OBJECT ORIENTED PROGRAMMING / MODELING

Handout

Seminar by

Falk-Jonatan Strube

May 4, 2016

Seminar outline

- Selected glossary
- Presentation of the topic
  - Generic programming
  - Object oriented programming and modeling
    - Inheritance
    - Polymorphism
  - Relevance of the topic
  - Sources
- (Language) practice
- Discussion

Glossary

Term	Meaning	In this context
Generic	Relating to a whole group of similar things, rather than to any particular thing	"Normal", nothing special
Redundant	Being similar to something else and thereby not needed	Duplicate code
Class	A group with certain qualities	Code with a concept
Object	A thing	An implemented class
Implementation	Putting a plan into action / starting to use something	The concept of a class applied to an object
Inheritance	A characteristic passed down from your parents	A shared characteristic between classes
Polymorphism	Poly: many, morph: form	Inherited classes may have many different implementations

Abstract

With this project the author aims to inform his fellow students about the benefits of object oriented programming and modeling.

The research was based on items from a diverse bibliography including articles, books and websites. The most important piece of literature is the book *Softwaretechnology for beginners* which is based on a lecture of the TU Dresden.

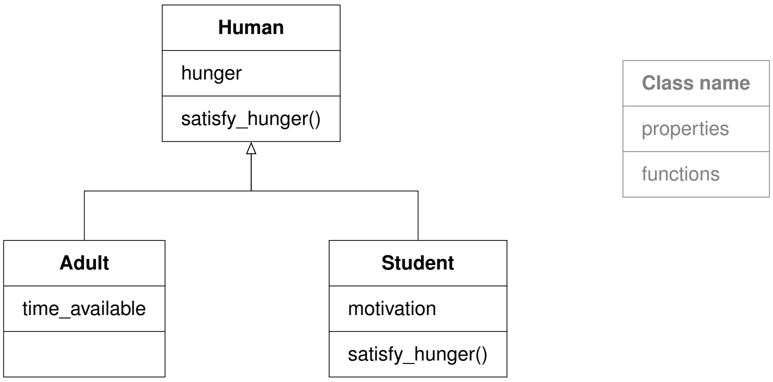
After reading the texts it can be stated that object oriented programming is an advanced way of structuring a computer program during development. By modeling dependencies and correlations in classes with the *Unified Modeling Language* (UML), a complex code formation can be made understandable. This does not only benefit team communication and communication with a client, it also improves scalability since the general overview allows specific changes in the model and the program itself.

The key concepts of object oriented programming are inheritance and polymorphism. With inheritance, similar parts of a program may be reused in *child*- and *parent*-objects. Polymorphism allows the modification of a inherited function. Both features add to comprehensibility and scalability.

It can be concluded that object oriented programming is used in almost every complex computer program. From operating systems such as *Windows* or *Linux* to word processor applications such as *Microsoft Word* – object oriented programs are everywhere. Though generic programming is more commonly known, since it is easier to grasp at first, object oriented programming is more common overall.

(Language) practice

Recall the UML-diagram from the presentation:



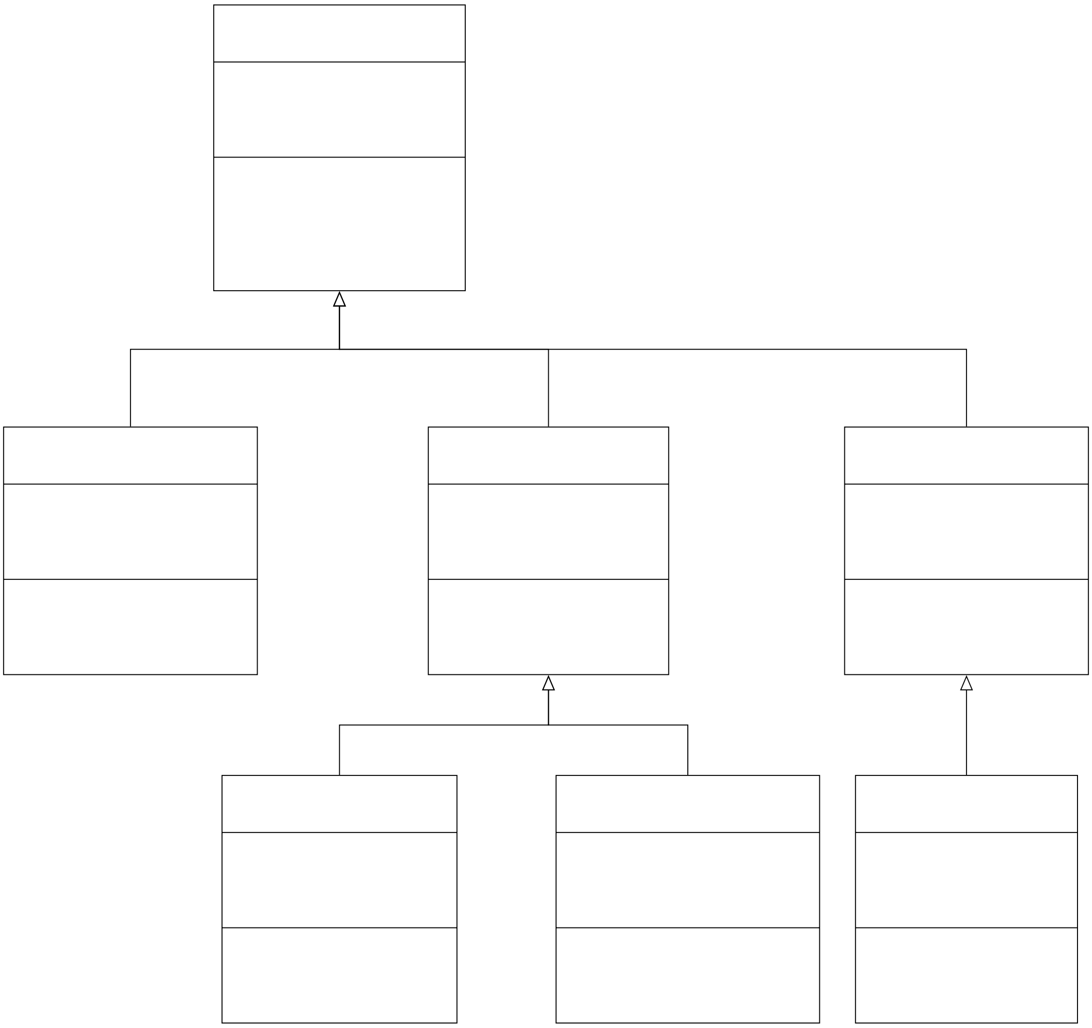
Task:

a.) Try to fill out the class names of the UML-diagram below based on the following sentences (use the diagram above as a reference on the structure).

- Students, alumni and employees of an university are persons.
- Students can be beginners or long-term students.
- A Professor is an employee.

b.) Now try to fill out the fields for properties and functions ($\widehat{=}$ actions).

- Think of properties or functions that a *parent*-class has (and therefore the child-class too) ⇒ inheritance.
- What new properties or functions may the child class(es) have, which do not fit to the parent?
- What properties or functions may be *overloaded* ⇒ polymorphism?



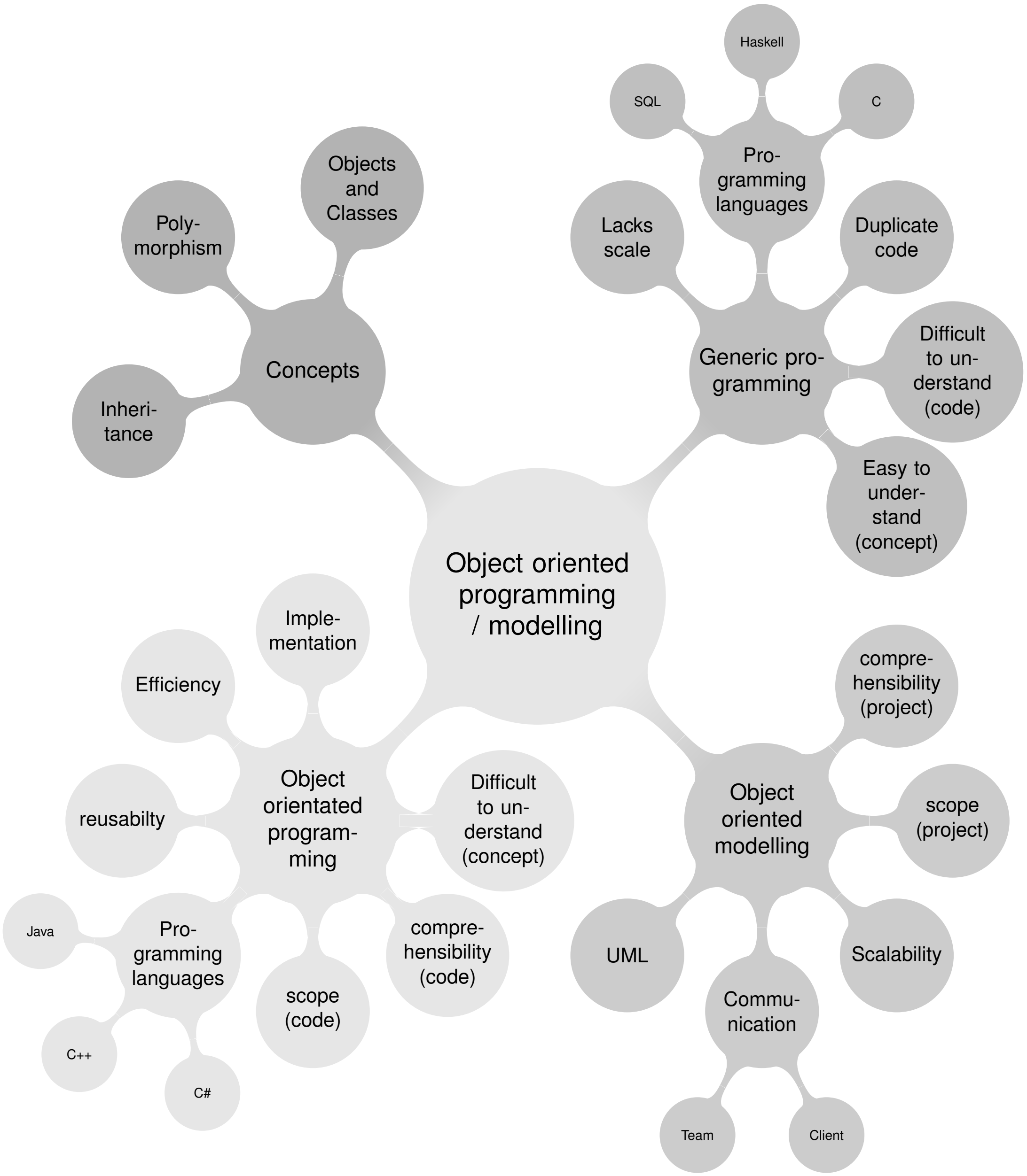
Discussion

a.) What do you think? Is object oriented modeling more comprehensive than the code from generic applications?

b.) Have you written a (small) program before? What challenges do you remember? Could they be tackled better with object oriented programming?

c.) Do you think the object oriented way of programming is used often in modern applications? Why? Why not?

Mind map


Object oriented programming / modelling
Concepts
Objects and Classes
Poly-morphism
Inheri-tance
Generic pro-gramming
Pro-gramming languages
Haskell
SQL
C
Lacks scale
Duplicate code
Difficult to un-derstand (code)
Easy to under-stand (concept)
Object orientated program-ming
Imple-mentation
Efficiency
reusabilty
Difficult to un-derstand (concept)
Pro-gramming languages
Java
C++
C#
scope (code)
compre-hensibility (code)
Object oriented modelling
compre-hensibility (project)
scope (project)
UML
Scalability
Commu-nication
Team
Client

Bibliography

[1] Birgit Demuth. *Softwaretechnologie für Einsteiger*. 2. geänd. Aufl. München: Pearson, 2014. ISBN: 9781784340230.

[2] Oracle. *Object-Oriented Programming Concepts*. [Online; accessed 20-April-2016]. URL: `https://docs.oracle.com/javase/tutorial/java/concepts/`.

[3] James Rumbaugh, Ivar Jacobson, and Grady Booch. *The unified modeling language reference manual*. 2. ed. Boston: Addison-Wesley, 2005. ISBN: 0321245628.

[4] Al Sweigart. "Why is Object-Oriented Programming Useful? (With a Role Playing Game Example)". In: *The "Invent with Python" Blog* (Dec. 2014). [Online; accessed 20-April-2016]. URL: `http://inventwithpython.com/blog/2014/12/02/why-is-object-oriented-programming-useful-with-an-role-playing-game-example/`.

[5] Matt Weisfeld. "The Importance of Object-Oriented Programming in the Era of Mobile Development". In: *informIT.com* (Apr. 2013). [Online; accessed 20-April-2016]. URL: `http://www.informit.com/articles/article.aspx?p=2036576`.