

제 35강

Scanner, BigDecimal, formatting

교재:p199~201, 218~220

목차

1. 기타 클래스

1. Scanner
2. BigDecimal
3. formatting

1. java.util.Scanner 클래스 p199~201

Scanner 클래스

: 문자 데이터를 읽어오는데 도움을 주는 클래스

<사용방법>

```
Scanner 변수 = new Scanner(System.in); // 입출력 스트림과의 연결  
변수.입력메서드( );
```

1. java.util.Scanner 클래스 p199~201

Scanner 클래스

<사용 메서드>

| 메서드 | 의미 |
|----------------|--------------------------|
| nextBoolean() | 논리값으로 입력 |
| nextByte() | byte 값으로 입력 |
| nextDouble() | double 값으로 입력 |
| nextFloat() | float 값으로 입력 |
| nextInt() | int 값으로 입력 |
| nextLine() | enter key 입력 전까지 문자열로 입력 |
| next() | 공백 전까지 문자열로 입력 |

1. java.util.Scanner 클래스 p199~201

<실습> Exam-76.java

```
import java.util.Scanner;

public class Scanner1 {
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        System.out.println("이름을 입력하세요.");
        String name= sc.nextLine();
        System.out.println(name+"님 환영합니다.");
    }
}
```

<실행 결과>

```
이름을 입력하세요.  
홍길동  
홍길동님 환영합니다.
```

1. java.util.Scanner 클래스 p199~201

<실습> Exam-77.java

```
public class Prac {  
    public static void main(String[] args) {  
        try {  
            FileInputStream fis = new FileInputStream("src/day35/sample.txt");  
            Scanner s = new Scanner(fis);  
            while(s.hasNext()) {  
                System.out.println(s.nextLine());  
            }  
        }  
        catch(Exception e){  
            e.printStackTrace();  
        }  
    }  
}
```

<sample.txt>

```
sample.txt ✖  
1 hello  
2 second line  
3 third line
```

<경로>

```
src  
└─ day35  
    ├── Prac.java  
    ├── Scanner1.java  
    └─ sample.txt
```

<실행 결과>

```
hello  
second line  
third line
```

2. BigDecimal 클래스 p218

BigDecimal 클래스?

: 정확한 소수점 연산을 해야할 때 사용

목적?

: double, long으로 소수점 연산 시 오차가 나는 부분을 보완

- java.math 패키지내 클래스
- 생성자: BigDecimal(문자열)

2. BigDecimal 클래스 p218

<실습> Exam-87.java

double로 계산한 결과와 BigDecimal로 계산한 결과 비교

```
import java.math.BigDecimal;

public class BigDecimal1 {
    public static void main(String[] args) {
        double a=24.3953;
        double b=50.343998;
        System.out.println(a+b);

        BigDecimal number1= new BigDecimal(String.valueOf(a));
        BigDecimal number2= new BigDecimal(String.valueOf(b));
        System.out.println(number1.add(number2));
    }
}
```

<실행 결과>

| |
|-------------------|
| 74.73929799999999 |
| 74.739298 |

double 형 변수를 문자열로 바꿔서
BigDecimal 생성자에 전달

3. formatting p209~210

formatting?

: 데이터의 형식을 지정하는 것을 의미

[1] DecimalFormat : 숫자를 형식화 하는 클래스

| 기호 | 의미 | 기호 | 의미 |
|----|------------|----|--------|
| # | 있으면 출력 | , | 콤마를 넣음 |
| 0 | 없으면 0으로 채움 | E | 지수 기호 |
| . | 소수점 | % | 퍼센트 |
| - | 음수 기호를 붙임 | | |

3. formatting p209~210

<실습> Exam-88.java

```
public class Format1 {
    public static void main(String[] args) {
        String [] pattern= {
            "###.###",
            "000.000",
            "-###.###",
            "000000.00%"};

        //형식화 전 배열
        double [] arr= {1.3,3.33,124.243,242};

        for(int p=0;p<pattern.length;p++) {
            DecimalFormat d = new DecimalFormat(pattern[p]);
            System.out.println("<<<<"+pattern[p]+">>>>");
            for(int i=0;i<arr.length;i++) {
                //          형식화 후 배열
                System.out.println(d.format(arr[i]));
            }
        }
    }
}
```

<실행 결과>

```
<<<<###.###>>>>
1.3
3.33
124.243
242
<<<<000.000>>>>
001.300
003.330
124.243
242.000
<<<<-###.###>>>>
-1.3
-3.33
-124.243
-242
<<<<000000.00%>>>>
000130.00%
000333.00%
012424.30%
024200.00%
```