LAB6 - Expression (II)

LAB of COMP2021 OBJECT-ORIENTED PROGRAMMING

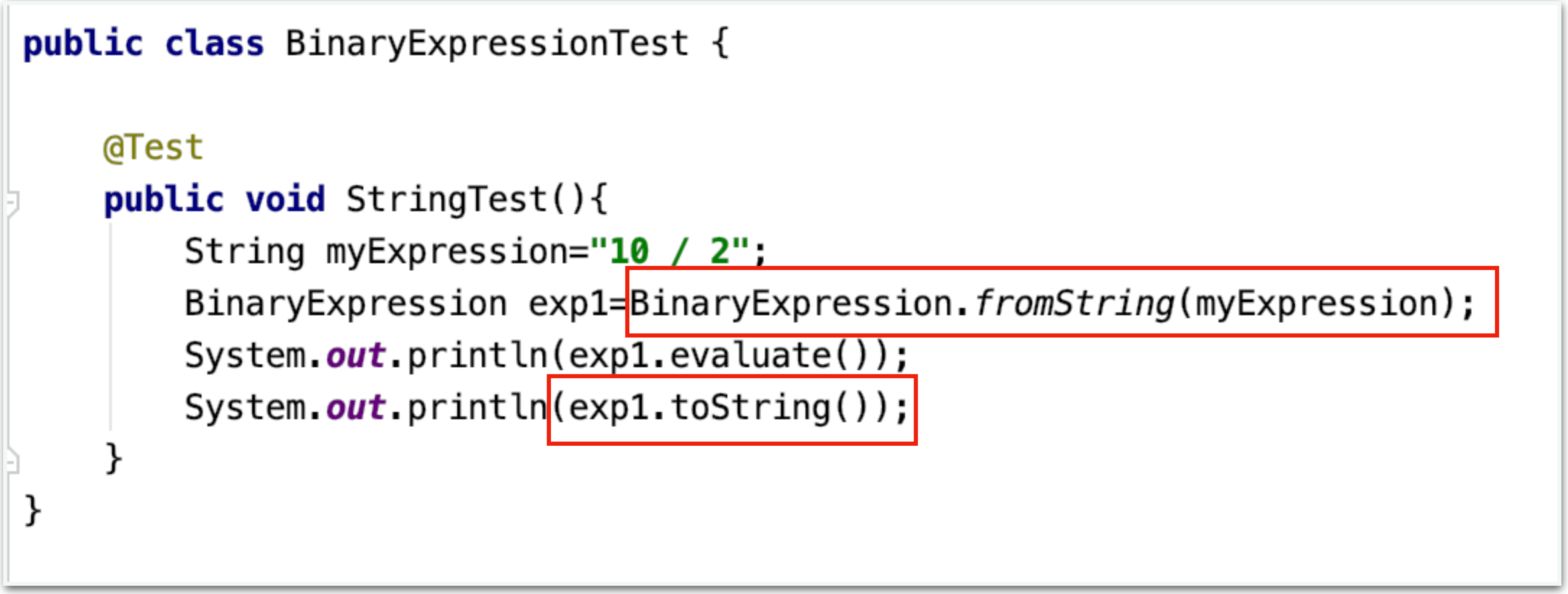
Objectives

- To enable the conversion between **binary expressions** and **strings**

- To add corresponding support for **unary expressions**

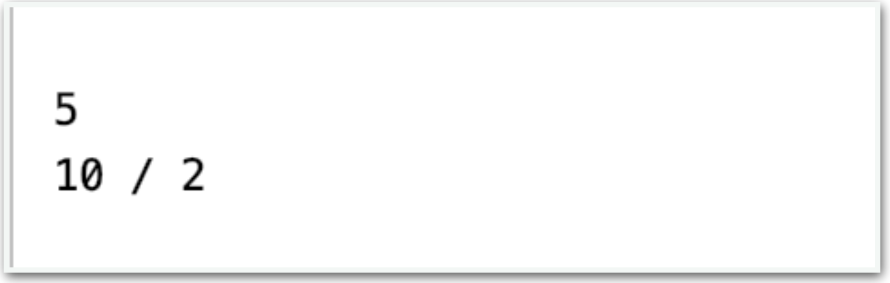
Conversion between Binary Expressions and Strings

- Example: Test method



```java
public class BinaryExpressionTest {

    @Test
    public void StringTest(){
        String myExpression="10 / 2";
        BinaryExpression exp1=BinaryExpression.fromString(myExpression);
        System.out.println(exp1.evaluate());
        System.out.println(exp1.toString());
    }
}
```

- Output



```
5
10 / 2
```

Conversion between Binary Expressions and Strings

- Assuming an operator is separated from each operand using exactly one space character;

- public String ***toString()***: return a string representation of *this* binary expression;

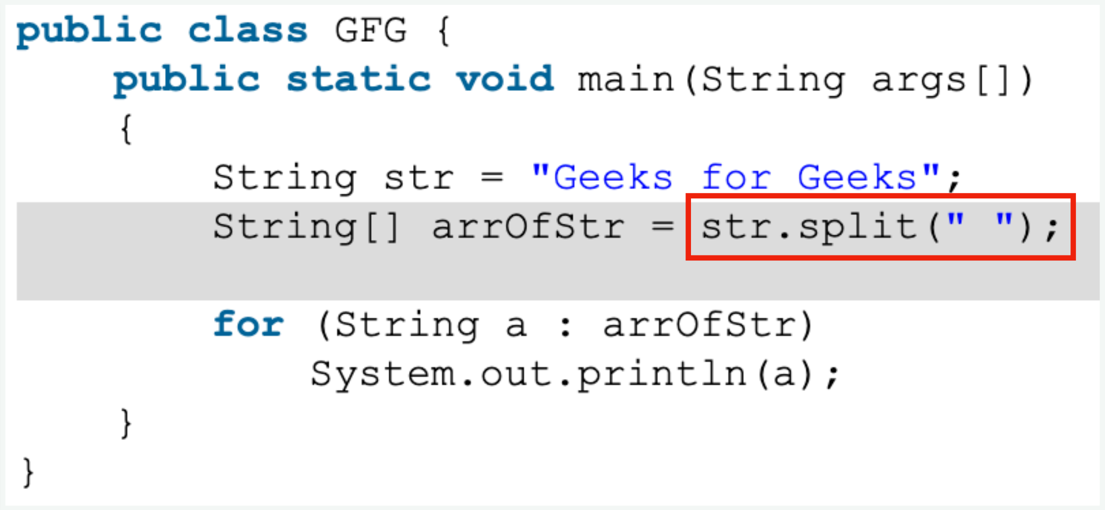
- public static BinaryExpression ***fromString(String str)***: create a new binary expression from “str”.

- Try to implement the methods yourself!

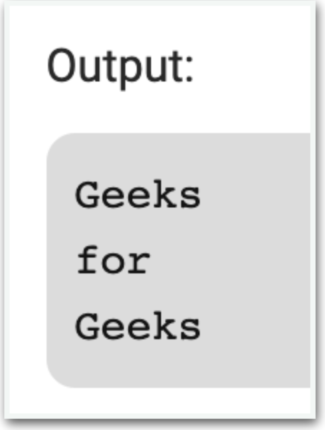
Conversion between Binary Expressions and Strings

- Useful methods from String, Examples include:

- *split()*



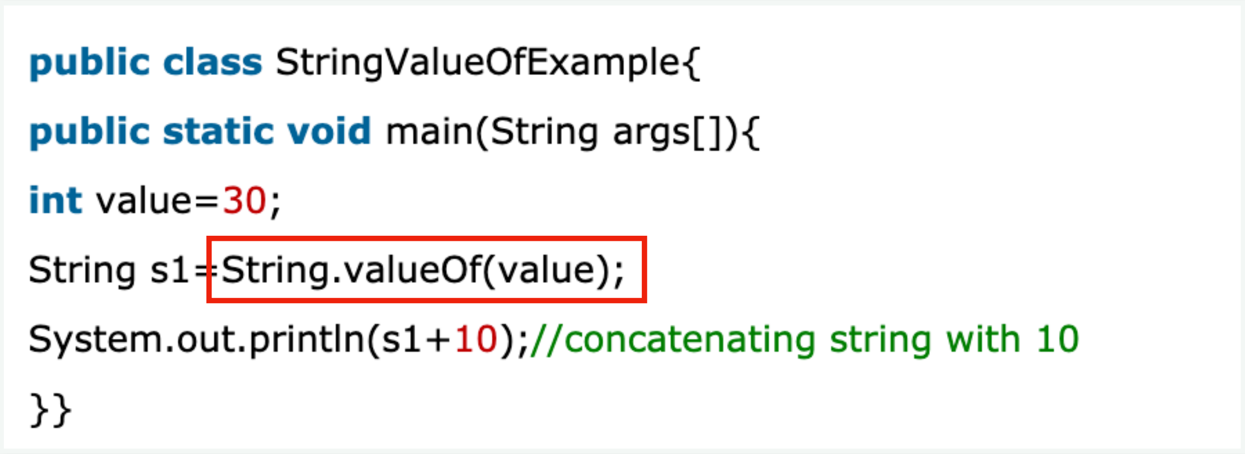
```
public class GFG {
    public static void main(String args[])
    {
        String str = "Geeks for Geeks";
        String[] arrOfStr = str.split(" ");

        for (String a : arrOfStr)
            System.out.println(a);
    }
}
```



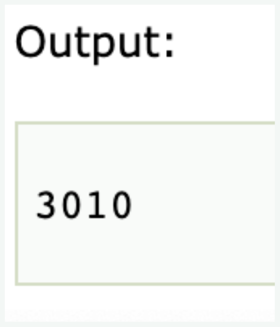
Output:

```
Geeks
for
Geeks
```

- *valueOf()*



```
public class StringValueOfExample{
public static void main(String args[]){
int value=30;
String s1=String.valueOf(value);
System.out.println(s1+10);//concatenating string with 10
}}
```



Output:

```
3010
```

UnaryExpression

- An unary operator is an operator that takes one operand
- Examples include +, -, ++, --, !, etc.
- We consider just + and - in this exercise.
- Follow classes BinaryOperator and BinaryExpression, implement UnaryOperator and UnaryExpression

- Thank you