



Alliance with  Education

The FPT logo consists of the letters "F", "P", and "T" in a stylized, italicized font. The "F" is blue, the "P" is orange, and the "T" is green. A small registered trademark symbol (®) is located to the right of the "T".

Assignment name: Assignment 1

Student name: Demo

Student ID: xxx

Tutor: Bui Duy Linh

Table of Contents

| | |
|--|-----------|
| Introduction | 3 |
| I. Algorithm..... | 3 |
| 1. Definition, characteristics and example of algorithm | 3 |
| 2. Criteria of a good algorithm | 4 |
| 3. Seven steps of building an application | 6 |
| 4. From writing code to execution | 8 |
| 5. The relationship between an algorithm and the code variant..... | 8 |
| II. Programming paradigm | 9 |
| 1. Procedural paradigm | 9 |
| 2. Object-oriented paradigm..... | 10 |
| 3. Event-driven paradigm | 10 |
| III. Integrated Development Environment | 11 |
| Conclusion..... | 14 |
| References | 14 |

Introduction

- Báo cáo sẽ thảo luận về **định nghĩa, cách sử dụng và đặc điểm của thuật toán và cách áp dụng các thuật toán** để viết chương trình hiệu quả và hiệu quả hơn.
- Ngoài ra, báo cáo cũng xem xét **các mô hình lập trình** khác nhau và đặc điểm của từng mô hình.
- Báo cáo sẽ kiểm tra các **tính năng phổ biến của Môi trường phát triển tích hợp** và cách chúng giúp các lập trình viên thực hiện công việc tốt hơn.

I. Algorithm

1. Definition, characteristics and example of algorithm

T

Thuật toán là "một loạt các hướng dẫn chi tiết để thực hiện một thao tác hoặc giải quyết vấn đề"
Ví dụ, các bước để giải khối Rubik, công thức chế biến một món ăn nhất định

Trong lập trình, các thuật toán xác định cụ thể từng bước cần thiết phải được thực hiện để máy tính tạo ra kết quả đầu ra từ đầu vào

Có năm đặc điểm quan trọng của thuật toán:

- **Tính sắp xếp hợp lý:** Các bước làm có thứ tự cụ thể, rõ ràng.
- **Hoạt động rõ ràng:** Mỗi thao tác trong một thuật toán phải cực kỳ rõ ràng để máy tính có thể hiểu được. Để được thực hiện bởi máy tính, mỗi lệnh phải được viết bằng ngôn ngữ máy tính.
- **Tính toán hiệu quả:** Mỗi thao tác trong một thuật toán phải có thể thực hiện được.
- **Tạo ra kết quả:** Để xác minh tính chính xác của thuật toán, chúng ta phải xem kết quả mà nó tạo ra.
- **Tính tạm dừng:** Thuật toán không thể được phép chạy mãi mãi. Mỗi thuật toán phải có các hoạt động báo hiệu điểm cuối cùng của nó.

Ví dụ: Thuật toán sắp xếp bubble sort

Các bước thực hiện:

Bước 1: Nhìn vào số đầu tiên, biến nó thành số hiện tại để so sánh.

Bước 2: So sánh số hiện tại với số tiếp theo.

Bước 3: Nếu số tiếp theo nhỏ hơn số hiện tại, hãy trao đổi vị trí của chúng. Nếu không, đừng trao đổi.

.

B

Bước 4: Di chuyển đến số tiếp theo trong danh sách được cập nhật, biến nó thành số hiện tại để so sánh.

Bước 5: Lặp lại từ bước 2 cho đến số cuối cùng trong danh sách.

Bước 6: Nếu có bất kỳ số nào được hoán đổi từ Bước 1 đến Bước 5, hãy lặp lại quy trình từ Bước 1.

Bước 7: Nếu không có giao dịch hoán đổi nào được thực hiện trong quá trình chạy từ số đầu tiên đến số cuối cùng, danh sách được sắp xếp và thuật toán dừng lại.

Hình dưới đây cho thấy thuật toán sắp xếp danh sách các số nguyên từ 1 đến 5.

Chú ý: Các bạn vào trang sau để xem các thuật toán cơ bản

https://www.cs.utexas.edu/~mitra/csFall2017/cs303/lectures/basic_algo.html

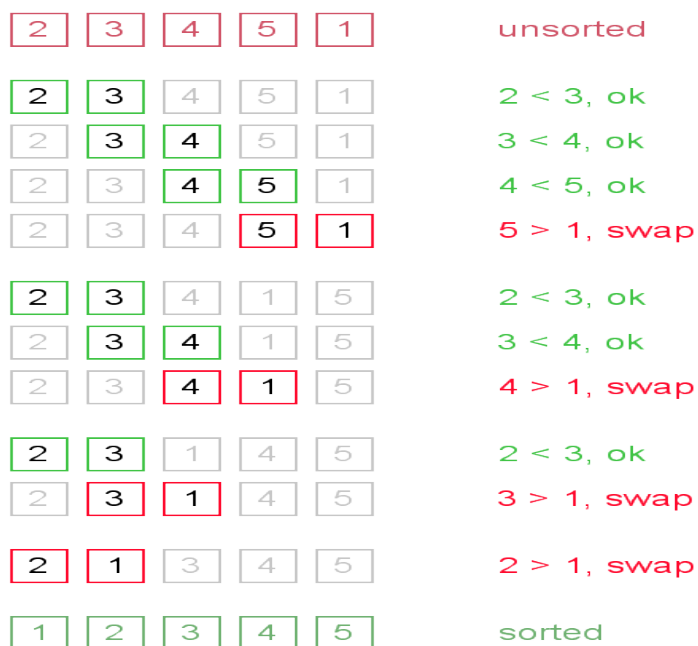


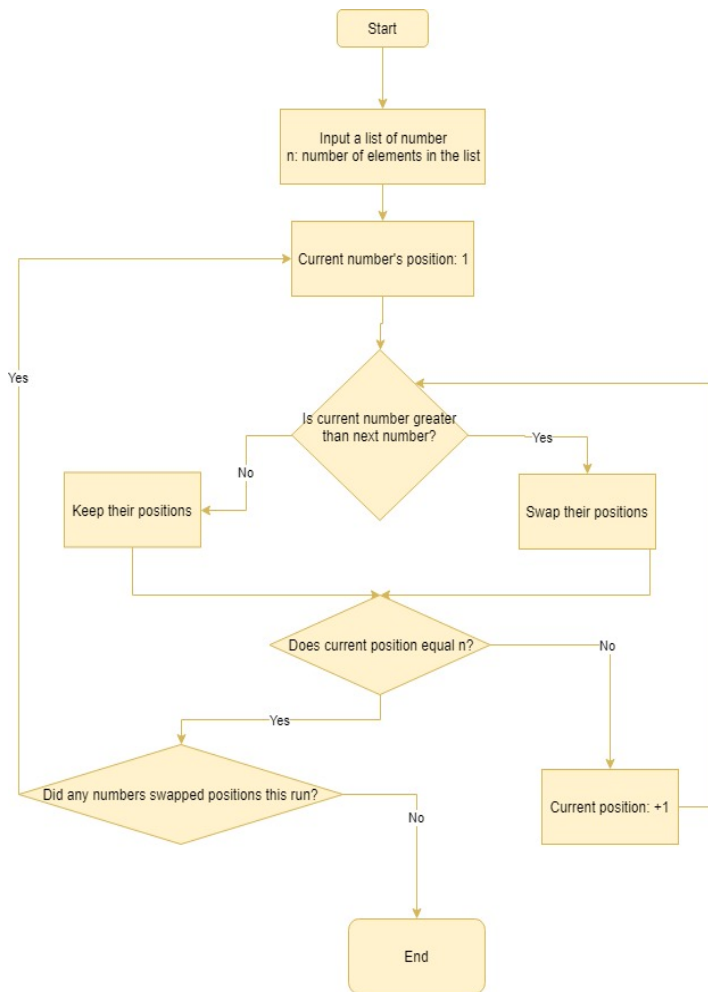
Figure 1. Bubble sort algorithm

2. Criteria of a good algorithm

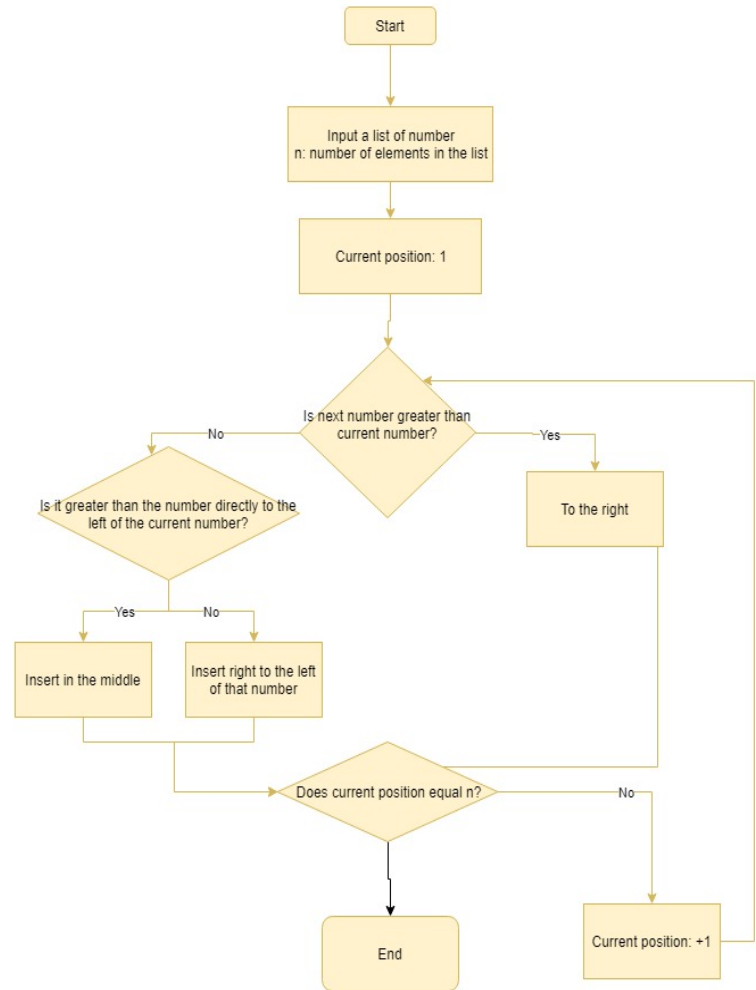
Để được coi là một thuật toán tốt, người ta phải đáp ứng các tiêu chí sau:

- Chính xác: Thuật toán phải có khả năng tạo ra kết quả tốt dựa trên đầu vào chính xác.
- Hiệu quả: Thuật toán nên đơn giản, càng ít bước càng tốt. Điều này có thể tăng hiệu suất và giảm tải ứng dụng trên các tài nguyên của máy tính như CPU và RAM.

Chúng ta hãy xem 2 thuật toán Bubble sort và insertion sort



Bubble sort algorithm



Insertion sort algorithm

Figure 2. Bubble sort vs Insertion sort

T

Thuật toán sắp xếp chèn liên tục chọn ra từng phần tử của danh sách. Phần tử đầu tiên được chọn ra và đưa vào danh sách mới gọi "danh sách được sắp xếp". Sau đó, từng phần tử khác được chọn ra từng cái một và so sánh với phần tử trước đó. Nếu phần tử đang xem xét (hãy gọi nó là phần tử A) lớn hơn phần tử trước (phần tử B), thì A được đặt ở bên phải của B trong danh sách được sắp xếp. Nếu không, A được so sánh với phần tử bên phải của B, hãy gọi nó là phần tử C. Nếu A lớn hơn C, thì A được chèn vào giữa B và C. Nếu không, nếu A nhỏ hơn C, sau đó A được chèn phải sang trái của C. Quá trình được lặp lại cho đến khi phần tử cuối cùng trong danh sách.

So sánh hai thuật toán, chúng ta có thể dễ dàng đưa ra rằng cách sắp xếp chèn tốt hơn nhiều so với sắp xếp bong bóng.

Sắp xếp chèn nhìn vào từng thành phần và xây dựng một danh sách được sắp xếp cùng một lúc. Trong khi đó, sắp xếp bong bóng phải lặp qua danh sách nhiều lần, thực hiện nhiều lần hoán đổi trong mỗi vòng lặp.

Do đó, chúng ta có thể kết luận rằng thuật toán sắp xếp chèn vừa nhanh hơn và hiệu quả hơn thuật toán sắp xếp bong bóng.

3. [Seven steps of building an application](#)

Prata (2004) đã mô tả bảy bước lập trình như sau:

B

Prata (2004) đã mô tả bảy bước lập trình như sau:

Bước 1: Xác định các Mục tiêu của Chương trình: Điều đầu tiên cần làm khi lập kế hoạch viết chương trình là tìm hiểu nhu cầu của khách hàng và xác định những gì được yêu cầu của ứng dụng. Chúng ta sẽ có thể xác định đầu vào cần thiết và đầu ra mong muốn.

Bước 2: Thiết kế chương trình: Sau khi xây dựng một hình ảnh chung về những gì chương trình nên làm, bước tiếp theo là tìm hiểu cách thức hoạt động của nó. Nói cách khác, đây là bước xây dựng thuật toán. Thuật toán nên có tất cả các đặc điểm của bất kỳ thuật toán nào, cũng như đáp ứng các tiêu chí của một thuật toán tốt. Những điều khác cần xem xét khi thiết kế chương trình bao gồm chọn giao diện người dùng và chọn cách thể hiện dữ liệu.

Hãy lưu ý, không nên viết mã ở Bước 1 và Bước 2.

Bước 3: Viết mã: Bây giờ thiết kế cho chương trình đã sẵn sàng, lập trình viên bắt đầu viết mã thực tế. Dựa trên thiết kế (hoặc thuật toán) của chương trình, lập trình viên hiện dịch mọi thứ thành mã bằng một ngôn ngữ phù hợp (ví dụ: C, C ++, Java, C # v.v.). Các lập trình viên cũng nên ghi lại chương trình. Các tài liệu có thể được bao gồm bên trong mã, hoặc được viết trong một tệp riêng biệt.

Bước 4: Biên dịch: Bước tiếp theo là dịch từ ngôn ngữ mã nguồn mà con người có thể hiểu sang ngôn ngữ máy cho máy tính hiểu. Đây là công việc của trình biên dịch. Mỗi môi trường phát triển tích hợp (IDE) nên bao gồm một trình biên dịch cho ngôn ngữ lập trình của nó. Trình biên dịch cũng tìm kiếm các lỗi bên trong mã. Máy tính chỉ có thể hiểu các lệnh được viết hoàn hảo. Ngay cả một lỗi đánh máy đơn giản hoặc một lệnh hơi mơ hồ cũng có thể gây ra lỗi biên dịch. Sau khi xác minh rằng mã không có lỗi, trình biên dịch sẽ dịch mã thành mã thực thi sẵn sàng để chạy.

Bước 5: Chạy chương trình: Thông thường, sau khi biên dịch, một tệp thực thi sẽ được tạo ra, có thể được mở từ hệ điều hành. Tuy nhiên, hầu hết các IDE đã tích hợp việc thực thi mã, vì vậy các lập trình viên có thể biên dịch và chạy chương trình bên trong IDE.

Bước 6: Kiểm tra và gỡ lỗi chương trình: Mã nguồn được biên dịch và thực thi không có nghĩa là chương trình đã sẵn sàng để sử dụng và phân phối. Lập trình viên phải chạy chương trình để tìm lỗi bên trong mã. Có rất nhiều sai lầm mà các lập trình viên có thể mắc phải, chẳng hạn như áp

dụng thuật toán sai hoặc bỏ qua các yêu cầu dữ liệu nhất định. Quá trình gỡ lỗi nhằm tìm và loại bỏ những sai lầm đó trước khi đưa ra sản phẩm cuối cùng.

Bước 7: Duy trì và sửa đổi chương trình: Làm thêm giờ, chương trình có thể thêm các tính năng mới để điều chỉnh nhu cầu mới của người dùng. Lập trình viên có thể muốn áp dụng các tinh chỉnh cho mã để tăng cường hiệu năng hoặc loại bỏ các lỗi mới tìm thấy. Bước này có thể được thực hiện dễ dàng nếu lập trình viên tuân theo một thiết kế tốt và cung cấp tài liệu phù hợp.

4. From writing code to execution

Hãy tưởng tượng quá trình lập trình của bạn như thể bạn là một kiến trúc sư phụ trách xây dựng một ngôi nhà. Đầu tiên, bạn phải nói chuyện với khách hàng để tìm hiểu nhu cầu của họ. Họ thích loại nhà nào? Cao hoặc thấp? Trường hiện đại hay cũ? Họ có muốn có một khu vườn không? Họ có cần một nhà để xe? Sau khi thu thập tất cả các thông tin cần thiết, bạn bắt đầu phác thảo kế hoạch cho ngôi nhà. Bạn xác định diện tích, số lượng vật liệu cần thiết, sơ đồ sàn và thậm chí là thiết kế điện. Sau khi hoàn thành, bạn mang công việc của mình đến người xây dựng chính để anh ấy và nhóm của anh ấy biến thiết kế của bạn thành hiện thực. Điều này tương đương với việc cung cấp mã nguồn cho máy tính để chạy. Thật không may, tất cả các nhà xây dựng là người nước ngoài và không thể hiểu bất kỳ từ nào trong hướng dẫn của bạn. Tương tự, máy tính chỉ hiểu ngôn ngữ máy cấp thấp, không phải ngôn ngữ cấp cao mà bạn nhập vào IDE. Kết quả là, bạn phải thuê một biên dịch viên (trình biên dịch) để truyền đạt ý tưởng của bạn cho các nhà xây dựng. Trình biên dịch biến tất cả các thiết kế và hướng dẫn của bạn thành ngôn ngữ của người xây dựng. Điều duy nhất còn lại phải làm là đưa các bản thiết kế được dịch cho các nhà xây dựng và để họ thực hiện công việc của họ.

Trong C #, các bước để thực hiện chương trình được mô tả như sau:

- Khung .Net sử dụng trình biên dịch C # để dịch mã nguồn của bạn sang mã Ngôn ngữ trung gian (MSIL) của Microsoft.
- Sau đó, Runtime Ngôn ngữ chung (CLR) tải gói của bạn bao gồm mã MSIL.
- Trình biên dịch JIT (Just in Time) trong CLR chuyển đổi mã MSIL thành ngôn ngữ máy.
- Cuối cùng, mã máy được thực thi bởi hệ điều hành..

..

5. The relationship between an algorithm and the code variant

Đ

Để đánh giá mối quan hệ giữa một thuật toán và biến thể mã, chúng ta hãy tham khảo lại định nghĩa của thuật toán. Nói tóm lại, đó là một loạt các hướng dẫn, hướng dẫn từng bước hoặc một công thức. Như vậy, thuật toán là độc lập với bất kỳ ngôn ngữ lập trình. Thuật toán là hướng dẫn và ngôn ngữ lập trình (biến thể mã) là sự triển khai của thuật toán đó. Một thuật toán có thể được thực hiện bởi nhiều ngôn ngữ và mỗi ngôn ngữ có cách riêng để thực hiện một thuật toán nhất định..

II. Programming paradigm

V

Về mặt ngôn ngữ học, mô hình có nghĩa là một mô hình hoặc một mô hình. Các mô hình lập trình là các phân loại dựa trên đặc trưng nhất định của các ngôn ngữ lập trình. Đặc biệt, các ngôn ngữ lập trình được chia thành các mô hình khác nhau dựa trên các tính năng và đặc điểm của chúng. Trong báo cáo này, chúng tôi sẽ thảo luận về bốn mô hình phổ biến được liệt kê dưới đây..

..

1. Procedural paradigm

M

Mô hình thủ tục, hoặc lập trình thủ tục, dựa trên các thủ tục hoặc chức năng. Nó chỉ định các bước phải được thực hiện để chương trình tạo ra một kết quả nhất định và chia chương trình thành các chức năng riêng biệt để chạy.

Đặc điểm của mô hình thủ tục:

- Nó tập trung vào quá trình, không phải dữ liệu.
- Nó coi một vấn đề là một chuỗi nhiệm vụ và mỗi nhiệm vụ được quản lý bởi một chức năng riêng biệt.
- Nó sử dụng chủ yếu dữ liệu toàn cầu, có thể được chia sẻ giữa các chức năng..

..

2. Object-oriented paradigm

Object-oriented

Object-oriented Mô hình hướng đối tượng dựa trên khái niệm "đối tượng". Một đối tượng có thể chứa cả dữ liệu và chức năng. Trong mô hình hướng đối tượng, một chương trình bao gồm các đối tượng tương tác với nhau.

Đặc điểm của mô hình hướng đối tượng:

- Lớp: Nền tảng của mô hình hướng đối tượng, một thực thể duy nhất chứa cả dữ liệu và thủ tục.
- Đối tượng: Các thể hiện hoặc việc thực hiện một lớp.
- Trừu tượng: Chỉ định những việc cần làm, nhưng không phải cách thực hiện, do đó tăng cường tính linh hoạt của chương trình.
- Đóng gói: Dữ liệu và thủ tục đều được đóng gói trong một đơn vị.
- Kế thừa: Các lớp có thể được sử dụng lại và mở rộng.
- Đa hình: Chức năng có cùng tên có thể có các hoạt động khác nhau.
- Các lớp chung: Được sử dụng cho dữ liệu chưa được phân loại.
- Thư viện lớp: Bộ sưu tập các lớp tích hợp để cung cấp các chức năng cơ bản.

- Truyền tin nhắn: Các đối tượng có thể tương tác với nhau bằng cách gửi và nhận dữ liệu..

3. Event-driven paradigm

Event-driven

paradigm

Event-driven paradigm Mô hình hướng sự kiện xoay quanh các sự kiện như nhấp chuột, nhấn bàn phím hoặc gửi tin nhắn. Nó được sử dụng hầu như chỉ trong lập trình giao diện người dùng đồ họa (GUI) hoặc các ứng dụng tập trung vào việc đáp ứng các đầu vào của người dùng. Nó sử dụng các chức năng gọi là "xử lý sự kiện" để chặn các sự kiện được nhắm mục tiêu và tạo ra phản hồi tương ứng với các sự kiện này.

Các tính năng chính của mô hình hướng sự kiện bao gồm các trình xử lý sự kiện được đề cập, chức năng kích hoạt, sự kiện, định hướng dịch vụ, điều khiển thời gian, v.v..

..

III. Integrated Development Environment

Integrated Development Environment (IDE)

Integrated Development Environment (IDE) is a software that provides a wide range of functionalities, utilities and facilities to programmers for software development. Most IDEs support multiple programming languages, with a few IDE only focusing on one certain language (e.g. MariaMole for Arduino, Judo for Java, or Zend Studio for PHP). Common features of an IDE may include:

- **Source code editor:** Basically a text editor program with extended features to help programmers type and edit their source code. Most of the work of a programmer is done in the source code editor. The source code editor provides functions and utilities to simplify and enhances the programmer's coding experience. For example, the NetBeans IDE has the "Auto Format" feature, which allows the user to apply indentations, position brackets, and remove empty lines with a single click. Another very important feature is "Auto complete". The programmer doesn't have to remember every single class required by the project. With only a few letters typed, the editor automatically suggests potential classes and methods for the programmer to choose, and even automatically import them from their packages, saving the programmer a ton of time. The editor also has templates to generate repetitive snippets of code such as constructors or getters/setters.

S.

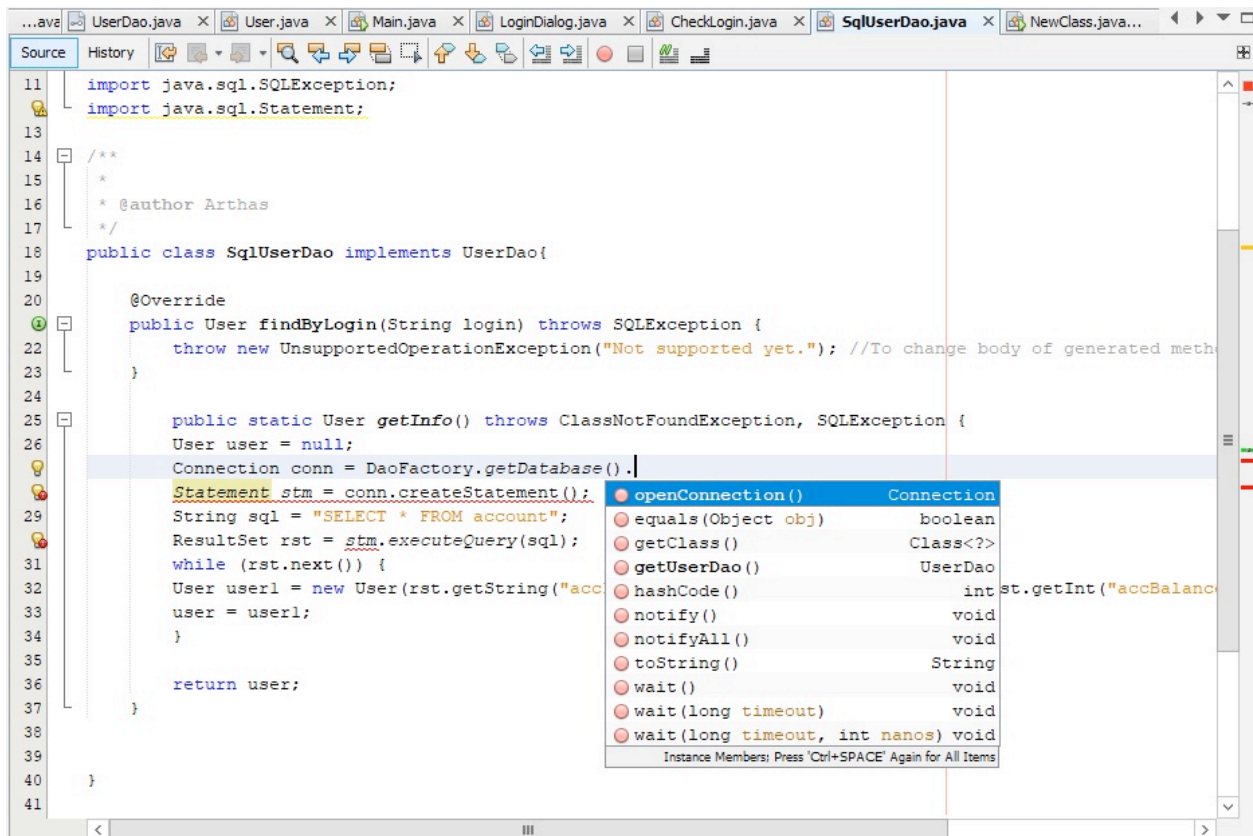


Figure 3. NetBeans IDE's auto-completion feature in action

- **Compiler/interpreter:** Based on the programming languages supported, an IDE usually integrates the matching number of compilers for the corresponding languages in order to help the users translate their codes into machine language and execute them.
- **Debugger:** Debugging is the process of identifying and eliminating bugs (errors) in the source code of a program. It is a vital part of programming, therefore, developers of many IDEs have incorporated a tool to help programmers in debugging, called "debugger". Typically, upon compiling or executing, if the program reaches a point where an error occurs, then the debugger is called, identifying the origin of the error. Alternatively, the user can choose to apply techniques such as step by step running or breakpoints to determine the exact event that caused the error.
- **Project Management:** Each program should have its own space to store the files and configurations. The IDE provide a tool to view and manage each project effectively.

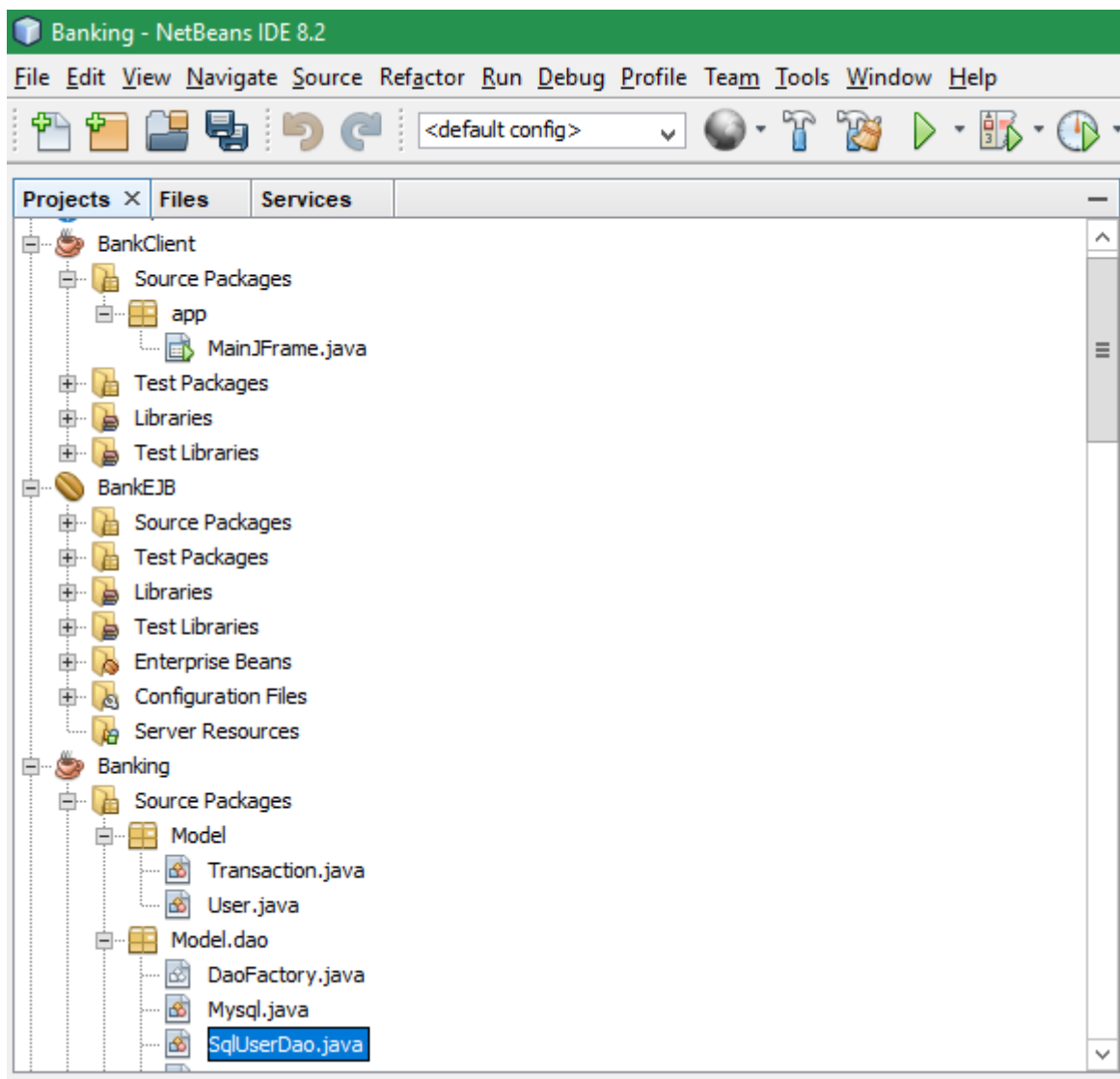


Figure 4. NetBeans IDE's Project Management Panel

Conclusion

T

Tóm lại, báo cáo đã cung cấp định nghĩa và đặc điểm của thuật toán, cũng như các bước xây dựng một ứng dụng. Ngoài ra, một ví dụ về chất lượng thuật toán cũng được thể hiện. Báo cáo cũng thảo luận về một số mô hình lập trình và các đặc điểm chính của chúng. Cuối cùng, một phân tích về Môi trường phát triển tích hợp được cung cấp, mà nhóm phát triển có thể sử dụng để nâng cao hiệu quả và hiệu quả của các dự án của mình..

..

References

Anon., n.d. *algolist.net*. [Online]

Available at: <http://www.algolist.net/img/sorts/bubble-sort-2.png>

[Accessed March 2018].

Anon., n.d. *Netbeans Features*. [Online]

Available at: <https://netbeans.org/features/>

[Accessed March 2018].

Prata, S., 2004. In: *C Primer Plus*. s.l.:Sams, pp. 6-8.

Schneider, M. & J. Gersting, 1995. In: *An Invitation to Computer Science*. New York: West Publishing Company, p. 9.

Techopedia, n.d. *Techopedia*. [Online]

Available at: <https://www.techopedia.com/definition/3739/algorithm>

[Accessed March 2018].

Wikipedia, n.d. *Integrated development environment*. [Online]

Available at: https://en.wikipedia.org/wiki/Integrated_development_environment

[Accessed March 2018].