OBJECT-ORIENTED PRINCIPLES SYNTHESIS

MASTER 1 Physiology, Biotechnology and Informatic Engineering

Group7: Grégoire DEVOUCOUX, Cléo MARCHAND, Eden DUPORT, Kenza BERRADA, Honorine BORDES, Ingrid GOMA-LOUAMBA, Laura METAIS, Kokou THONOU (Absent), Noémie UGOLIN

Supervising professors : Allan FOUSSE and Patrick GIRARD

Date
3 sept 2024

School Year: 2024 - 2025

I. Group composition

- DEVOUCOUX Grégoire
- MARCHAND Cléo
- DUPORT EDEN
- BERRADA Kenza
- BORDES Honorine
- GOMA-LOUAMBA Ingrid
- METAIS Laura
- THONOU Kokou (Absent)
- UGOLIN Noémie

II. Distribution of roles

- DEVOUCOUX Grégoire : Groupe Leader
- MARCHAND Cléo : Time Master
- DUPORT EDEN : Scribe

Introduction

A programming paradigm is a way to structure the implementation of a computer program. It provides a framework for approaching computer programming and formulating solutions to problems as well as their formalization in an appropriate programming language.

Programming paradigms can be categorized into various types:

- Imperative Paradigm(procedural, object-oriented,structured)
- Declarative Paradigm(functional, logic)

In the imperative paradigm, the order in which operations occur is explicitly stated, often with constructs that control that order, and side effects are allowed. In contrast, declarative programming emphasizes the goal more than the order of operations.

The Imperative way of programming is easy to implement and intuitive for beginners. Its main drawback is solving complex problems, because of its low efficiency and productivity. Examples of Imperative programming languages can be Java, Python and C.
In contrast, Declarative way of programming has a more concise code structure and is less prone to errors. Its main drawback lies in less expressive code structure for complex algorithms and in controlling the actual execution process. As examples, there are Haskell, Java and Prolog.

Definition of Object-Oriented Programming

The word “objects” meant as object-oriented programming was first mentioned at the artificial intelligence group at Massachusetts Institute of Technology (MIT) in the 1950s. This term refers to LISP atoms, which is a family of programming languages, with identified properties or attributes. Later in the 1970s, the American Alan Kay precise its principle. Object-oriented programming, or OOP, is a programming model which consists of a concept of class and objects.
The main goal is to link data and functions that work on this data so that no other part of the code can act on it except for these functions.
An object is a symbolic and autonomous component of a program that is encoded to perform certain actions and to interact with other elements of the program. It can be named and described, and they are modeled in the program.
A class is a model or a base structure which defines objects’ attributes and functions. It can be seen as an architect's plan while objects are the results of this plan.
For example, an object’s type “Car” could have attributes like “color” and “brand”, and functions like “accelerate” and “brake”.

Definition of Object-oriented analysis and design

It’s a software development method based on object oriented program (OOP) that comes down to analyzing and conceiving a system focusing on the objects, their attributes, behaviors and interactions.
OOAD’s concept is based on two important steps : object-oriented analysis and object-oriented design .

- Object-oriented analysis is a stage where the problem domain is studied to identify the relevant object.
- Object-oriented design is a step that defines how the objects interact with each other.

OOP allows objects to interact with each other using four basic principles: encapsulation, inheritance, polymorphism, and abstraction.

Encapsulation is the principle of having information and details in an object. It presents all main information inside an object and exposes only the data chosen in the outside world.

Abstraction occurs when the user interacts solely with selected attributes and methods of an object. Therefore, it is based on the use of simple “things” to represent complex notions. The underlying code is represented by objects and classes for the user to see, while complex details of the code are hidden to the user. It is an extension of encapsulation.

Inheritance defines hierarchic relations between classes so that common attributes and functions can be used another time.

With base attributes and behaviors in class, we can create a secondary class containing functionalities of the main class and add attributes as well as more behaviors.

Polymorphism is a key concept in OOP, it allows objects from different classes to use functions with the same name but they can act differently depending on the class of the object. It allows for more flexibility and it is easier to manipulate objects.

Definition of Unified Modeling Language

Unified Modeling Language (UML) is a visual modeling language that is used as a way to visualize the structure of a system and its different processes. It uses pictograms and regroups 3 types of diagrams : behavior, interaction and structure diagrams. It enables non-programmers to understand the structure of a system and the interactions between different systems.

It is the result of the combination of the notations of former existing modeling methods such as the Booch method, the object-modeling technique and the object-oriented software engineering. It was adopted as a standard by the Object Management Group in 1997. Many updated versions have been released since.

Conclusion

Although OOP is a powerful and useful way of solving many informatic problems, it has some negative effects as it is not easy to use intuitively, it is vague, needs rigor in the code, as well using “virtual methods” adds detour for each action, which greatly increases execution time. While its advantages are the reusability of code, modularity and its extensibility.