

3주차 과제

강제 타입변환에서 작은 크기=(작은 크기 타입)큰 크기 타입 의 코드를 작성하면 대부분 해결되지만, 참조형인 String 타입의 경우 다른 방법을 사용해야 타입 변환이 가능하다

String 타입 변환

자료형 클래스 parse() 메소드 사용하는 방법(String→)

String → int

`Integer.parseInt(str값)`

String → double

`Double.parseDouble(str값)`

이외에도 `byte short long float` 사용가능

Ex)

```
String a = "100"
```

```
Int b = Integer.parseInt(a);
```

```
System.out.println(a+1);
```

```
System.out.println(b+1);
```

→

```
1001//문자열+int
```

```
101//int+int
```

*변환하는 문자열이 정수로 이루어지지 않은 경우 `NumberFormatException` 에러가 발생

자료형 클래스의 toString()메소드 사용하는 방법(→String)

int→String

Integer.toString(int값)

double→String

Double.toString(double값)

이외에도 모든 기본 타입이 사용 가능

Ex)

```
double a = 1.1;
```

```
String b = Double.toString(a);
```

```
System.out.println(a+2);
```

```
System.out.println(b+2);
```

→

```
3.1 // double + int
```

```
1.12 // 문자열 + int
```

String 클래스의 valueOf() 메소드를 사용하는 방법(→String)

모든기본타입 → String

String.valueOf(값)

Ex)

```
Int a = 100;
```

```
String b = String.valueOf(a);
```

```
System.out.println(a+2);
```

```
System.out.println(b+2);
```

→

```
102 // int+ int
```

```
1002 // 문자열 + int
```

toString()와 valueOf()의 차이점

기본적으로 valueOf()는 toString 함수를 호출하는 방식

valueOf()는 입력받은 인자 값을 지정된 Number 객체 형으로 변환하여 반환

toString()는 객체가 가지고 있는 정보나 값들을 문자열로 만들어 반환

문자열로 변환되는 값이 Null(아무것도 아님)일 때, valueOf()는 "null"이라는 문자열로 처리

toString()는 Null PointException라는 오류를 발생