



LAUREA
UNIVERSITY OF APPLIED SCIENCES
Together we are stronger

Designing your project application

Object-Oriented programming

Antonius Camara

Proposed architecture for the course project - OO application

Java Project in Eclipse: MyApp
Package defined under the project: myApp

MyApp.java

private Class variables
(global access within
MyApp.java)

"Constructor" MyApp()

private methods()

Calls to public methods
of other Classes in the
application

Some other UI
interaction with the user
(Dialogs, Messages)

Providing reusable
functionality to event
handlers

Event Handlers
(Nested, inner classes)

public void main()
(Starting the App)

Initializes
the UI

MyApp
GUI

(Main screen)

Responding to
user actions

Start the app
and set the
main screen
visible

SomeAppClass1.java

private Class variables

private instance variables

"Constructor"
SomeAppClass1()

public methods()

Setting object data

Getting object data

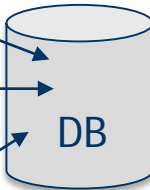
Doing some other
operations on objects

private methods()
(Reusable logic for
internal use in the Class)

JDBC access

JDBC access

JDBC access

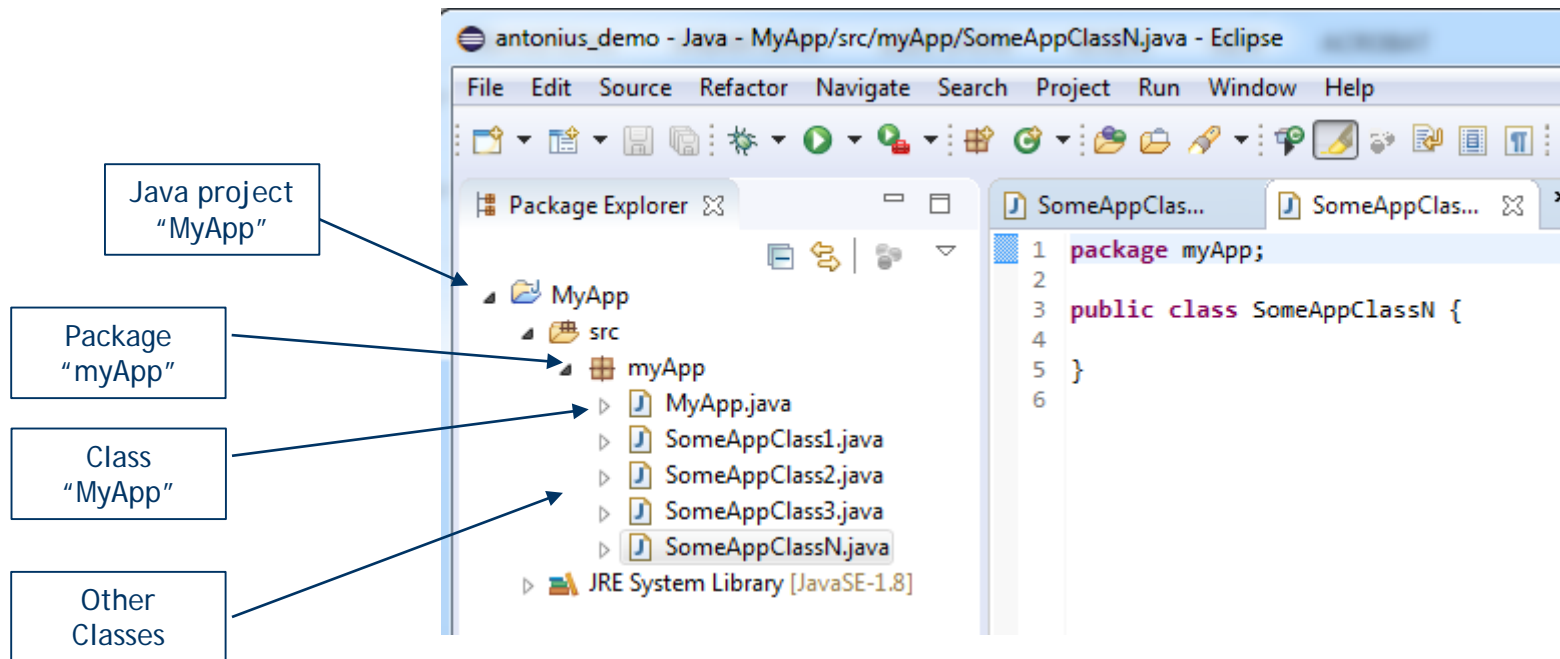


SomeAppClass2.java

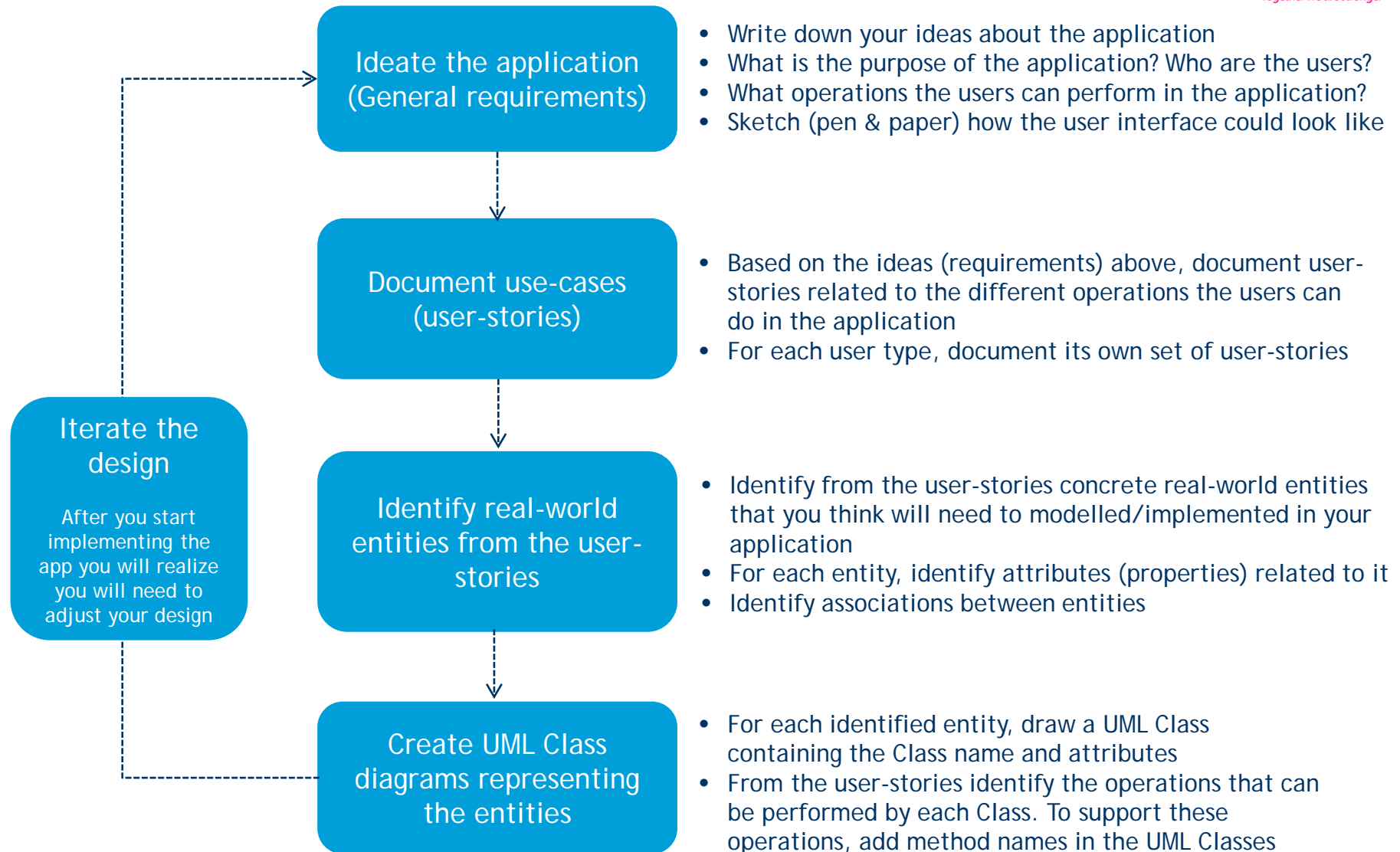
SomeAppClass3.java

SomeAppClassN.java

In Eclipse...



How to figure out which Classes are needed in your project App?



Case example: Internship tracking system

Case: General requirements

Name of the app:

Internship Tracking System

Users:

Internship coordinators at Laurea

Background, needs, purpose:

As an internship coordinator at Laurea I'm not happy with the systems the school offers to me to efficiently manage the internship process of my assigned students. To properly manage the internship process I need a new application. The application should work as an individual desktop app that I could launch in my laptop. The main user of the app would be me (an internship coordinator). In the application I would be able to register students that are starting an internship and I could track their progress from start to end. Via the application I could set a different status to the student according to the stage he is in the process. The application should also allow me to generate and store important documentation related to the internship process.

Case: User-stories for user “Internship coordinator”

#1: Registering a new student

A student that wants to initiate an internship sends me an email containing the following information about the internship: Company name, Supervisor at the company (Name), Start/End date of the internship, Workload in hours per week, Objective of the internship (student's motivation), Job description.

When I receive this email, I will check the information. If I think the information is not clear enough, or the internship is not appropriate for the student, I will require more info from the student via email. If the information is clear and I decide to accept the internship I register the student in the application in case he has not been registered earlier (student ID and name) and create for him a new internship document with the information above. At this stage I will set a status value of “accepted” to this internship

#2: Generating the placement agreement and starting the internship

After I enter the internship data into the system (student ID, status: “accepted”, company name, ...) I will use the system to automatically generate to me an internship agreement document. This can be a MS word file that I can sign and send to the student via email. The student should also sign the document, get the company supervisor's signature and return the signed document to me. Once the agreement is signed by the parties, the student can start his internship and I will update the internship status to “ongoing”

#3: Closing the internship

At the end of the internship, the student will send me via email an internship report written by him and a job certificate signed by the company's supervisor. If the documentation is according to the required criteria, I will approve the internship by changing the internship status to “completed”. The system will also allow me to upload the internship report and job certificate so I can have access to it later on if needed

#4: Tracking progress

As time goes by I will have many students registered in the system and each student may have one more internships under different statuses. To allow me to efficiently track the internships, the system should have a user interface that lists internships based on different criteria set by me. For example: List all internships of student ID xxxx, List all internships currently in status “accepted”, List all internships with a planned end date of xxxxx, etc...

Case: Identifying entities, attributes, and operations

#1: Registering a new student

A **student** that wants to initiate an **internship** sends me an email containing the following information about the internship:

Company name, Supervisor at the company (Name), Start/End date of the internship, Workload in hours per week, Objective of the internship (student's motivation), Job description.

When I receive this email, I will check the information. If I think the information is not clear enough, or the internship is not appropriate for the student, I will require more info from the student via email. If the information is clear and I decide to accept the internship I **register** the student in the application in case he has not been registered earlier (**student ID and name**) and I will enter the internship data with the information above. At this stage I will set a **status** value of "accepted" to this internship

#2: Generating the placement agreement and starting the internship

After I enter the internship data into the system (**student ID, status: "accepted", company name, ...**) I will use the system to automatically **generate** to me an **internship agreement document**. This can be a MS word file that I can sign and send to the student via email. The student should also sign the document, get the company supervisor's signature and return the signed document to me. Once the agreement is signed by the parties, the student can start his internship and I will **update the internship status** to "ongoing"

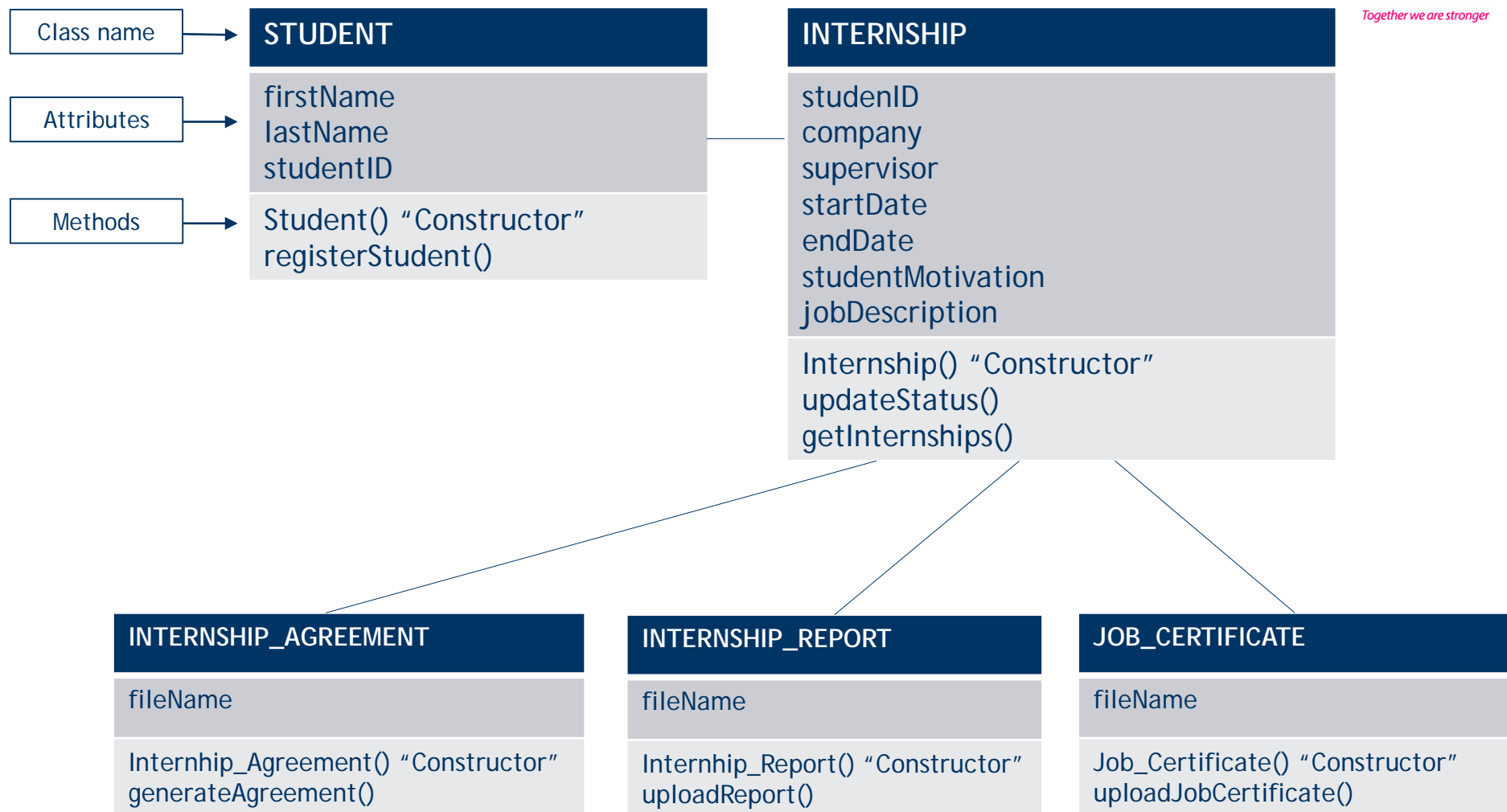
#3: Closing the internship

At the end of the internship, the student will send me via email an internship report written by him and a job certificate signed by the company's supervisor. If the documentation is according to the required criteria, I will approve the internship by changing the internship status to "completed". The system will also allow me to **upload** the **internship report** and **job certificate** so I can have access to it later on if needed

#4: Tracking progress

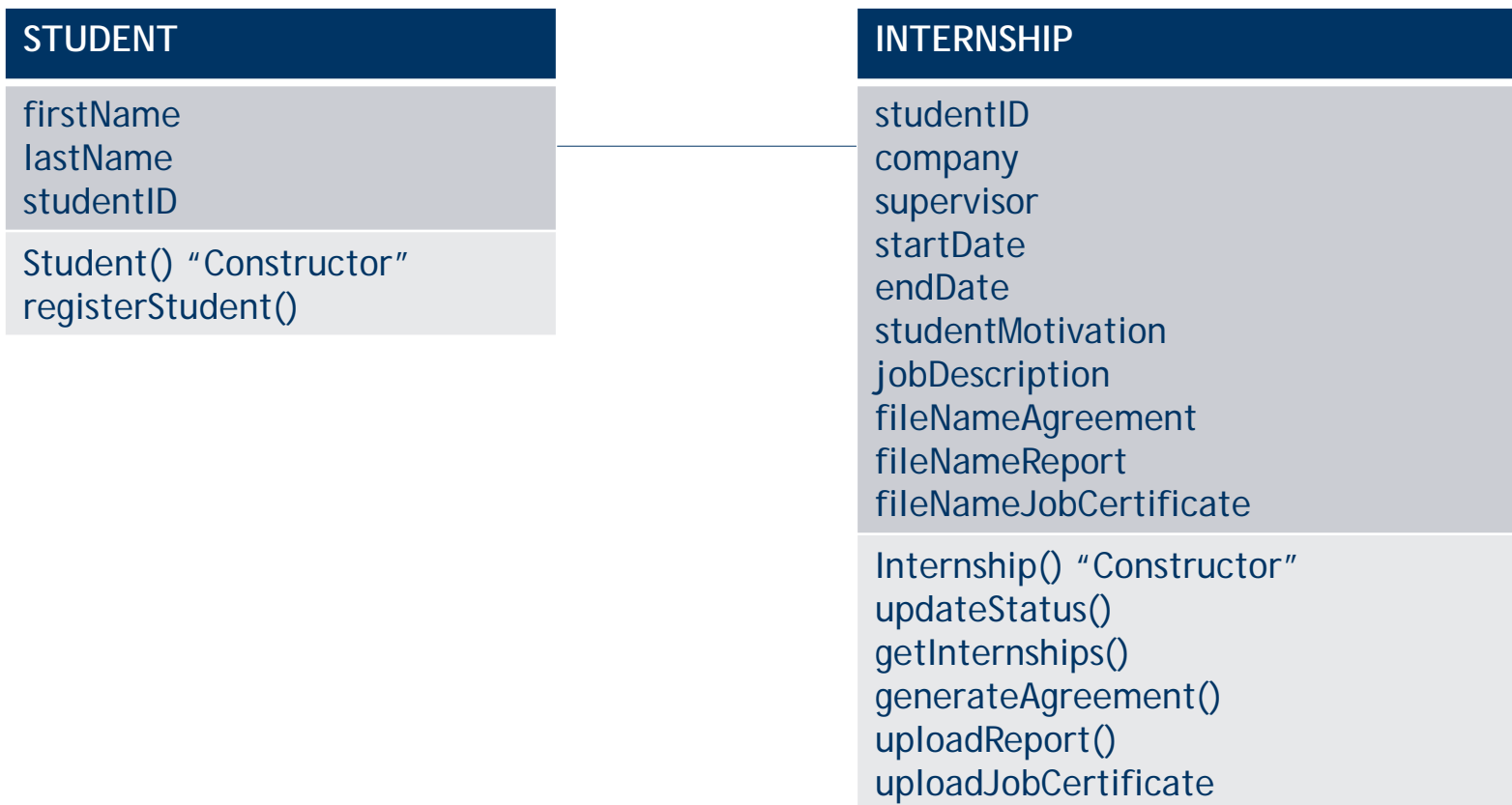
As time goes by I will have many students registered in the system and each student may have one more internships under different statuses. To allow me to efficiently track the internships, the system should have a user interface that **lists internships** based on different criteria set by me. For example: List all internships of student ID xxxx, List all internships currently in status "accepted", List all internships with a planned end date of xxxxx, etc...

Case: UML Classes (to get started)



Case: UML Classes (first iteration)

From the previous UML Class diagrams we can see that the Internship Agreement, Report and Job Certificate are closely related to the Internship object itself. A simpler design could be achieved by specifying these elements as attributes under the Internship Class (it is enough we specify the file name of these elements), and define the associated operations (generateAgreement, uploadReport, uploadJobCertificate) as methods under the Internship Class



Case: UML Classes (final format)

After you start implementing your app you will need to update your design by adding/removing classes, adding/removing attributes, etc... Once you reach a stable design, you should properly document your UML Class Diagrams:

- use (+) and (-) signs to identify public and private elements
- define attributes' data types
- define return values (data types) of methods
- define method parameters

See diagrams below. All Classes in your application should be documented.

STUDENT

- firstName: String
- lastName: String
- studentID: int

+ Student(String fName, String lName, int studentId)
+ registerStudent(int studentId): Boolean

MYAPP

+ MyApp()
+ main()

INTERNSHIP

- internshipId: int
- studentId: int
- internshipStatus: int
- company: String
- supervisor: String
- startDate: Date
- endDate: Date
- studentMotivation: String
- jobDescription: String
- fileNameAgreement: String
- fileNameReport: String
- fileNameJobCertificate: String

+ Internship(int studentId)
+ updateStatus(int internshipId, int status): Boolean
+ getInternships(int studentId): Internship[]
+ generateAgreement(int internshipId): String fileName
+ uploadReport(int internshipId, String fileName): Boolean
+ uploadJobCertificate(int internshipId, String fileName): Boolean

Case: Next steps

- ▶ Create a first version of your design specifications
 - ▶ General requirements, User-stories
 - ▶ UML Class diagrams
- ▶ Plan the application architecture according to the model proposed in slide 1
- ▶ Start planning (sketching) the user interface
- ▶ Start implementing the Classes in java
- ▶ Design and implement the relational database using the UML Class diagram as a starting point
 - ▶ Check separate PPT file about “App DB Design”
- ▶ As you keep creating your implementations you will realize that you will need to fix, change, and update your design. Keep iterating your design/implementation until you reach a stable solution

About OO software design

- ▶ This presentation provides a practical “top-down” approach to help you designing your project application
- ▶ OO software design is a large topic with many theoretical subjects
 - ▶ Data structures, Software architectures, Design principles, Design patterns, UML (all diagram types: Class, Interaction, Activity, Sequence, Object, ...)
 - ▶ Typically, these subjects are taught under different courses in Software Engineering degrees
 - ▶ Not in the scope of this course
- ▶ Related courses at Laurea
 - ▶ Ohjelmistotuotteen määrittely ja suunnittelu (5cr) (Tikkurila)