Jorge Adrián Padilla Velasco
A00570894
August 21st 2018

Porgramming Paradigms

Imperative Paradigm

- **Description:**

  It's a paradigm in which the program describes a sequence of steps that change the state of the computer. Imperative programming explicitly tells the computer *how* to accomplish something. Programs written this way often compile to binary executables that run more efficiently since all CPU instructions are themselves imperative statements.

- **Where it's commonly used:**

  Imperative languages can resolve practically any problem in any area: from simple to complex mathematics calculus (aereal control, Artificial Intelligence, ATMs, smartphones).

- **Advantages and disadvantages:**

  Advantages: its relatively simple and easy for compilers and interpreters implementation, ability to reuse the same code in different parts of the program without copying it, it's easier to follow the program's flow.

  Disadvantages: the data is exposed to the entire program (hence, no security), difficult to relate with real world objects, it reduces extend-ability because it's difficult to create new data types.

- **Programming Languages:**

  Fortran: is a general-purpose, compiled imperative programming language that is especially suited to numeric computation and scientific computing.

  Pascal: is an imperative and procedural programming language, which Niklaus Wirth designed in 1968–69 and published in 1970, as a small, efficient language intended to encourage good programming practices using structured programming and data structuring.

  C: is a general-purpose, imperative computer programming language, supporting structured programming, lexical variable scope and recursion, while a static type system prevents many unintended operations.

Functional Paradigm

- **Description:**

  It’s a paradigm in which we try to bind everything in pure mathematical functions style. It is a declarative type of programming style. Its main focus is on “what to solve” in contrast to an imperative style where the main focus is “how to solve”. It uses expressions instead of statements. An expression is evaluated to produce a value whereas a statement is executed to assign variables.

- **Where it’s commonly used:**

  It is used in mathematical computations and it is needed where concurrency or parallelism is required.

- **Advantages and disadvantages:**

  Advantages: pure functions are easier to understand because they don’t change any states and depend only on the input given to them, the ability to treat functions as values and pass them to functions as parameters make the code more readable, testing and debugging is easier, used to implement concurrency/parallelism.

  Disadvantages: writing programs in recursive style instead of using loops can be intimidating, combining pure functions with the rest of the application and I/O operations is difficult, immutable values and recursion can lead to decrease in performance.

- **Programming Languages:**

  Haskell: is a standardized, general-purpose compiled purely functional programming language, with non-strict semantics and strong static typing.

  Elixir: is a functional, concurrent, general-purpose programming language that runs on the Erlang virtual machine (BEAM). Elixir builds on top of Erlang and shares the same abstractions for building distributed, fault-tolerant applications.

  Scala: is a general-purpose programming language providing support for functional programming and a strong static type system. Designed to be concise, many of Scala's design decisions aimed to address criticisms of Java.

Object Oriented Paradigm

- **Description:**

  It’s programming paradigm based upon objects (having both data and methods) that aims to incorporate the advantages of modularity and reusability. Objects, which are usually instances of classes, are used to interact with one another to design applications and computer programs.

- **Where it’s commonly used:**

  OOP is about encapsulating mutable state and is therefore more appropriate for interactive applications, GUI's, and API's exposing mutable state.

- **Advantages and disadvantages:**

  Advantages: improved software development productivity, improved software maintainability, faster development, lower cost of development, higher quality software.

  Disadvantages: steep learning curve, larger program size, slower programs, not suitable for all types of problems.

- **Programming Languages:**

  Simula: is considered the first object-oriented programming language. As its name suggests, Simula was designed for doing simulations, and the needs of that domain provided the framework for many of the features of object-oriented languages today.

  Smalltalk: is an object-oriented, dynamically typed, reflective programming language. Smalltalk was created as the language to underpin the "new world" of computing exemplified by "human–computer symbiosis."

  Java: is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible.

Logic Paradigm

- **Description:**

  It’s a computer programming paradigm in which program statements express facts and rules about problems within a system of formal logic. Rules are written as logical clauses with a head and a body; for instance, "H is true if B1, B2, and B3 are true." Facts are expressed similar to rules, but without a body; for instance, "H is true."

- **Where it’s commonly used:**

  Logic programming can be viewed as controlled deduction. An important concept in logic programming is the separation of programs into their logic component and their control component. With pure logic programming languages, the logic component alone determines the solutions produced. The control component can be varied to provide alternative ways of executing a logic program. This notion is captured by the slogan

  Algorithm = Logic + Control

  where "Logic" represents a logic program and "Control" represents different theorem-proving strategies.

- **Advantages and disadvantages:**

  Advantages: improved software development productivity, improved software maintainability, faster development, lower cost of development, higher quality software.

  Disadvantages: steep learning curve, larger program size, slower programs, not suitable for all types of problems.

- **Programming Languages:**

  Prolog: is a general-purpose logic programming language associated with artificial intelligence and computational linguistics. Prolog has its roots in first-order logic, a formal logic, and unlike many other programming languages, Prolog is intended primarily as a declarative programming language: the program logic is expressed in terms of relations, represented as facts and rules.

  Answer Set Programming (ASP): is a form of declarative programming oriented towards difficult (primarily NP-hard) search problems. It is based on the stable model (answer set) semantics of logic programming. In ASP, search problems are reduced to computing stable models, and *answer set solvers*—programs for generating stable models—are used to perform search.

  Datalog: is a declarative logic programming language that syntactically is a subset of Prolog. It is often used as a query language for deductive databases. In recent years, Datalog has found new application in data integration, information extraction, networking, program analysis, security, and cloud computing.

References

https://www.computerhope.com/jargon/i/imp-programming.htm

https://en.wikipedia.org/wiki/Imperative_programming

https://www.slideshare.net/JFREDYOLAYARAMOS/paradigma-imperativo-39302522

https://en.wikipedia.org/wiki/Fortran

https://en.wikipedia.org/wiki/Pascal_(programming_language)

https://en.wikipedia.org/wiki/C_(programming_language)

https://www.geeksforgeeks.org/functional-programming-paradigm/

https://en.wikipedia.org/wiki/Haskell_(programming_language)

https://en.wikipedia.org/wiki/Elixir_(programming_language)

https://en.wikipedia.org/wiki/Scala_(programming_language)

https://www.tutorialspoint.com/object_oriented_analysis_design/ooad_object_oriented_paradigm.htm

https://softwareengineering.stackexchange.com/questions/325807/when-do-we-actually-use-object-oriented-programming

https://www.saylor.org/site/wp-content/uploads/2013/02/CS101-2.1.2-AdvantagesDisadvantagesOfOOP-FINAL.pdf

https://en.wikipedia.org/wiki/Simula

https://en.wikipedia.org/wiki/Smalltalk

https://en.wikipedia.org/wiki/Java_(programming_language)

https://www.computerhope.com/jargon/l/logic-programming.htm

https://en.wikipedia.org/wiki/Logic_programming

https://en.wikipedia.org/wiki/Prolog

https://en.wikipedia.org/wiki/Answer_set_programming
https://en.wikipedia.org/wiki/Datalog