Programming languages and programming paradigms

Prof.ssa Filomena Ferrucci

Overview

- Claudio: history presentation
- Overview: App Inventor site
- App Inventor Research
- Ragazze e Informatica
  - Progetto Nerd http://www.progettonerd.it/
- Programming languages and programming paradigms

Activity: Programming Paradigms—Exploration of Learners' Knowledge

- Stage A: Worksheet, individual work
  - The students are asked to work individually on the worksheet presented in Table 3.3.
- Stage B: Class discussion
  - After the students work on this worksheet individually, the instructor collects their answers and discusses them with the class.

Table 3.3 Worksheet on programming paradigms

- Answer the following questions
  1. What is a programming *language*
  2. What is a programming *paradigm?*
  3. Give examples of three programming paradigms. For each paradigm
     - explain why it is a programming paradigm
     - list at least two programming languages that implement the said paradigm
  4. Compare the two concepts: a programming paradigm and a programming language

A definition for the concept of programming paradigm

- A paradigm is a way of doing and seeing things, a framework of thought in which one's world or reality is interpreted.
- The concept paradigm became popular in the scientific world mainly following Kuhn's book *The Structure of Scientific Revolution (Kuhn 1962), in which he used this term with respect to a conceptual* world view that consists of
  - formal theories, classic experiments, and trusted methods.

Programming paradigm

- The concept of programming paradigm is defined in different ways each emphasizing different aspects of the concept.
- see, e.g., Abelson et al. 1996; Ambler et al. 1992; Floyd 1979; Sethi 1996; Tucker and Noonan 2002; Van Roy and Haridi 2004; Watt 1990

Programming paradigm

- Programming paradigms are heuristics used for solving algorithmic problem.
- A programming paradigm analyzes a problem through specific lens, and based on this analysis, formulates a solution for the given problem by breaking the solution down to specific building blocks and defining relationships among them.
- Major programming paradigms:
  - procedural (imperative),
  - object-oriented,
  - functional,
  - logical,
  - Concurrent.

The difference between a programming paradigm and a programming language

- A programming paradigm is a heuristic for solving algorithmic problems
- A programming language is a means of expression for a programming paradigm

Functional programming paradigm

- Functional programming paradigm represents the process of computation as calculation of stateless mathematical functions and thus attempts to eliminate or minimize side effects.
- The concept of functions as in functional programming differs from the one used in procedural programming, and is closer to mathematical understanding: functions don't have side effects, i.e., their result depends only on the values of their arguments and not on other factors.
- Some functional languages can allow usage of local variables or side effects, but in purely functional languages such things are prohibited. Non-functional languages can apply functional paradigm in process of development as well; the benefits of immutable data and absence of side effects are worth of incorporating in real-life projects.
- Functional programming is usually considered to be a sub-paradigm of **declarative programming**: it doesn't matter how exactly a function is implemented as long as it returns the same value each time it is called, so the details of implementation can vary unless they don't change the behavior of the function.

Lisp ("LISt Processing")

- It is a high-level expression-oriented programming language
- 1958 – 1959 (first interpreter)
- The first language to use if-then-else control structure
- Common Lisp, Scheme

```
(defun factorial (n)
  (if (= n 0)
      1
      (* n (factorial (- n 1))) ) )
```

Lisp - Fibonacci

```
(defun fibonacci (n)
  (if (< n 3)
      1
      (+ (fibonacci (- n 1)) (fibonacci (- n 2))) ) )
```

Logic programming paradigm

- It is largely based on formal logic.
- Any program written in a logic programming language is a set of sentences in logical form, expressing facts and rules about some problem domain.
- Major logic programming language families include Prolog, Answer set programming (ASP) and Datalog .
- *Rules* are written in the form of *clauses*:

H :- B_1, …, B_n. and are read declaratively as logical implications:
H if B_1 and … and B_n.
H is called the *head* of the rule and B_1, …, B_n is called the *body*.

- *Facts* are rules that have no body
  H

Prolog

- Prolog has its roots in first-order logic, a formal logic,
- Prolog is declarative: the program logic is expressed in terms of relations, represented as facts and rules.
- A computation is initiated by running a *query* over these relations

Highlights

- Since programming languages implement programming paradigms, different programming languages may implement the same programming paradigm,
- if one is familiar with one programming language which represents a specific programming paradigm, it is reasonable to assume that he or she will be able to switch smoothly to another programming language that represents the same programming paradigm
- The differences between programming paradigms are more fundamental and therefore, unlike switching between programming languages that represent the same programming paradigm, the switching between between programming paradigms is not (in most cases) a trivial cognitive task.

How to introduce programming?

- http://malchiodi.di.unimi.it/teaching/tfa/didattica-1-1/#/

Python

- https://www.youtube.com/watch?v=4iZ9kBoDVSU