.println(v3);
// 코드 한줄로 시행
System.out.println(Math.round(Math.PI*1000)/1000.0);
```

3.141592653589793

3.142

3.142

➤ P328 Wrapper 클래스

- ✓ 기본 자료형을 객체화 할 수 있는 클래스를 통틀어서 Wrapper 클래스라고 부름
- ✓ 대표적인 Wrapper 클래스

기본자료형	Wrapper 클래스
boolean	Boolean
byte	Byte
char	Character
short	Short
int	Integer
long	Long
float	Float
double	Double

- ✓ 기본 자료형과 Wrapper 클래스 사이에 자동 형변환 가능 (boxing)

i1객체와 i2객체는 각각 다른 객체이므로 각각의 주소값을 가짐.

```
i1==i2 : false
i1.equals(i2) : true
i4==i5 : true
i1.toString() : 10
i1==10 : true
i1==i3 : true
```

Wrapper 클래스에서 equals 를
값을 비교하는 기능으로 오버라이딩 함

생성자에 들어가는 인자는 기본 타입의 값을 사용함

```
//정수 10이 Integer 클래스 객체로 변환 (boxing)
Integer i1 = new Integer( value: 10);
Integer i2 = new Integer( value: 10);

System.out.println("i1==i2 : " +(i1==i2));
System.out.println("i1.equals(i2) : "+(i1.equals(i2)));

int i4 = 10;
int i5 = 10;

System.out.println("i4==i5 : " +(i4==i5));
System.out.println("i4.equals(i5) : "+(i4.equals(i5)));

equals는 int 기본자료형은 매개변수로 안받음(객체를 받음)

System.out.println("i1.toString() : "+i1.toString());

//i1 객체가 10 정수로 변환 (unboxing)
System.out.println("i1==10 : "+(i1==10));
int i3 = 10;
System.out.println("i1==i3 : "+(i1==i3));
```

✓ Wrapper 클래스의 상수

```
System.out.println("정수의 최대값 : "+Integer.MAX_VALUE);
System.out.println("정수의 최소값 : "+Integer.MIN_VALUE);
System.out.print("\n");
System.out.println("Byte 최대값 : "+Byte.MAX_VALUE);
System.out.println("Byte 최소값 : "+Byte.MIN_VALUE);
System.out.print("\n");
System.out.println("정수의 사이즈 : "+Integer.SIZE);
System.out.println("float 사이즈 : "+Float.SIZE);
System.out.println("double 사이즈 : "+Double.SIZE);
```

정수의 최대값 : 2147483647
정수의 최소값 : -2147483648

Byte 최대값 : 127
Byte 최소값 : -128

정수의 사이즈 : 32
float 사이즈 : 32
double 사이즈 : 64

✓ Wrapper 클래스 자동 형변환 (문자열을 숫자로)

```
String number = "100";

int i1 = Integer.parseInt(number);
int i2 = new Integer(number).intValue();
int i3 = Integer.valueOf(number);
```

문자열 자료형인 number 를
Integer 클래스의 static 메서드인 parseInt 메서드를 이용하여 문자열 -> 정수 형변환
intValue : static이 아닌 메서드로, Integer 객체에서 int형 값을 뽑아내는 메소드.
valueOf : String 을 parseInt 하여 리턴받은 int 타입의 데이터를 Integer클래스의 객체로 리턴함

문자열 ⇒ 기본자료형	문자열 ⇒ Wrapper 클래스 객체
byte b = Byte.parseByte("10");	Byte b = Byte.valueOf("10");
short s = Short.parseShort("10");	Short s = Short.valueOf("10");
int i = Integer.parseInt("10");	Integer i = Integer.valueOf("10");
long l = Long.parseLong("10");	Long l = Long.valueOf("10");
float f = Float.parseFloat("10.5");	Float f = Float.valueOf("10.5");
double d = Double.parseDouble("10.5");	Double d = Double.valueOf("10.5");

JDK 버전 9 이후

Integer 클래스 객체를 생성할 때
기존 생성자로 생성하는 것보다
valueOf 사용하는 것을 권장함

✓ 형변환, 연산

```
int i = 10;
int i2 = 10;
int i3 = 10;

//기본형(i, i2, i3) 을 참조형 (intg, intg2, intg3) 으로 변환
Integer intg = (Integer)i;
Integer intg2 = Integer.valueOf(i2);
Integer intg3 = i3; // 변환을 생략해도 됨

//참조형을 기본형으로 변환
int i4 = (int)intg3; //참조형을 기본형으로 변환 (변환생략가능)

//참조형과 기본형간 연산 (형변환 미리 할 필요 없이 가능)
int i5 = intg + 10; //참조형과 기본형간 연산 가능 (형변환 미리 할 필요 없음)
System.out.println("i5 = "+i5);

//참조형간 덧셈 연산 가능
Long lng = 10L;
long l = intg2 + lng ;
System.out.println("l = "+l);
```

i5 = 20
l = 20

● P332 java.util 패키지

- ✓ 자바 프로그래밍에서 자주 사용되는 유용한 기능을 모아둔 패키지 : 클래스 앞에 import 해서 사용

```
import java.util.Random;
public class RandomEx2 {
    public static void main(String[] args) {
```

➤ P332 Random 클래스

- ✓ Random 은 데이터를 임의로 발생시켜서 특정상황에 대한 시뮬레이션을 할 때 주로 사용
- ✓ 동일한 seed 값(숫자)을 넣으면 랜덤하게 생성된 값을 재현할 수 있음 (시뮬레이션할 때 동일한 시나리오로 재현할 수 있음)

```
Random r1 = new Random( seed: 42);
Random r2 = new Random( seed: 42);

System.out.println("r1");
for (int i = 0; i < 5; i++) {
    System.out.println(i+ " = " + r1.nextInt());
}
System.out.println("r2");
for (int i = 0; i < 5; i++) {
    System.out.println(i+ " = " + r2.nextInt());
}
```

```
r1
0 = -1170105035
1 = 234785527
2 = -1360544799
3 = 205897768
4 = 1325939940
r2
0 = -1170105035
1 = 234785527
2 = -1360544799
3 = 205897768
4 = 1325939940
```

✓ 주요 메서드

메서드	기능
Random()	Random 객체를 생성하는 생성자
Random(long seed)	seed값을 매개변수로 Random 객체 생성
nextBoolean()	boolean 랜덤수 리턴
nextBytes(byte[] bytes)	byte 배열에 랜덤수를 추가
nextDouble()	double 자료형의 랜덤수 리턴
nextGaussian()	평균이 0.0 표준편차가 1인 가우시안분포의 랜덤수 리턴
nextInt()	정수 랜덤값 리턴
nextInt(int n)	0부터 n미만까지의 정수타입의 랜덤값 리턴
nextLong()	long 타입의 랜덤값 리턴
setSeed(long seed)	seed값 변경

✓ 주사위 예제

```
Random rand = new Random();

for (int i = 1; i < 5; i++) {
    System.out.println(i+"번째 주사위 던진 결과 : "+(rand.nextInt( bound: 6)+1));
}
```

1번째 주사위 던진 결과	: 3
2번째 주사위 던진 결과	: 1
3번째 주사위 던진 결과	: 4
4번째 주사위 던진 결과	: 1

nextInt(n); 는 0부터 n 미만의 정수를 랜덤으로 리턴한다.

nextInt(6); 은 0부터 6미만의 정수 범위에서 랜덤 값을 리턴 -> 0~5의 값이 출력됨
주사위는 1~6 의 숫자가 있으므로 +1 을 해서 1~6의 값이 나오도록 함

- 선생님 Random 예제 : 사람이름, 국영수 점수 랜덤으로 뽑기

```
class Person{
    private String name;    private int korean;    private int english;    private int math;

    public Person() {
        Random random = new Random();
        this.name = generateRandomKoreanName();
        this.korean = random.nextInt( bound: 101);
        this.english = random.nextInt( bound: 101);
        this.math = random.nextInt( bound: 101);
    }

    public String getName() { return name; }
    public int getKorean() { return korean; }
    public int getEnglish() { return english; }
    public int getMath() { return math; }

    public double getAverage(){ return (korean+english+math)/3.0; }

    public static String generateRandomKoreanName(){
        Random random = new Random();
        String[] familynames = {"김", "이", "박", "최", "정", "강", "조", "윤", "장", "임", "한",
        String[] givenNames1 = {"가", "나", "다", "라", "마", "바", "사", "아", "자", "차", "카",
        String[] givenNames2 = {"은", "진", "미", "현", "주", "성", "영", "지", "종", "광", "준",
        //random.nextInt([ 배열명 ].length): 배열의 인덱스 값을 랜덤하게 선택함
        String familyname = familynames[random.nextInt(familynames.length)];
        String givenName1 = givenNames1[random.nextInt(givenNames1.length)];
        String givenName2 = givenNames2[random.nextInt(givenNames2.length)];
        //랜덤하게 선택한 값을 순서대로 조합하여 완성된 이름 세글자를 리턴
        return familyname+givenName1+givenName2;
    }
}
```

→ 랜덤으로 만들어진 학생 (1 명) 뽑기

```
public class ScoreManagementSimulation {  
    public static void main(String[] args) {  
        Person p1 = new Person();  
        System.out.println(p1.getName());  
        System.out.println(p1.getKorean());  
        System.out.println(p1.getEnglish());  
        System.out.println(p1.getMath());  
    }  
}
```

허타규

93

51

87

→ 여러명의 학생 뽑기

1) person 클래스에 toString 메서드 오버라이딩

```
@Override  
public String toString() {  
    return String.format("%s\t %d\t %d\t %d\t %.1f", name, korean, english, math, getAverage());  
}
```

2) main 클래스에서 출력

```
int numStudents=60;  
Person[] students = new Person[numStudents];  
  
System.out.printf("%s\t\t%s\t%s\t%s\t%s\n", "학생", "국어", "영어", "수학", "평균점수");  
for(int i=0; i<numStudents;i++){  
    students[i] = new Person();  
    System.out.println(students[i].toString());  
}
```

학생	국어	영어	수학	평균점수
홍자현	70	10	73	51.0
이자기	24	72	85	60.3
김차성	49	84	81	71.3
박타태	93	29	20	47.3
송나동	76	72	85	77.7
전가현	37	46	50	44.3
양마현	34	47	39	40.0
조자수	21	30	47	32.7
정나성	69	23	0	30.7

- 예제 2 : 학급 평균 뽑기

```
public static void main(String[] args) {  
    int numStudents=60;  
    Person[] students = new Person[numStudents];  
  
    for (int i = 0; i < numStudents; i++) {  
        students[i] = new Person();  
    }  
  
    // 학급의 평균 점수 계산  
    double classAverage = 0;  
    for(Person student:students){  
        classAverage = classAverage+student.getAverage(); // 학생별 평균점수 누적합  
    }  
    classAverage=classAverage/numStudents; // 학급 전체 인원의 평균점수  
    System.out.println("학급의 평균 점수 : "+classAverage);  
    System.out.println("학급의 평균 점수 소수점 첫째자리까지 반올림 : "+((Math.round(classAverage*10))/10.0));  
  
    System.out.println("프로그램 종료");  
}
```

학급의 평균 점수 : 47.28333333333334
학급의 평균 점수 소수점 첫째자리까지 반올림 : 47.3
프로그램 종료

- 예제 3 : 최고 평균 점수 학생 찾기

```
public static void main(String[] args) {  
    int numStudents=60;  
    Person[] students = new Person[numStudents];  
  
    for (int i = 0; i < numStudents; i++) {  
        students[i] = new Person();  
    }  
  
    //최고 평균 점수를 가진 학생 찾기  
    Person topStudent = students[0];  
    for (int i = 0; i < numStudents; i++) {  
        if(students[i].getAverage()>topStudent.getAverage()){  
            topStudent=students[i];  
        }  
    }  
  
    System.out.println("최고 평균 점수를 가진 학생 정보 : "+topStudent.toString());  
  
    System.out.println("프로그램 종료");  
}
```

최고 평균 점수를 가진 학생 정보 : 임바동 75 99 92 88.7
프로그램 종료

➤ P335 Scanner 클래스

- ✓ 입력되는 문자를 읽어주는 클래스
- ✓ 사용 : Scanner 클래스의 객체 생성 후 Scanner 클래스의 메서드 사용
- ✓ next@@@() 메서드를 이용해서 입력받은 문자열을 다양한 자료형으로 변환 가능

메서드	리턴타입
nextBoolean()	boolean
nextByte()	byte
nextShort()	short
nextInt()	int
nextLong()	long
nextDouble()	double
nextFloat()	float
nextLine()	String

- ✓ 이름 입력받기 예제

```
import java.util.Scanner;
public class ScannerEx {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        int cnt=0;
        while(true){
            System.out.print("이름을 입력하세요: ");
            String name = scan.nextLine();
            if("".equals(name)) break;
            System.out.println(name+"님 안녕하세요.");
            cnt++;
        }
        System.out.println("총 입력된 회원 수: "+cnt);
    }
}
```

"" 빈문자열입력하면 break(반복종료)

➔ 문자 입력하지않고 엔터입력

이름을 입력하세요: 홍길동
홍길동님 안녕하세요.
이름을 입력하세요: 이순신
이순신님 안녕하세요.
이름을 입력하세요: 임꺽정
임꺽정님 안녕하세요.
이름을 입력하세요:
총 입력된 회원 수: 3

➤ P336 Date 클래스

- ✓ 자바 1버전때 나온 클래스로 지금은 사용 X. 하위 호환을 위해서만 사용

```
import java.util.Date;
import java.text.SimpleDateFormat;
public class DateEx {
    public static void main(String[] args) {
        Date now = new Date();
        System.out.println(now);

        SimpleDateFormat sf = new SimpleDateFormat(pattern: "yyyy-MM-dd HH:mm:ss E a");
        System.out.println(sf.format(now));
    }
}
```

Tue May 02 14:53:53 KST 2023
2023-05-02 14:53:53 화 오후

yyyy-MM-dd HH:mm:ss E a

연도 - 월 -일 시간 : 분 :초 요일 오전/오후

➤ P337 Calendar 클래스

- ✓ Date 클래스보다는 나중에 추가된 클래스지만 최근에는 사용하지 않는 클래스로 하위 호환을 위해 사용함
- ✓ 추상클래스이며 생성자 접근제한자가 protected 라서 외부에서 new 연산자 사용 불가하며
- ✓ getInstance() 메서드를 통해 객체 생성하거나, Calendar 클래스를 상속받는 GregorianCalendar 클래스를 이용해서 생성한 객체를 사용해야함

- ✓ 주요메서드
- ✓ 날짜 관련
 1. get(Calendar.YEAR) : 연도
 2. get(Calendar.MONTH) : 월 (0부터 시작함, 0~11)
 3. get(Calendar.DATE) : 일
 4. get(Calendar.DAY_OF_YEAR) : 해당 연도의 첫날(1월 1일) ~ 현재일까지의 일수
 5. get(Calendar.DAY_OF_MONTH) : 해당월의 첫날(1일) ~ 현재일까지의 일수
 6. get(Calendar.DAY_OF_WEEK) : 해당주의 첫날 ~ 현재일까지의 일수 (일,월,화,수,목,금,토 순)
 7. get(Calendar.DAY_OF_WEEK_IN_MONTH) : 현재요일에 대해 해당 월의 첫 번째 요일 ~ 현재 요일의 수
 8. get(Calendar.WEEK_OF_YEAR) : 올해의 현재주차
 9. get(Calendar.WEEK_OF_MONTH) : 이번달의 현재주차

- ✓ 시간 관련
 1. get(Calendar.HOUR) : 시간 (12시간제, 0 ~ 11로 표시됨)
 2. get(Calendar.HOUR_OF_DAY) : 시간 (24시간제, 0 ~ 23으로 표시됨)
 3. get(Calendar.MINUTE) : 분
 4. get(Calendar.SECOND) : 초
 5. get(Calendar.MILLISECOND) : 1/1000초
 6. get(Calendar.AM_PM) : 오전/오후 (오전:0, 오후:1)

● (교재X) LocalDate

- ✓ 최근 날짜, 시간 관련 처리는 Date, Calendar, GregorianCalendar 클래스 보다 LocalDate, LocalTime, LocalDateTime을 주로 사용함.
- ✓ LocalDate 날짜 처리 / LocalTime 시간 처리 / LocalDateTime 날짜+시간 처리

```
import java.time.LocalDate;
public class LocalDateTest {
    public static void main(String[] args) {
        // 현재날짜
        LocalDate today = LocalDate.now();          System.out.println("현재 날짜: "+today);

        // 날짜 구성 요소
        int year = today.getYear();
        int month = today.getMonthValue();
        int day = today.getDayOfMonth();             System.out.printf("오늘은 %d년 %d월 %d일입니다.\n", year, month, day);

        // 날짜 계산
        LocalDate nextMonth = today.plusMonths(1);   System.out.println("다음달: "+nextMonth);
        LocalDate nextYear = today.plusYears(1);     System.out.println("내년: "+nextYear);
        LocalDate tenDaysAgo = today.minusDays(10);  System.out.println("10일 전: "+tenDaysAgo);

        // 날짜 비교
        LocalDate anotherDate = LocalDate.of(2023, 6, 15);
        if (today.isBefore(anotherDate)) { System.out.println("오늘은 "+anotherDate+" 이전입니다."); }
        else if (today.isAfter(anotherDate)) { System.out.println("오늘은 "+anotherDate+" 이후입니다."); }
        else { System.out.println("오늘은 " + anotherDate + " 입니다."); }
    }
}
```

현재 날짜: 2023-05-02

오늘은 2023년 5월 2일입니다.

다음달: 2023-06-02
내년: 2024-05-02
10일 전: 2023-04-22

오늘은 2023-06-15 이전입니다.

- (교재X) LocalTime

import java.time.LocalTime;

// 현재시간

```
LocalTime now = LocalTime.now();          System.out.println("현재 시간: "+now);
```

현재 시간: 16:03:44.068349100

// 시간 구성요소

```
int hour = now.getHour();  
int minute = now.getMinute();  
int second = now.getSecond();          System.out.printf("현재 시각은 %d시 %d분 %d초 입니다.%n", hour, minute, second);
```

현재 시각은 16시 3분 44초 입니다.

// 시간 계산

```
LocalTime nextHour = now.plusHours( hoursToAdd: 1);          System.out.println("1시간 뒤: "+nextHour);  
LocalTime inTenMinutes = now.plusMinutes( minutesToAdd: 10);          System.out.println("10분 뒤: "+inTenMinutes);
```

1시간 뒤: 17:03:44.068349100

10분 뒤: 16:13:44.068349100

// 시간 비교

```
LocalTime anotherTime = LocalTime.of( hour: 20, minute: 30, second: 0);  
if (now.isBefore(anotherTime)) { System.out.println("지금은 "+anotherTime+" 이전입니다."); }  
else if (now.isAfter(anotherTime)) { System.out.println("지금은 "+anotherTime+" 이후입니다."); }  
else { System.out.println("지금은 " + anotherTime + " 입니다."); }
```

지금은 20:30 이전입니다.

● (교재X) LocalDateTime

import java.time.LocalDateTime;

// 현재 날짜와 시간

LocalDateTime now = LocalDateTime.now();

System.out.println("현재 날짜와 시간: "+now); //ISO-8601 표준포맷으로 출력됨

현재 날짜와 시간: 2023-05-02T16:05:20.964274700

// 날짜와 시간 구성요소

int year = now.getYear();

int month = now.getMonthValue();

int day = now.getDayOfMonth();

int hour = now.getHour();

int minute = now.getMinute();

int second = now.getSecond();

현재 날짜와 시간은 2023년 5월 2일 16시 5분 20초 입니다.

System.out.printf("현재 날짜와 시간은 %d년 %d월 %d일 %d시 %d분 %d초 입니다.\n", year, month, day, hour, minute, second);

// 날짜와 시간 계산

LocalDateTime nextMonth = now.plusMonths(1); System.out.println("다음달: "+nextMonth);

LocalDateTime nextYear = now.plusYears(1); System.out.println("내년: "+nextYear);

LocalDateTime tenDaysAgo = now.minusDays(10); System.out.println("10일 전: "+tenDaysAgo);

System.out.println("\n");

다음달: 2023-06-02T16:05:20.964274700

내년: 2024-05-02T16:05:20.964274700

10일 전: 2023-04-22T16:05:20.964274700

// 날짜와 시간 비교

LocalDateTime anotherDateTime = LocalDateTime.of(year: 2023, month: 6, dayOfMonth: 15, hour: 10, minute: 0);

if (now.isBefore(anotherDateTime)) { System.out.println("지금은 "+anotherDateTime+" 이전입니다."); }

else if (now.isAfter(anotherDateTime)) { System.out.println("지금은 "+anotherDateTime+" 이후입니다."); }

else { System.out.println("지금은 " + anotherDateTime + " 입니다."); }

지금은 2023-06-15T10:00 이전입니다.

- (교재X) DateTimeFormatter

```
import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
import java.util.Locale;
```

```
// 현재 날짜와 시간  
LocalDateTime now = LocalDateTime.now();  
  
// DateTimeFormatter 생성  
DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy년 M월 d일 EEEE h:mm a", Locale.KOREA);  
  
// 포맷 적용하여 문자열로 변환  
String formattedDateTime = now.format(formatter);  
System.out.println(formattedDateTime);
```

2023년 5월 2일 화요일 4:07 오후