Introduction to object oriented programming in JAVA

Computer Applications in
Power Systems – Advance course

EH2750

Contents

- Programming Techniques
  - Unstructured programming
  - Procedural programming
  - Modular programming
- Object orientation
  - Thinking in OOP
  - Basic components
    - Class
    - Object
    - Properties
    - Methods
  - Basic characteristics
    - Encapsulation
- Design exercise
- Setting up the environment in JAVA

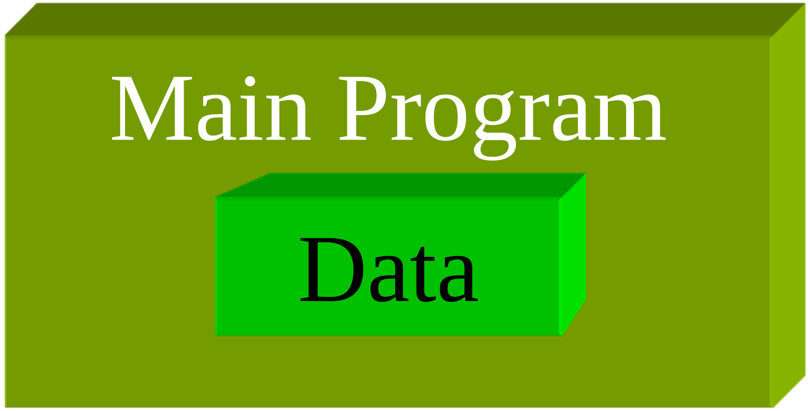
Programming Techniques

- Unstructured Programming
- Procedural Programming
- Modular & Structural Programming
- Object-Oriented Programming
- Agent Oriented Programming
  - For projects we will AOP in JACK which is java based

evolution

Unstructured Programming

- Writing small and simple programs consisting only of one main program. Here ``main program'' stands for a sequence of commands or statements which modify data which is global throughout the whole program.

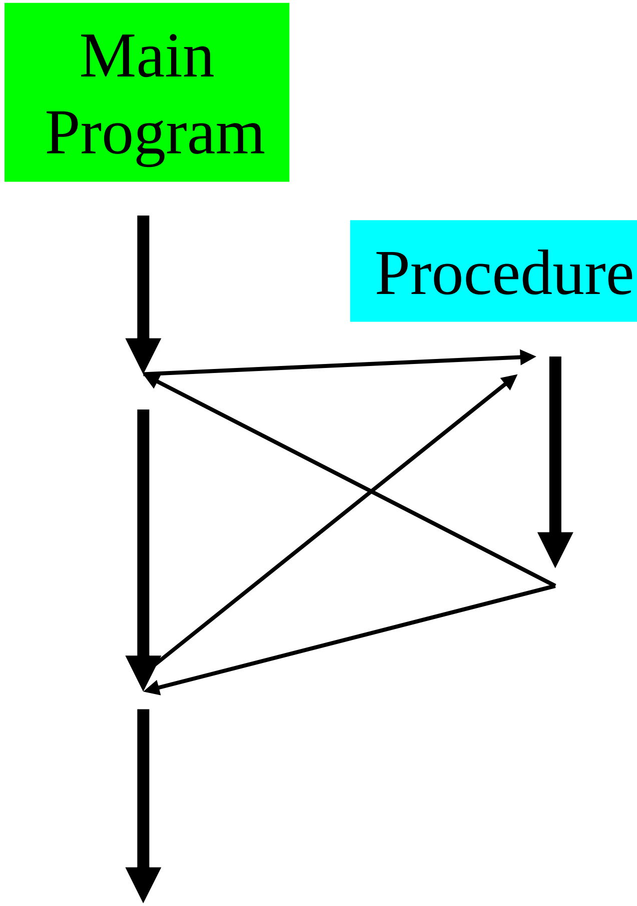


Drawbacks

- This programming technique can only be used in a very small program
- For example, if the same statement sequence is needed at different locations within the program, the sequence must be copied. If an error needed to be modified, every copy needs to be modified
- This has lead to the idea to extract these sequences(procedure), name them and offering a technique to call and return from these procedures

Procedural Programming

- With procedural programming, you are able to combine sequences of calling statements into one single place.
- A procedure call is used to invoke the procedure. After the sequence is processed, flow of control proceeds right after the position where the call was made .



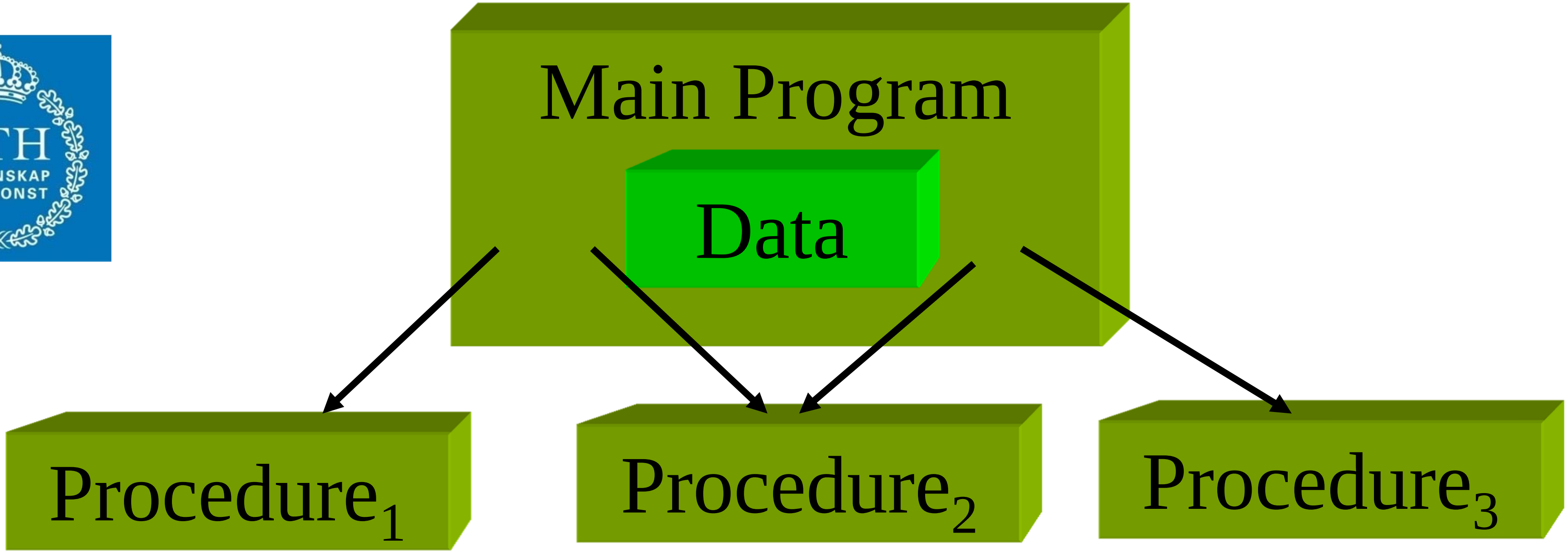
Procedures

- With parameters and sub-procedures (procedures of procedures) , programs can now be written more structured and error free
- For example, if a procedure is correct, every time it is used it produces correct results
- Consequently, in cases of errors you can narrow your search to those places which are not proven to be correct

Procedure Program view

- Now a program can be viewed as a sequence of procedure calls
- The main program is responsible to pass data to the individual calls, the data is processed by the procedures and the resulting data is presented
- Thus, the flow of data can be illustrated as a hierarchical graph, a tree.

Procedure Program view

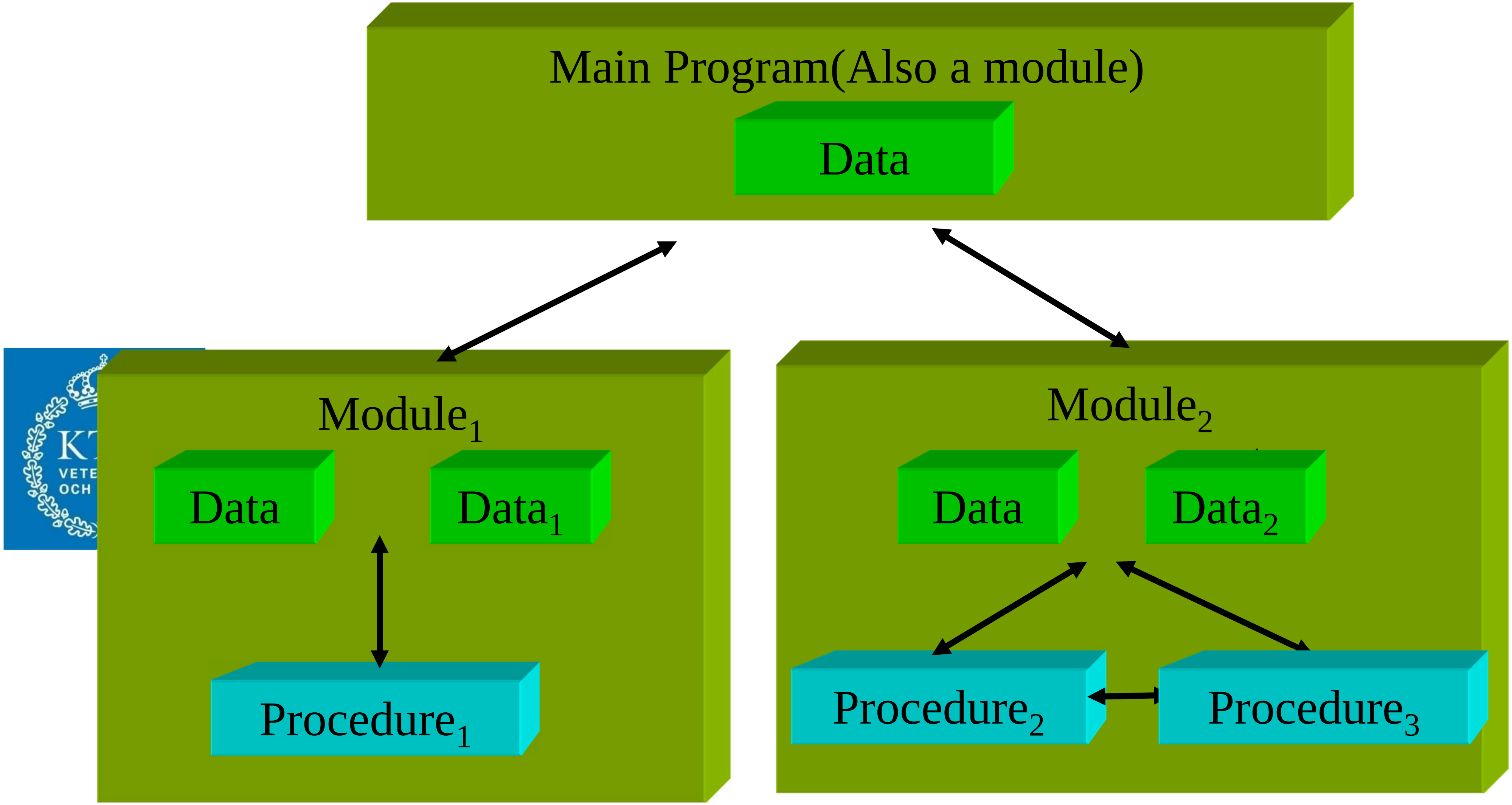


Modular Programming

- Modular programming is subdividing your program into separate subprograms such as functions and subroutines

- With modular programming, procedures of a common functionality are grouped together into separate modules

- A program therefore no longer consists of only one single part. It is now divided into several smaller parts which interact through procedure calls and which form the whole program



Main Program(Also a module)

Data

Module$_1$

Data

Data$_1$

Procedure$_1$

Module$_2$

Data

Data$_2$

Procedure$_2$

Procedure$_3$

The main program coordinates calls to procedures in separate modules and hands over appropriate data as parameters.

Modular Programming

- Each module can have its own data. This allows each module to manage an internal state which is modified by calls to procedures of this module

- Each module has its own special functionalities that supports the implementation of the whole program.

Object-Orientation

- Thinking in OOP
  - It is a kind of thinking methodology
  - Everything in the world is an object (Pure OOP*)
  - Any system is composed of objects
  - The evolution and development of a system is caused by the interactions among the objects inside or outside the system

Everything in the world is an object

- A flower, a tree, an animal
- A student, a professor
- A machine, an operator, a screen
- A university, a city, a country
- The world, the universe

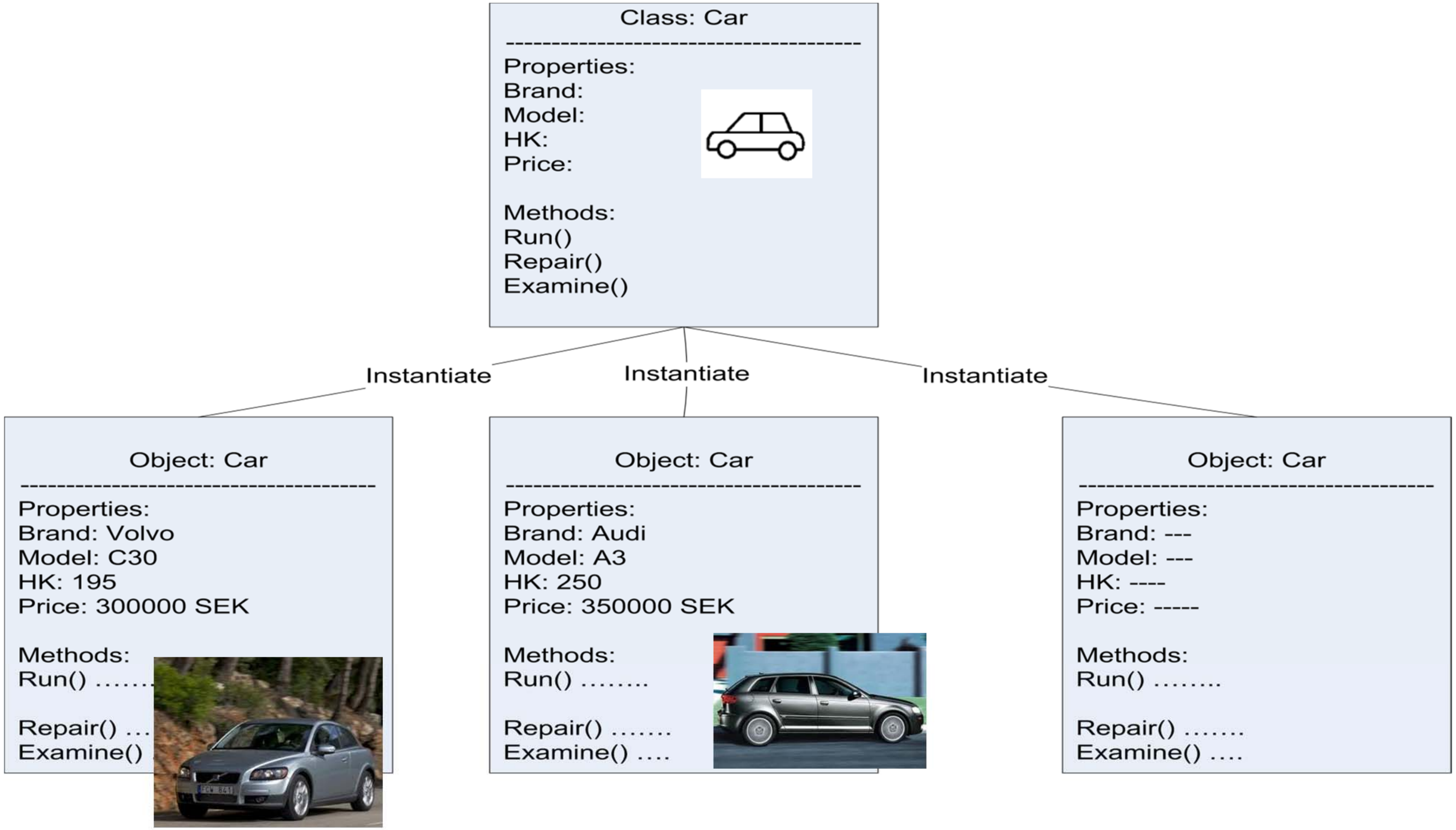
Any system is composed of objects

- A law system
- An engineering system
- A substation automation system
- An educational system
- An economic system
- An Information system

Basic compoenets of OOP

- Concept/Class (e.g., car)
- Object is an instance of a class (e.g., volvo c30)
- Properties
  - Are data attribites (brand, model, color, HK, engine size)
- Methods
  - Are functions/capabilities (start, drive, stop, service,)

Basic compoenets of OOP



Class: Car

Properties:
Brand:
Model:
HK:
Price:

Methods:
Run()
Repair()
Examine()

Object: Car

Properties:
Brand: Volvo
Model: C30
HK: 195
Price: 300000 SEK

Methods:
Run() ……

Repair() …
Examine()

Object: Car

Properties:
Brand: Audi
Model: A3
HK: 250
Price: 350000 SEK

Methods:
Run() ……..

Repair() …….
Examine() ….

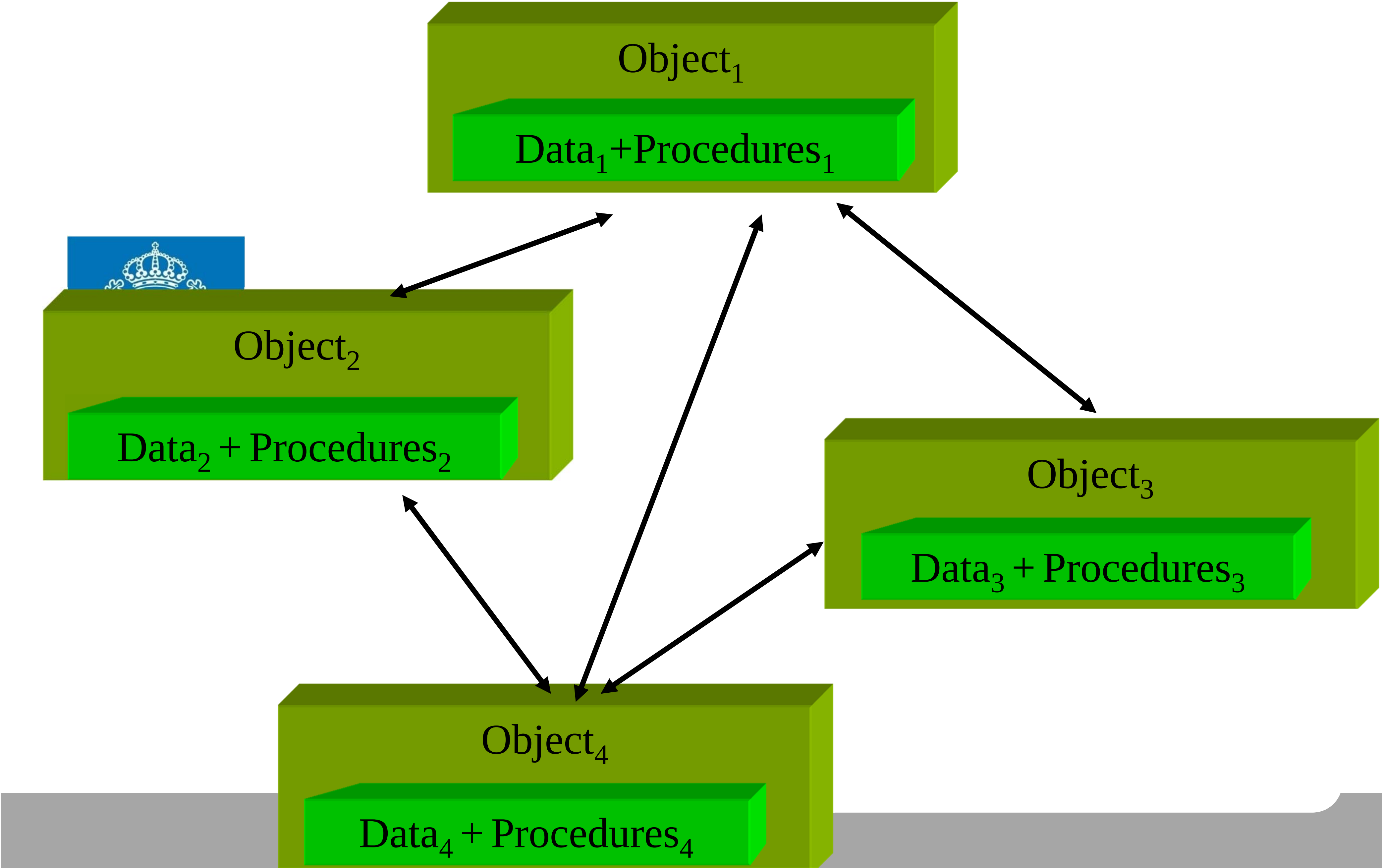
Object: Car

Properties:
Brand: ---
Model: ---
HK: ----
Price: -----

Methods:
Run() ……..

Repair() …….
Examine() ….

OOP based system view



Object$_1$

Data$_1$+Procedures$_1$

Object$_2$

Data$_2$ + Procedures$_2$

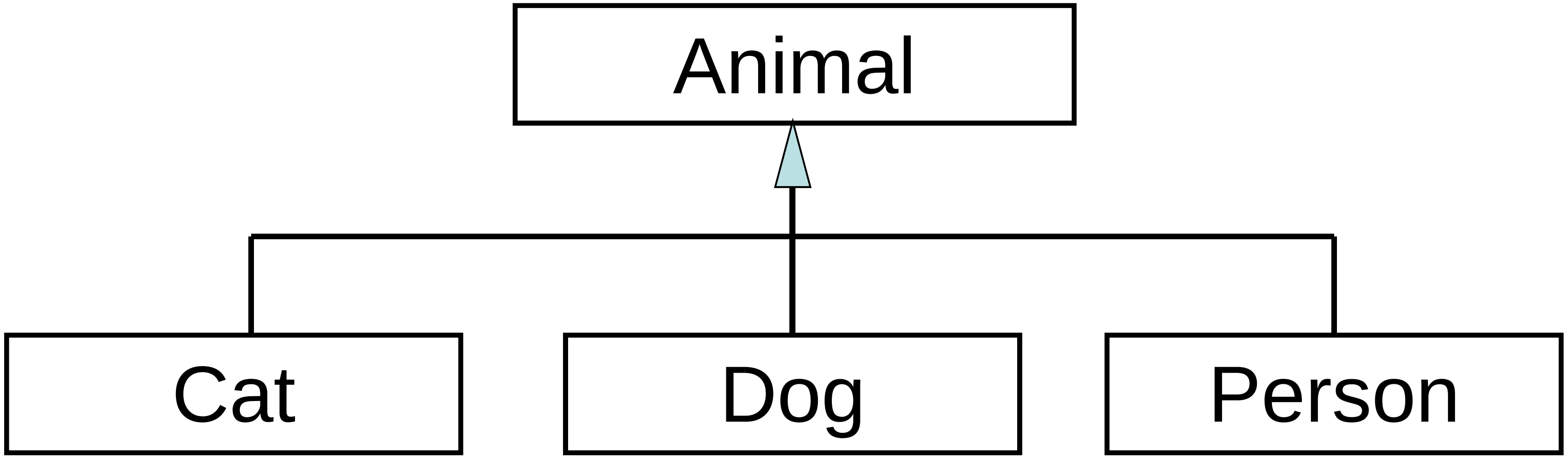
Object$_3$

Data$_3$ + Procedures$_3$

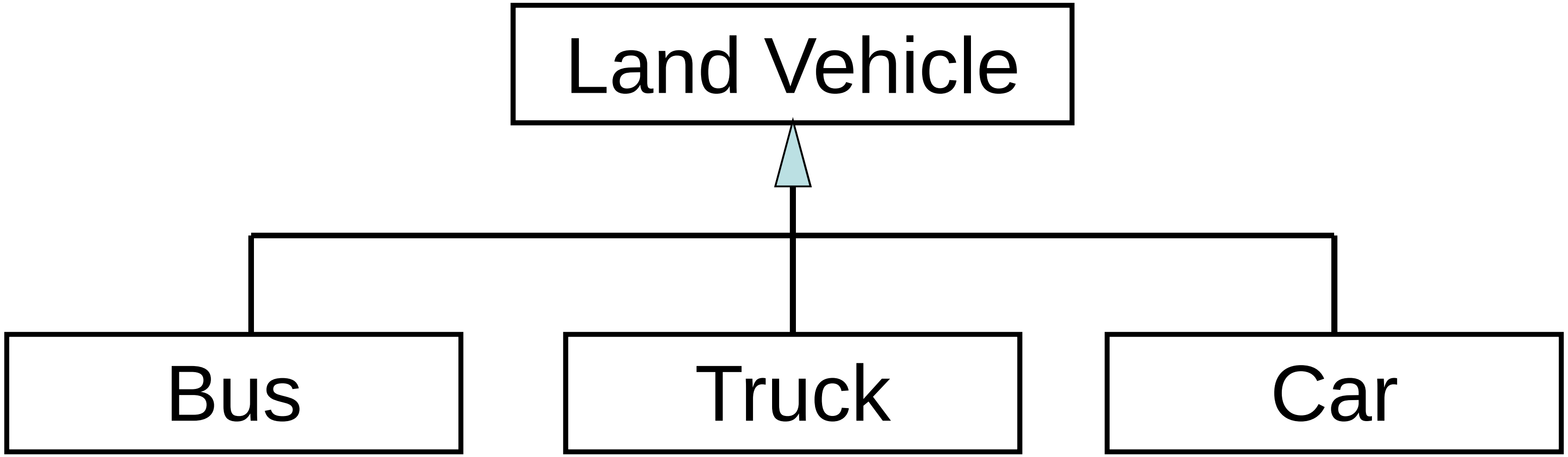
Object$_4$

Data$_4$ + Procedures$_4$

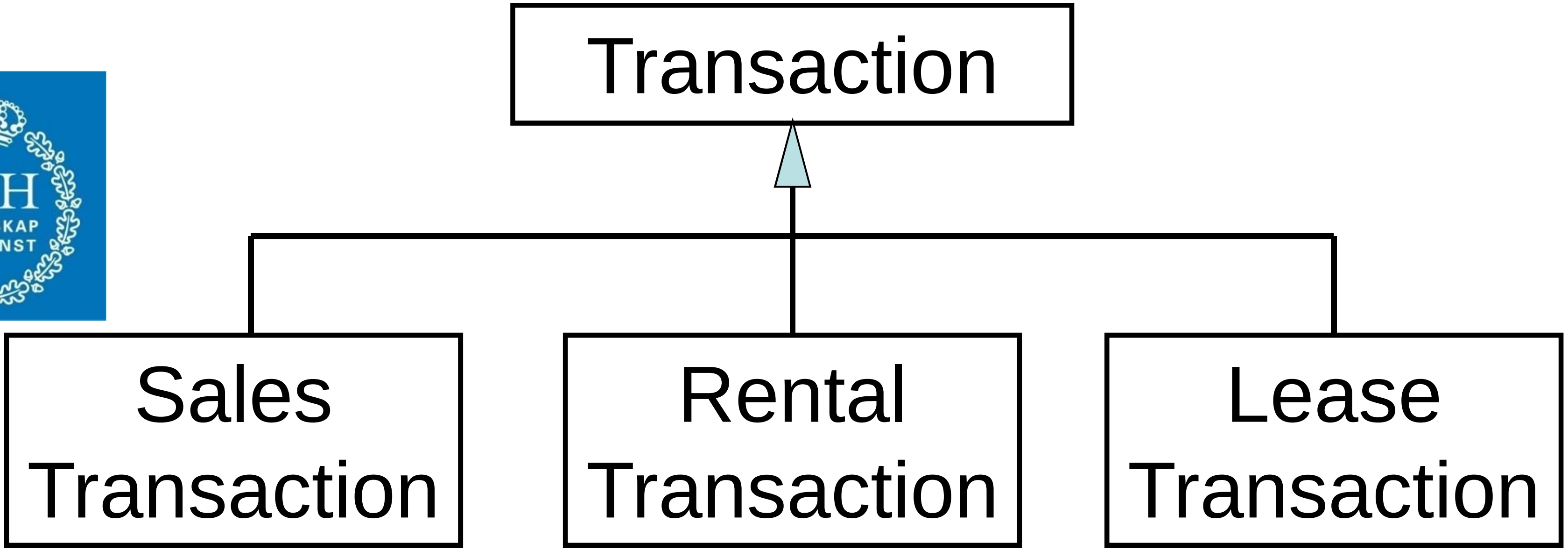
Inheritance



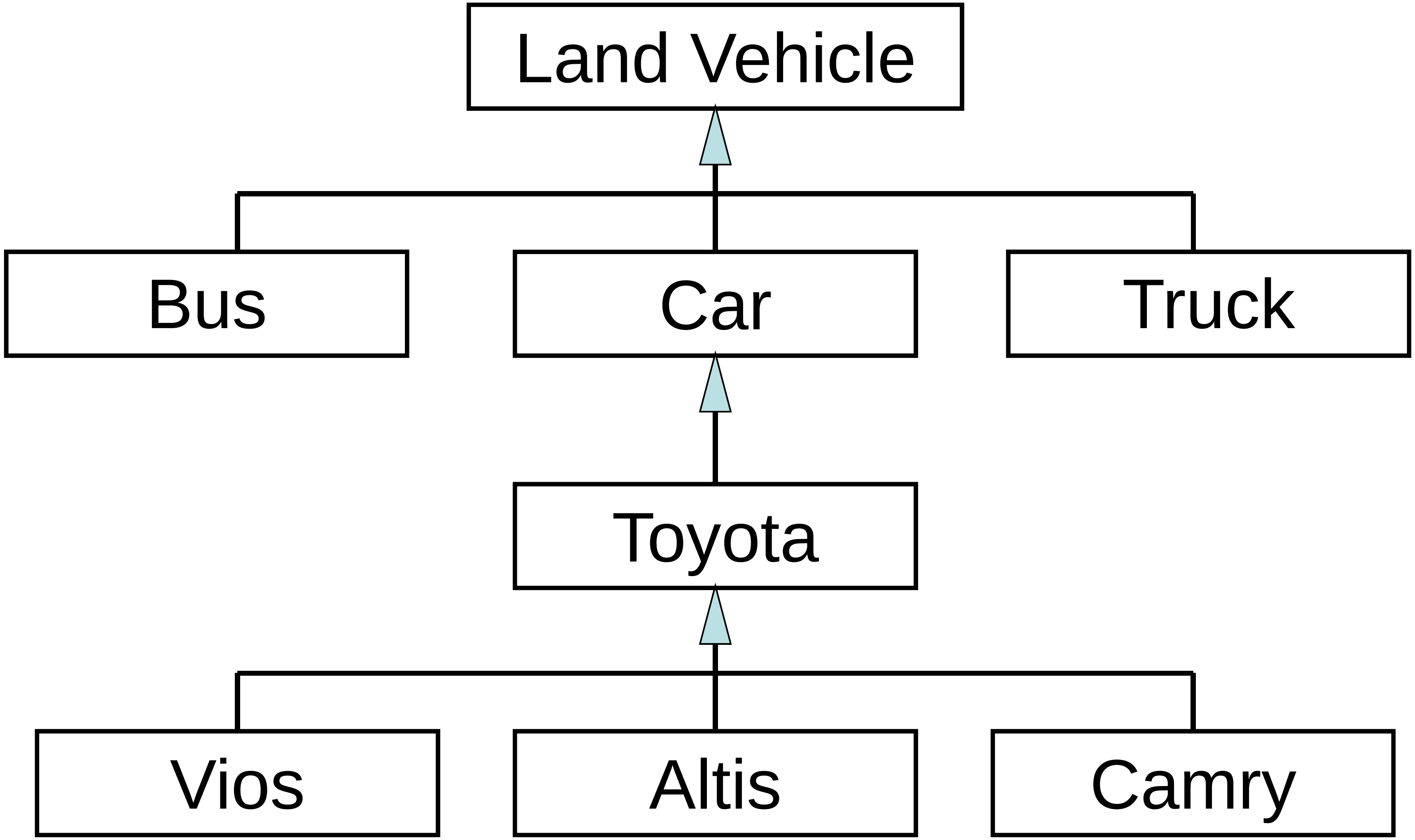
Inheritance

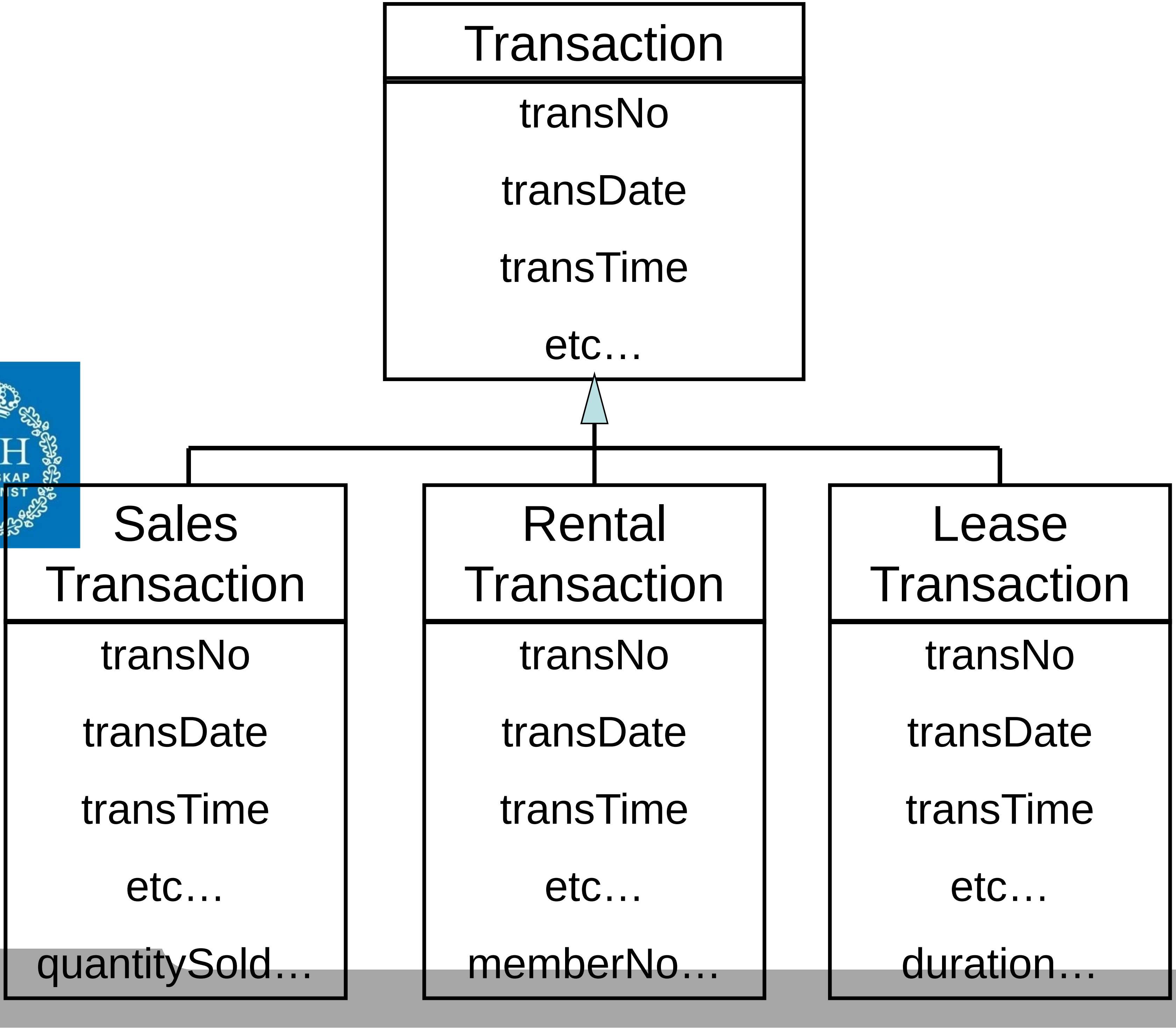


Inheritance



Inheritance




Transaction
transNo
transDate
transTime
etc…
Sales Transaction
transNo
transDate
transTime
etc…
quantitySold…
Rental Transaction
transNo
transDate
transTime
etc…
memberNo…
Lease Transaction
transNo
transDate
transTime
etc…
duration…

Setting up the environment for JAVA

- JAVA run time
- Java compiler JDK
- Download and Installation
  - If my compuyer is 32 or 64 bit ?
- Environemnt variables and class path
- Programming environment