UNIVERSITY OF TWENTE.

JML Syntax

Topic of Software Systems (TCS module 2)

Lecturer: Marieke Huisman

JML: JAVA MODELING LANGUAGE

- Widely used contract specification language
- Many tools that can handle JML specifications
  - Does the program respect the specifications?
- In this module: IntelliJML
  - Syntax and type checking of JML

EXAMPLE JML SPECIFICATION

```
public class Counter {

   /*@
     *   invariant value >= 0;
     */

   /*@  spec_public */ private int value;

   /*@  requires n >= 0;
        ensures value == n;
   */
   public Counter(int n) {
        value = n;
   }


}
```

BOOLEAN CONDITIONS

- Preconditions, postconditions and invariants are all boolean conditions
- Some special expression notation
  - `\result` refers to the return value of a method
  - `\old(`*<expr>*`)` is the value of *<expr>* before calling the method
  - Basic logical operators (`&&`, `||`, `!=`) and implication (`==>`)
  - Universal and existential quantification may be used:

    `(\forall` *<type>* `x; ...)`, `(\exists` *<type>* `x; ...)`

UNIVERSITY OF TWENTE.

SIDE-EFFECT-FREE BOOLEAN CONDITIONS

- All expressions in specifications should be free of side effects

  ~~`requires n++ >= 0;`~~ disallowed
- Specification expressions can contain method calls

  `requires getValue() >= 0;` allowed

  provided the method called does not have side effects
- This is checked separately

  `/*@ pure */ public int getValue() { .. }`