

**Object: Account**

Propose attributes for an account.

Propose operations for an account.

**Object: Account**

Attributes for an account:

attribute balance: Integer := 0  
 attribute interestRate: range 0..1000 := 30  
 attribute number: String -(is it String) ??  
 attribute ownerName: String -(does it belong to account or owner) ??

Operations for an account.

operation getBalance() return Integer  
 operation withdraw (amount: Positive)  
 operation deposit (amount: Positive)  
 operation computeInterest()

**Extension or Intension?**

For each statement, say if it is about the "intension" or the "extension" of a class, attribute or operation.

A professor has a name and teaches a subject.  
 Bob is a professor.

He teaches software engineering.

Yesterday, he asked John, a junior student, to explain the difference between the extension and the intension of a class.

Since John is a clever student, he was able to answer the question.

**Extension or Intension, Answers**

A professor has a name and teaches a subject. -- Intension  
 Bob is a professor. -- Extension

He teaches software engineering. -- Extension. The value of the attribute Subject is "software engineering".

Yesterday, he asked John, a junior student, to explain the difference between the extension and the intension of a class. -- Extension.  
 The professor instance Bob sends a message "ask" to the student instance John.

Since John is a clever student, he was able to answer the question. -- Extension. The "rank" attribute value of John is "clever". Difficult to model the sentence "he was able to answer the question".

## Class: Professor and Student

Suppose professors can ask questions to students. What are the consequences for the interfaces of the classes Professor and Student. Sketch these interfaces. Show on an example how professor Bob can ask the student John a question.

## Class: Professor and Student: Answers

Suppose professors can ask questions to students. 1. What are the consequences for the interfaces of the classes Professor and Student. Sketch these interfaces. 2. Show on an example how professor Bob can ask the student John a question.

1. No impact on the interface of the Professor class. The Student class must provide in its interface an operation Ask: class Student

operation ask (question: String)

...

end class Student

2. john.ask ("Difference between Intension and Extension?")

One cannot show that professor Bob asks the question, and John cannot know that he is asking. Somewhere in their behavior (code), professors may send this message, call this operation, but other classes may also ask questions to students.

## Finding Classes

- Preparation of a newspaper

- Answering machine

- Graphical editor

- Automatic Teller Machine

## Finding Classes

Find some of the main classes in the following systems:

- Preparation of a newspaper  
Journal, Journalist, Reporter, Manager, TelecommunicationEquipment, Article, News, Photograph, etc.

- Answering machine  
Message, Prompt, Phone Number, PhoneSet, Line, etc.

- Graphical editor

Point, Line, Figure, Cursor, CurrentPosition, CurrentSelection, Character, Mouse, Screen, Keyboard, etc.

- Automatic Teller Machine

BankingCard, CreditCard, BillDispenser, FinancialTransaction, Screen, Keyboard, CardReader, Printer, Message, PIN, etc.

Conclusion: It's easy to find candidate classes. It's more difficult to sort out the essential ones.

## Finding Attributes

- A person who might receive a letter
- A person suspected in a criminal case
- A person having a bank account
- The person the phone bill is sent to
- An employee of a company using a security access card



## Finding Attributes

Find some typical attributes for the following classes:

- A person who might receive a letter  
first name, last name, address, etc.
- A person suspected in a criminal case  
photograph, fingerprint, measurements, alibi, police record, motive, etc.
- A person having a bank account  
first name, last name, bank, account number, account type, etc.
- The person the phone bill is sent to  
first name, last name, address, dialing number, connection rate, communication cost, etc.
- An employee of a company using a security access card  
"his" access card number

**Conclusion: Whether or not something is an attribute is context-dependent.**

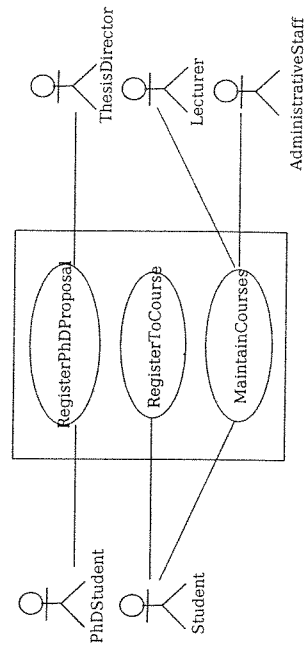


## Registration System

- Problem statement:

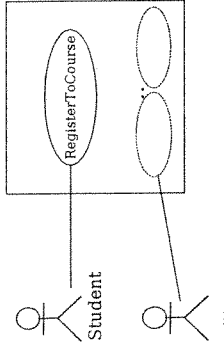
A university wants to have a system for course and PhD thesis registration by which all students can register to some course or postgraduate students register their PhD thesis proposal. The list of current courses can be modified by authorized users. Upon modification all involved persons have to be notified. Each PhD thesis proposal submitted by a PhD student has to be approved by a professor who acts as the thesis supervisor.

- Hint: The statement is left "vague" on purpose. Make a solution as simple as possible.



## Use Case Description for Registration System

- For the Registration System problem you are asked to provide one Use Case description. This description has to depict scenario of "RegisterToCourse" Use Case, that represents a goal of the student actor.



## Use Case Description for Registration System

### Extensions:

- 2a. List of available courses is empty:
  - 2a.1. System informs user that all courses are canceled or full
  - 2a.2. Use Case ends in failure
- 2b. Registration Period has not begun or it is finished:
  - 2b.1. System informs user about the registration period
  - 2b.2. Use Case ends in failure
- 3a. Student chooses to exit the application:
  - 3a.1. Use Case ends

## Use Case Description for Registration System

**Use Case:** RegisterToCourse  
**Brief Description:** A student wants to register for some of existing courses  
**Primary Actor:** Student  
**Precondition:** The Student has already identified him/herself to the System

### Main Success Scenario:

1. Student requests System to show available courses
2. System retrieves and displays the list of available courses
3. Student chooses one course from the list
4. System registers student for the chosen course and sends a message
5. Use Case ends

## Use Case Description for Registration System

- 4a Student is already registered for the selected course:
  - 4a.1. System informs Student about reason not to register
  - 4a.2. Use Case continues at the step 3
- 4b. Student's current semester does not match the semester of the selected course:
  - 4b.1. System informs Student about reason not to register
  - 4b.2. Use Case continues at the step 3
- 4c. Student did not pass all courses that are prerequisite for registration to the selected course:
  - 4c.1. System informs Student about reason not to register
  - 4c.2. Use Case continues at the step 3

### Glossary:

available courses:  
 - all courses that are not canceled or fully booked