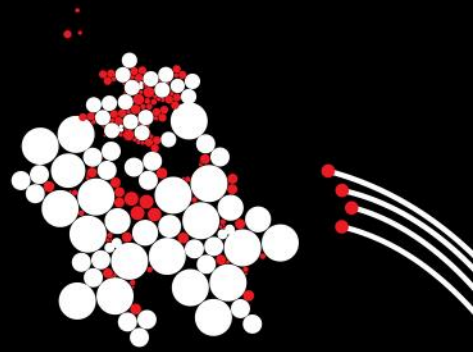
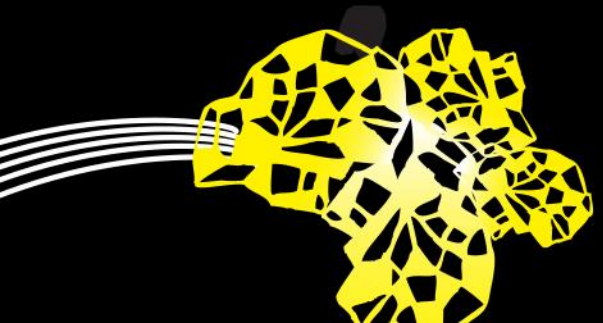
UNIVERSITY OF TWENTE.



Contracts

Topic of Software Systems (TCS module 2)

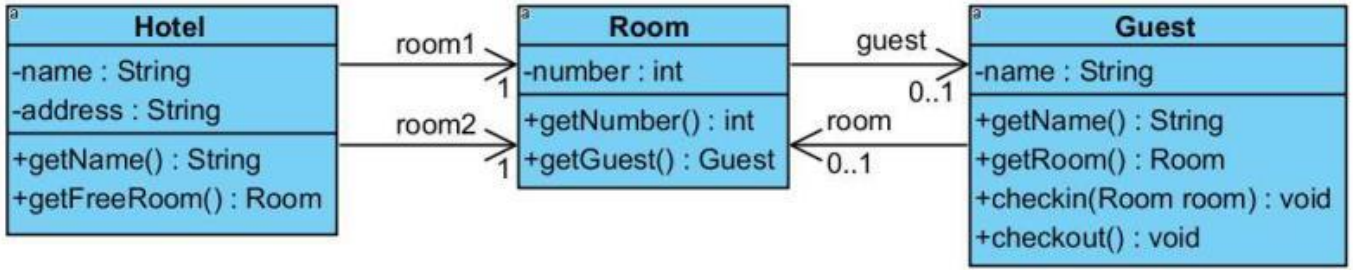
Lecturer: Marieke Huisman



PROGRAM DESIGN

- Consists of classes and their relationships
- How do they interact with each other?

Example: Hotel Information System



CLASS GUEST

FOR THIS EXAMPLE

- Instance variables: name, room
- Constructor: Guest (String name)
- Getters for name and room
- No basic setters!
- checkin (Room room): assign this guest to a room
- checkout(): remove the guest from a room

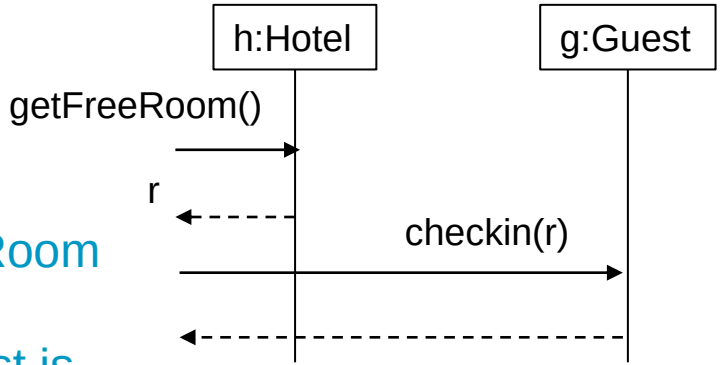

```java
/**
 * Hotel guest with a name and possibly a hotel room.
 * @author Arend Rensink
 */
public class Guest {

    private String name;
    private Room room;

    public Guest(String name) {
        this.name = name;
    }

    public String getName() {
        return this.name;
    }

    public Room getRoom() {
        return this.room;
    }

    public boolean checkin(Room room) {
        boolean result = false;
        if (this.room == null && room.getGuest() == null) {
            room.setGuest(this);
            this.room = room;
            result = true;
        }
        return result;
    }

    public boolean checkout() {
        boolean result = false;
        if (this.room != null) {
            this.room.setGuest(null);
            this.room = null;
            result = true;
        }
        return result;
    }
```

OBJECT INTERACTIONS

Assumption: all instance variables are private!

- When a program is running, objects exist that are instances of the classes defined in the program design
- Objects can only interact by calling each other's methods!

```
Hotel h = new Hotel("Fawlty Towers");
Guest g = new Guest("Major Gowen");
g.checkin(h.getFreeRoom());
```



Implicit assumptions:

- Result of getFreeRoom is an empty room
- After checkin, guest is signed in into room

CLASS SPECIFICATIONS

PRECONDITIONS, POSTCONDITIONS AND CLASS INVARIANTS

For each class and each method the program designer must specify the conditions for objects of this class to work properly!

- Preconditions: Which conditions should hold before a method is called, for it to work correctly?
- Postconditions: What conditions are satisfied once the method has finished correctly?
- Class invariants: What are the conditions that must always hold in an object of a class?

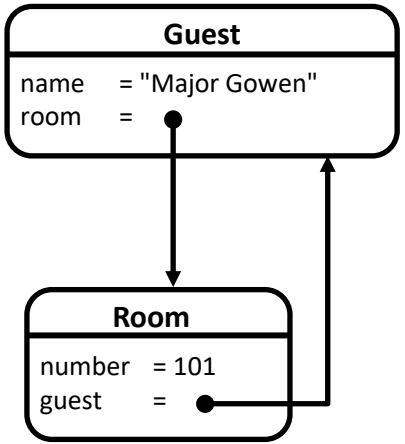
GUEST CLASS SPECIFICATION

- Method checkin(Room room)
  - Precondition: Room is empty

    (room.getGuest() == null)
  - Postcondition: Guest related to room is this guest

    (room.getGuest() == this)
- Class Invariant: If room attribute is not null then the guest related to the room is this guest

  (room != null ==> room.getGuest() == this)



CLASS SPECIFICATIONS

PRECONDITIONS, POSTCONDITIONS AND CLASS INVARIANTS

For each class and each method the program designer must specify the conditions for objects of this class to work properly!

- Preconditions: Which conditions should hold before a method is called, for it to work correctly?
- Postconditions: What conditions are satisfied once the method has finished correctly?
- Class invariants: What are the conditions that must always hold in an object of a class?