**Introduction to the Software Engineering – Test 4. Sequence Diagram**

**Assignment:**

1. **Draw the "Update of Modules" sequence diagram to describe the following communication:**

The *administrator* sends an *update* message to the *Application* object. This object, as part of the processing of this message, sends itself a message *get installed modules*. As a part of execution of this message the *module list* object is created. On this object, *Add Module* messageis periodically sent until modules are available. After all modules have been added, processing of the *get installed modules* message ends.

Then continues execution of *update* message on *Application* object, it sends *return a list of new modules* message to the *update manager* object. After the *update manager* object sends back the *list of new modules* to *Application* object, the *Application* object sends itself a message – *update modules*. This message is sent in the loop until new modules are available. As a part of the processing of the message *update modules*, the *give a new version of module* message is sent to *update manager* object. This message returns the new version of the module. Then, as the execution of the *update modules* message continues, it calls the *update* message on the *module* object. After returning value from the *update* message, there could be two alternatives based on the returning value:

- if the *returning value* is *failed*, the execution of the *update modules* message sends an update *after the restart* message to the *module* object

- if the *returning value* is *successful*, the execution of the *update modules* message send an *reinitialize* message to the *module* object

By merging both alternatives, the loop message *update modules* ends.

Then continues execution of *update* message, it sends destroy message to the *module list* object. And the *update* message execution on *Application* object ends by returning OK back to the *administrator*.

1. **Write in the programming language the *Application* class.**