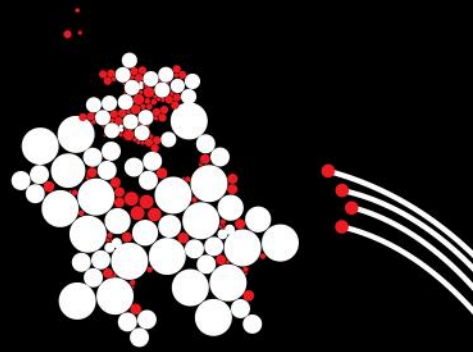
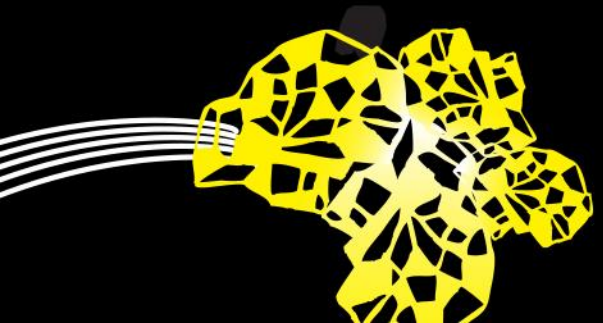
UNIVERSITY OF TWENTE.



Preconditions

Topic of Software Systems (TCS module 2)

Lecturer: Marieke Huisman



WHAT IS A PRECONDITION?

A precondition belongs to a method

- Condition that should always hold when a method is called
  - Caller has to ensure this
  - Method implementation can rely on this
- Condition expressed using method parameters and/or public methods
- Typical preconditions
  - Restriction on method arguments
  - Requiring the object to be in a particular state

COUNTER CLASS EXAMPLE

```
public class Counter {

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

  private int value;

  public Counter() {
      value = 0;
  }

  public int getValue() {
       return value;
  }

  public void setValue(int value) {
       this.value = value;
  }

  public void increment() {
       value++;
  }
}
```

Invariant: value is always positive

PRECONDITION

Question

Given the Counter invariant (value >= 0), what should be the precondition of setValue(int value)?

Answer

```
requires value >= 0;
```

PRECONDITION

RESPONSIBILITY OF THE CALLER

Suppose we add to class Counter

- A constant MAX to define the maximum value of the counter
- A method reset() that can only be called when the counter has reached MAX

Question

What should be the precondition of reset()?

Answer

```
requires getValue() == Counter.MAX;
```

ANOTHER EXAMPLE

PRECONDITION AS A RESPONSIBILITY OF THE CALLER

```
public class Lock {
    private int code;
    private boolean open;

    public Lock(int code) { }

    public boolean isOpen() { return open; }

    public void close() { open = false; }

    public void enterDigit(int digit) { }
}
```

Class invariant

invariant 0 <= code && code <= 999;

enterDigit precondition

requires 0 <= digit && digit <= 9;

- If we define this as precondition: caller must ensure this!
- Lock implementation does not have to check it!