

Aspect-Oriented Software Development 2018/19

doc. Ing. Valentino Vranić, PhD., ÚISI FIIT STU

Test – November 5, 2018

The space probe information system is equipped with a communication service the probe uses to send messages to the control center on Earth and to receive commands from there. This service is being activated by instantiating the corresponding class called `CommService`:

```
public class CommService {
    ...

    public CommService() {
        ...
    }

    public void send(String message) {
        ...
    }
    ...
}
```

That is, to send a message, one would first need to create a communication service object:

```
CommService commService = new CommService();
commService.send("Houston, we have a problem.");
```

To improve reliability, an additional communication service is being added to the space probe information system. This service, represented by the `CommServiceAdd` class:

```
public class CommServiceAdd extends CommService {
    ...

    public CommServiceAdd() {
        ...
    }

    public void send(String message) {
        ...
    }
    ...
}
```

is to be activated if the original communication service fails. Provide an aspect-oriented solution to this problem.

- 1. (2 b)** Provide the corresponding analytic model in the Theme/Doc notation in the view of themes and relationships (the basic view). Transform this view into the crosscutting theme view.
- 2. (3 b)** Provide the corresponding design model in the Theme/UML notation. You may use the indirect relationships known from the JPDD notation.
- 3. (5 b)** Provide the code of the corresponding aspect in the AspectJ programming language. If appropriate, apply the Worker Object Creation, Wormhole, or Cuckoo's Egg design pattern. Explain your decision.