

Object Oriented Programming Introduction

History (1)

- [60s] **Simula-67**, Dahl and Nygaard at Oslo University
 - First programming language with OO concepts.
- [70s] **Smalltalk**, Xerox
 - First implementation of an OO language.
- [80s] **C++**, Stroustrup at Bell Labs
 - C extension with OO concepts.
 - Popularized the OO programming methodology.

History (2)

- [90s] **Java**, James Gosling at Sun:
 - Syntax close to C.
 - Java programs are executed in a virtual machine. This turns Java programs much slower than equivalent programs in C (around 50 times slower).
 - Interfaces, expose methods with empty bodies.
 - Many libraries available. One of the most popular one is JNI (*Java Native Interface*), that enables to run programs developed in other languages.
 - Very compact code (applications in client environment) attracted the attention of WWW developers.

History (3)

- [00s] **C#** (*C-sharp*), Microsoft
 - Arise from Java for the **.NET** platform.

Java popularity – Feb, 2017

The TIOBE Programming Community index is an indicator of the popularity of programming languages, inspected from normalized queries performed at popular search engines such as Google and Yahoo!

1. Java: 16.7%
2. C: 8.5%
3. C++: 5.4%
4. C#: 4.9%
5. Python: 4.0%
6. ...

Index TIOBE Programming Community:

<http://www.tiobe.com/tpci.htm>

Motivation (1)

- Before OOP languages, the typical distribution of human resources in a software house was:
 - **80%** were devoted to **maintenance** of existing systems.
 - 20% were free to design and develop new applications.

Motivation (2)

- In the context of maintenance:
 - Most of the cost is due to the growing need of doing updates over updates.
 - Successive updates are commonly performed over the source code, splitting up from the initial specifications that are not updated accordingly.
 - Subsequent updates require to work exclusively with the source code, without the support of the high level descriptions created in the analysis and design phase.

Motivation (3)

- The increase of software production can be accomplished in two different ways:
 - Increasing the number of programmers.
 - **Increasing their productivity and/or promoting the reuse of software components already existing.**

Motivation (4)

- The decrease of maintenance points out for a growing automation in transforming high level specifications into final source code.
- In this context, **CASE tools** (*Computer-Aided Software Engineering*) appeared, allowing to integrate tasks that comprise:
 - design,
 - analysis
 - programming, and
 - tests.

Basic concepts in OO programming (1)

- **Object:**
 - Something that exists (in the real world or in an information system), that is created and eventually destroyed.
 - Meanwhile, during its existence, it might suffer updates in its **state** by interacting with other entities.
 - Its state is reflected in the value of its **attributes/fields**.
 - It changes its state whenever a **method** is called.
- For instance:
 - Object: bank account.
 - Attributes/fields: balance.
 - Methods: withdraw, deposit, interest_rate.
- **Specifying a system consists in defining a set of objects.**

Basic concepts in OO programming (2)

- **Class:**
 - In general several similar objects, of the same type, coexist and are grouped in classes.
 - In fact, we have to specify the classes and not the objects, the objects appear as **instances** of the classes.
- For instance:
 - We specify the bank account class.
 - We admit that there might be several instances of that bank account (bank account objects).
- **The system built in this manner consists in a community of objects (instances of classes) that interact with each other (through method calls).**
- **The objects of the community evolve independently of each other, except when interactions take place.**

OO ingredients (1)

- **Encapsulation:**
 - Bundling of related ideas in an unit so that it can be referenced by a name.
 - In OO, it refers to pack together attributes with methods.
 - It concerns combining the state jointly with the mechanism to access and modify that state.

OO ingredients (2)

- **Specialization/inheritance:**
 - Mechanism that promotes the reuse of code.
 - Inheritance allows one object to be simultaneously an instance of more than one class.
 - The inheritance mechanism of **subclass** B over a **superclass** A allows class B to reuse some or all the methods from class A. Subclasses are also called **child classes**, whereas, superclasses are also called **parent classes** or **base classes**.
 - **Simple inheritance**, where one subclass has only a direct superclass, versus **multiple inheritance**, where one subclass has more than one direct superclass.

OO ingredients (3)

- **Specialization/inheritance** (cont):
 - For instance:
 - In the specification phase, or later in maintenance, it might be necessary to introduce a time deposit which is a bank account with some particularities (more attributes, more methods, and constrained behaviour).
 - In a time deposit, the money cannot be withdrawn for a certain period of time, unless a penalty is paid.
 - Specialization allows to specify a time deposit subclass reusing all the code needed from the bank account superclass (for instance, balance and deposit).
 - Other objects that interact with bank account objects may also interact with time deposit objects, seeing them as bank accounts (without noticing that they are in fact time deposit objects).

OO ingredients (4)

- **Polymorphism:**
 - **Redefinition** of a method with the same identifier in different classes within the same hierarchy, being possible to have distinct implementations in each of the classes.
 - In OO, polymorphism is usually implemented through **dynamic binding**, i.e., the method being executed is determined only in runtime (and not in compile time).

OO ingredients (5)

- **Polymorphism** (cont):
 - For instance:
 - The time deposit redefines the method withdraw, as it differs from the withdraw method of the bank account.
 - In compile time, the system offers a way to withdraw money of all account types (without distinguishing bank account from time deposit).
 - In runtime, the withdraw method being executed is determined by the type of the account from where it was called.

Advantages/disadvantages of OO programming

- Advantages:
 - Approximation to the real world.
 - Encapsulation of information.
 - Extensible, being easier to update and/or accommodate new requisites.
 - Reuse, by inheritance of more general classes.
- Disadvantages:
 - New approach, with more complex concepts.
 - Lower performance.

Applications of OO programming

- Bank and insurances
- Robotic
- Telecommunications
- VLSI design
- Simulation
- Databases
- Mathematic modelling
- Air traffic control
- Graphical user interfaces
- ...