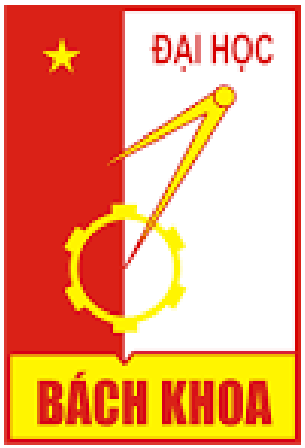
Nature of Programming Languages

Topic 3. Programming paradigms

Han, Nguyen Dinh (han.nguyendinh@hust.edu.vn)



Faculty of Mathematics and Informatics
Hanoi University of Science and Technology

Outline

1. The imperative programming paradigm
2. The functional programming paradigm
3. The logic programming paradigm
4. The object-oriented programming paradigm
5. Language selection

1

The imperative programming paradigm

The imperative programming paradigm

The language designer has in mind *a particular style of programming* for which the language is intended to be suitable. Hence, a major design decision such as including or omitting certain concepts must imply a really distinctive style - or paradigm - of programming

Bjarne Stroustrup

Bjarne Stroustrup (Creator of C++) notes that a language cannot be merely a collection of "neat" features. Thus, language design is not just design from first principles, but an art that requires experience, experiments, and sound engineering trade offs. Adding a major feature or concept to a language should not be a leap of faith but a deliberate action based on experience and fitting into a framework of other features and ideas of how the resulting language can be used

The imperative programming paradigm

Imperative programming is so called because it is based on *commands* (or *imperatives*) that update variables held in storage. The imperative programming is the oldest but still the dominant paradigm

A programming language is imperative if it is characterized by three properties: (1) the sequential execution of instructions, (2) the use of variables representing memory locations, and (3) the use of assignment to change the values of variables

Most programming languages today are imperative. An imperative programming language simply abstract the operational aspects of a machine (e.g. Pascal, C) and programs in this language are written with *control* as the key element

Example 1. The power function in imperative form

The following function (in Pascal) works by iteration; it uses local variables to accumulate the product and to control the iteration

```
function power (b: Real; n: Integer): Real;
    var p: Real; i: Integer;
    begin
        p := 1.0;
        for i := 1 to abs (n) do
            p := p * b;
        if n >= 0 then
            power := p
        else
            power := 1.0 / p
    end
```

2

The functional programming paradigm

The functional programming paradigm

We can often think of a program as implementing a *mapping*. This is because the program takes some input values and maps them to output values. In imperative programming the mapping is archived by commands that read input values, manipulate them, and write output values

In functional programming, instead of commands, tasks will be done by the use of expressions and functions. Thus, programs are written with focus on operations (functions) on data values

The extensive use of functions is the basis of the functional programming paradigm. However, this requires more powerful concepts, notably *higher-order functions* and *lazy evaluation*

Example 2. The power function in functional form

The following function (in Pascal) is the *functional version* with recursion of the power function given in Example 1

```
function power (b: Real; n: Integer): Real;
    begin
        if n = 0 then
            power := 1.0
        else if n > 0 then
            power := b * power (b, n-1)
        else
            power := 1.0 / power (b, -n)
    end
```

3

The logic programming paradigm

The logic programming paradigm

Logic programming is based on the notion that a program implements a *relation* rather than a mapping. Since relations are more general than mappings, logic programming is potentially higher-level than imperative or functional programming

In logic programming, programs are written with focus on *relationships* between data values. Indeed, that programs specify "what" must be true about a problem's solution

Programming in this paradigm, however, requires programmers to know a little bit about mathematical logic. The kind of logic used in logic programming is the first-order predicate calculus, which is a way of formally expressing logical statements, that is, statements that are either true or false

The logic programming paradigm

Can anyone teach us

how to write and run some simple programs

in Prolog?

4

The object-oriented programming paradigm

The object-oriented programming paradigm

The object-oriented programming paradigm is based on the concepts of *object* and *object class*. An object is a variable equipped with operations that have exclusive right to access it. An object class is a set of objects that share the same operations. In a sense, this paradigm is an extension of the imperative programming, particularly in the implementation of objects

The object-oriented programming paradigm, has acquired enormous importance over the last 30 years. It allows programmers to write reusable code that operates in a way that mimics the behavior of objects in the real world; as a result, programmers can use their natural intuition about the world to understand the behavior of a program and construct appropriate code

Example 3. The ATM system

The software to be designed will control a simulated automated teller machine (ATM) having a magnetic stripe reader for reading an ATM card, a user console (keyboard and display) for interaction with the user, a slot for depositing envelopes, a dispenser for cash, a printer for printing user receipts, and a key-operated switch to allow an operator to start or stop the machine. The ATM will communicate with the bank's computer over an appropriate communication link. The ATM will service one user at a time. A user will be required to insert an ATM card and enter a personal identification number (PIN) - both of which will be sent to the bank for validation as part of each transaction. The user will then be able to perform one or more transactions. The card will be retained in the machine until the user indicates that he/she desires no further transactions, at which point it will be returned

(by Prof. Russell Bjork, Gordon College)

NOTE $\rightsquigarrow$ Functional, non-functional, and usability requirements are in red, blue and violet, respectively

Example 3. The ATM system (cont.)

Now, main functions of the ATM system can be described briefly as follows

Insert card and validate PIN: This allows the PIN entered against the central record held on the bank system. If the PIN is correct then the session can continue. If the PIN is not correct then the session will terminate, and the card should be returned to the user

Manage PIN: It allows changing the four digit PIN with a new value

Get balance: It displays/prints out the current balance of the bank account

Withdraw cash: The user chooses an amount to be withdrawn. If all required conditions (e.g. overdraft limit, cash available, etc.) are true then the bank account, the card limit per day and the dispenser amount are updated, the card is released and the cash is dispensed; if one of these conditions is false then the user will be prompted to consider another option

The object-oriented programming paradigm

Can anyone teach us how to model

(i.e. give the general usecase and class diagrams)

and

implement the ATM system

in an object-oriented programming language?

5

Language selection

Language selection

How should we select

a programming language

for a specific software project?

Your third assignment

Students work individually to carry out the following tasks:

1. Try to run pieces of codes given in chapters 3 to 5 of the textbook
2. Select some examples of codes that reflect clearly the style of programming mentioned in each chapter
3. Make changes of the selected codes if necessary (i.e. they can be run as a program) and collect all obtained programs
4. For each program in your collection, describe its purpose and guide how to run it
5. Report (in Word or Latex) and submit your results

NOTE ⇝ The report should have separate parts (i.e. each for one programming paradigm). Please convert your report to PDF when you submit it

Programming paradigms

Thank you very much for your attention!