Java How to Program

Early Objects

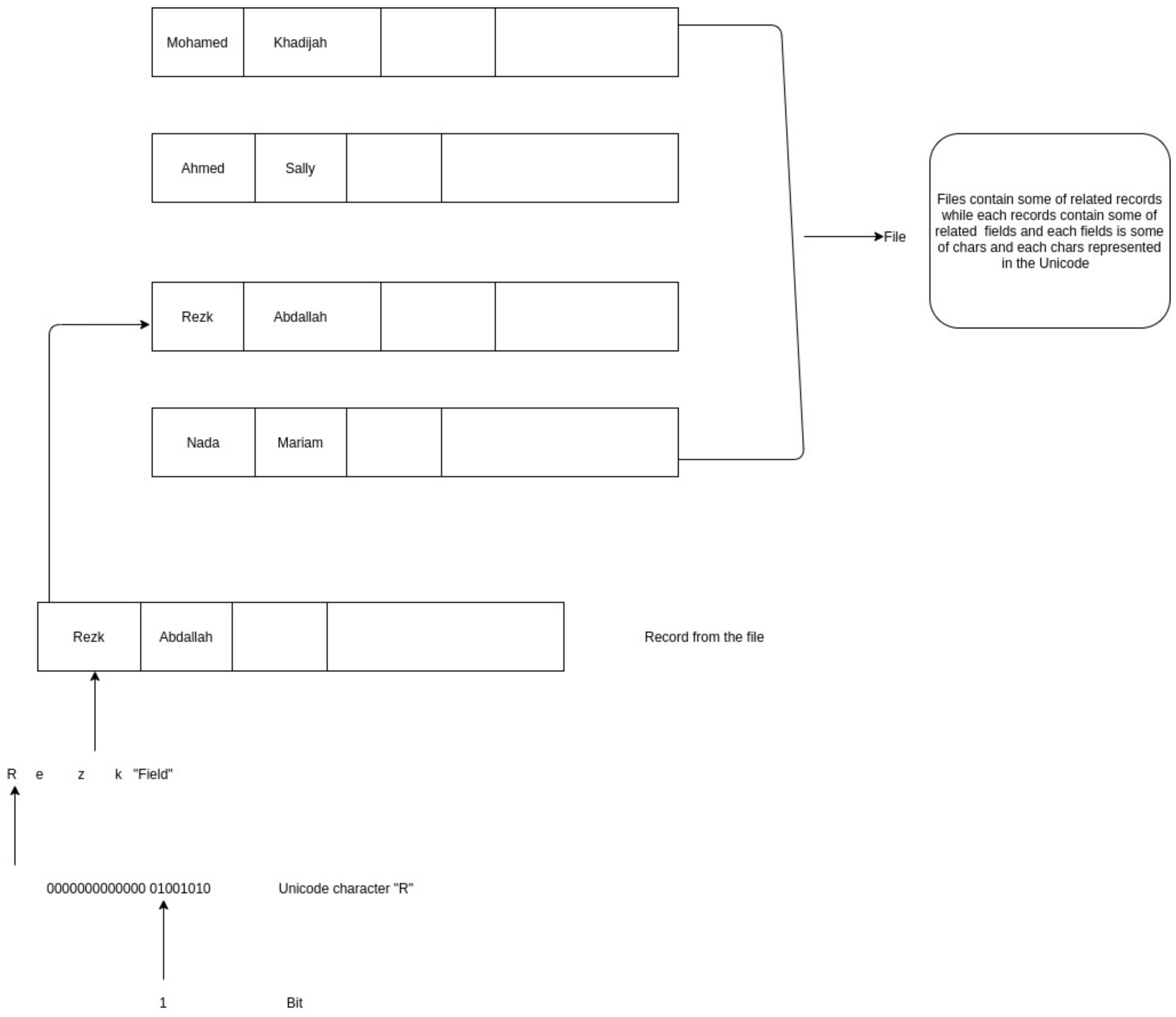
TENTH EDITION

Paul Deitel & Harvey Deitel

Chapter 1

After the introudction about the java, Hardware and Software beside of Moores Law, Data Hierarchy which how data are from the 0,1 that computer accept to the Massive data todays **"Big Data"**.

UML Graph below to display this Data Hierarchy:


Mohamed
Khadijah
Ahmed
Sally
Rezk
Abdallah
Nada
Mariam
File
Files contain some of related records while each records contain some of related fields and each fields is some of chars and each chars represented in the Unicode
Rezk
Abdallah
Record from the file
R e z k "Field"
0000000000000 01001010
Unicode character "R"
1
Bit

After the file, we have the database which represents in some ways some of the files or millions of records in tables, now days we talked about Big Data

Introduction to Object Technology

After some of these concepts, I found that there are some of the most important points he talked about and the first one is the **1.5** section which talks about **Object Oriented**.

- intro
- Methods and classes
- Instantiation
- Reuse
- Messages and Methods calls
- Attributes & Instance Variables
- Encalpsulation & Information Hiding
- Inheritance
- Interfaces
- Object Oriendt Analysis & Design with UML

Introduction

We should **Avoid reinventing the wheel**, this is the main part that we can take care of when we think in Object-Oriented Design, where we do not need to write the same method for each time we will use we can **reuse these methods**. You also drive the different cars without relies on the mechanism of the methods you are dealing with you just know how to use but not how they are work.

Now we have the time object we used from different libraries, do not need to recreate your own time object. We can represent the term object in the last of terms mentioned above like **Methods, Attributes & others**. So let us go on the methods and classes:

Methods and classes

As we mentioned above you have to drive your car but you use some of method like increase the speed of the car you can use it but you do not know the mechanism of this, it's like using the gun but you do not know how its work, just you throw your enemy.

Instantiation

You will not have the ability to use the class methods or libraries object if you do not take Instantiation of them which we call object from the class or call the object of the library to be used for you.

Reuse

Now once you have an instance of the class as we mentioned in Instantiation you have to resue the class methods and operation that the class handle, not only you, anyone will Instantiation of the class will have the ability to use, and everyone have their own of usage.

Message and Method calls

You just send a message to the class to use the methods of it, just we means by that call the methods of the calls using the instance object you take from.

Attributes & Instance Variables

Besides having the methods that perform some tasks also class may have some of the Attributes and Instance Variables to be used as the color of the car to display when user aimed to know the color of some types of cars you have in your Exhibition, the amount of the tank in your car to give the car driver alert when its need to fill.

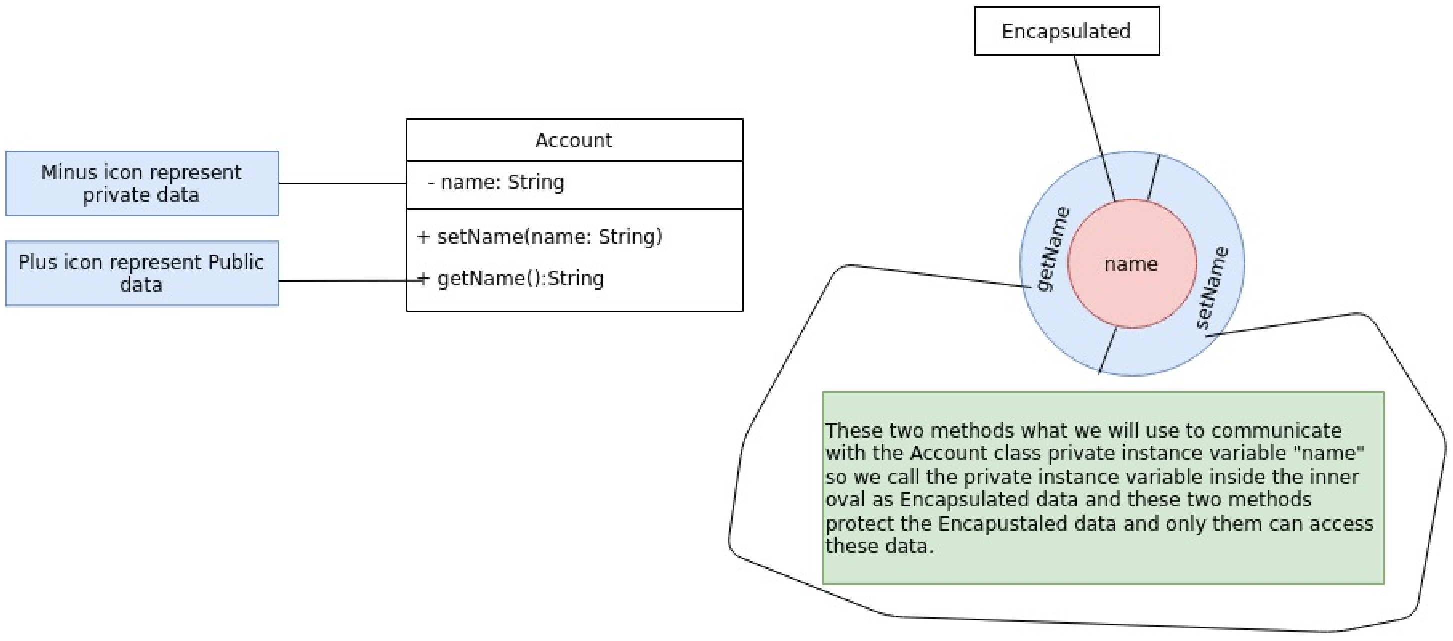
Note !

Actually, some of the methods are working on those Attributes like get the type or color of the car.

Encalpsulation & Information Hiding

Every instance(object) of the class has their own of usage to methods and attributes but they also can communicate with another object, but they are not allowed how other objects are implemented, not just implementation details normally are hidden within the object themselves like most of the attributes, so we have used methods to communicate with the attributes of class which are private Attributes.

UML Graph can represent like this Hiding information:


Minus icon represent private data
Plus icon represent Public data
Account
- name: String
+ setName(name: String)
+ getName():String
Encapsulated
getName
name
setName
These two methods what we will use to communicate with the Account class private instance variable "name" so we call the private instance variable inside the inner oval as Encapsulated data and these two methods protect the Encapustaled data and only them can access these data.

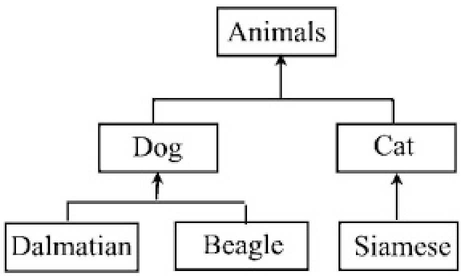
Note !

We will represent this graph again to explain all of the details contained but for now to represent the view of Encapsulation and Information Hiding

Inheritance

While we have the ability to take instances of the class and reuse them, we have the ability to edit some of the class methods, in either the header or the body of the methods, by customizing these methods according to your usage. The class you inherit from is called **Super Class** While your extended class is called **SubClass**.

Image may take your attention to understand the view:



Interfaces

Is a collection of methods that typically enable you to tell an object what to do not how to do it, interfaces are when you dealing with different types of car like the interface for turning, accelerating, and braking, but actually each type of car has its own of implementation.

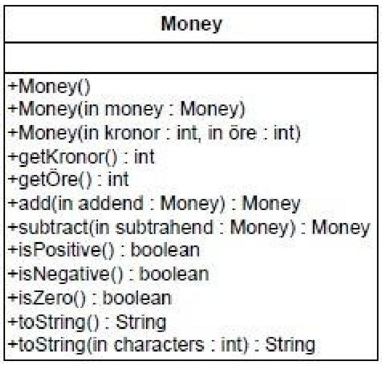
Object Oriendt Analysis & Design with UML

The last and most important part of this is this term OOAD with UML, while days of the university we take some hint about **UML** without the big picture of this term which is actually the most important part when we get into large system first thing to think about what the objective of our system is, how its looks like.

when you work with large teams first thing starts to study the requirements not Sitting down and start writing the code, you need the time to study the view of the system how it should work, what is the functionality the system require, we should consider all of the client requirements.

All of what we mentioned above have to be proven in the way of system design after collecting all of these requirements we start to analyze and design them, which we can combine together, some of the requirements may be ignored because we can use it from maybe library, but should be contained in your design to remember it.

The way we represent this design of analysis in is the **UML** Unified Model Languaged and actually from this UML you can write your code easily in no time, because each statement, variable, methods are considered in this graph, which you will just convert from its language to programing language that the computer can understand.



A typical Java Development Environment

After that the author has passed through different concepts from different operating systems to programming languages and Java itself as one of them, but also the key here is the development process that the program going on from the editor to you run the app.

As we talked about this view in Java the program go through **Five Phases**:

- Edit
- Compile
- Load
- Verify
- Execute

Edit

When you opened the text editor you use and write the program to solve some problems.

Compile

When you are finished and end from writing the code and start to compile from the language you use to the computer understanding language.

You start edit in file with extension .java, then java compiler convert to .class extension which represent your program as bytecodes.

Load

Run the program are load the bytecode you have write into the RAM or the memory of your machine.

Verify

It's a time to check you code against Java security restriction to ensure that your code with a free bugs.

Execute

When your program display the result of your code.

---------------------------------------تم بحمد الله   ------------------------------------