TRƯỜNG ĐẠI HỌC XÂY DỰNG HÀ NỘI

**KHOA CÔNG NGHỆ THÔNG TIN**

----- 🙡 🕮 🙣 -----

Icon

Description automatically generated

**BÁO CÁO ĐỒ ÁN**

**Lập trình hướng đối tượng**

***Đề tài:* Bài toán tìm đường đi ngắn nhất**

Giảng viên: **Lê Đức Quang**

Nhóm sinh viên thực hiện: 01

|  |  |  |
| --- | --- | --- |
| STT | Họ và tên | MSSV |
| 1 | Mai Ngọc Đoàn | 1509765 |
| 2 | Nguyễn Mai Nhật Linh | 1529265 |

**Hà Nội, năm 2022**

**LỜI NÓI ĐẦU**

Công nghệ 4.0, cuộc cách mạng công nghiệp thứ 4 hay cách mạng 4.0 là những cái tên được nhắc đến nhiều nhất trên các diễn đàn về khoa học, kỹ thuật, kinh tế, y tế, giáo dục, công nghiệp… Nói một cách khác, công nghệ 4.0 bùng nổ vào đầu thế kỷ 21 kéo theo phát triển bùng nổ của Công nghệ Thông tin. Nhiều ngành nghề, dịch vụ, thị trường cũng được công nghệ hóa để phù hợp với nhu cầu sử dụng của xã hội ví dụ như ngành giáo dục – giáo dục điện tử, tài chính – tài chính điện tử, ngân hàng – ngân hàng điện tử… đều được lồng ghép công nghệ, hiện đại hóa để trở nên ngày một tiện dụng. Công nghệ thông tin giữ vai trò quan trọng ở mọi quốc gia trên thế giới, trong đó có Việt Nam. Đây được xem là nền tảng vững chắc nhằm phục vụ cho tiến trình phát triển đất nước bền vững, đặc biệt là trong đời sống xã hội hiện đại.

Trong đời sống, chúng ta thường bắt gặp tình huống như sau: để đi từ địa điểm A đến địa điểm B có rất nhiều đường đi, nhiều cách đi, nhưng ta sẽ luôn muốn chọn đường đi nào sao cho ngắn nhất (theo nghĩa cự ly), nhanh nhất (theo nghĩa thời gian) và hao tốn ít chi phí nhất… Vì vậy để đáp ứng nhu cầu đó và thực hiện tin học hoá trong việc giải quyết các vấn đề trong xã hội, nhóm chúng em quyết định chọn đề tài “ Bài toán tìm đường đi ngắn nhất” áp dụng thuật toán Dijkstra và các kiến thức được học trong quá trình học tập tại nhà trường để tạo ra được 1 ứng dụng có tác dụng trong thực tế.

Mặc dù đã rất cố gắng để hoàn thành đề tài, xong thời gian có hạn và kiến thức còn hạn hẹp nên đồ án của nhóm chúng em vẫn còn nhiều sai sót. Vì vậy, chúng em mong nhận được ý kiến đóng góp của thầy cô và các bạn để đồ án hoàn thiện tốt hơn

Cuối cùng, nhóm chúng em xin chân thành cảm ơn thầy Lê Đức Quang, giảng viên bộ môn Phương pháp lập trình hướng đối tượng đã tận tình chỉ bảo hướng dẫn chúng em hoàn thành đề tài này

Xin Chân Thành Cảm Ơn!

**MỤC LỤC**

[**Chương I: Giới thiệu chung** 4](#_Toc123510854)

[1. Đặt ra bài toán 4](#_Toc123510855)

[2. Phạm vi 4](#_Toc123510856)

[**Chương II. Công nghệ sử dụng** 4](#_Toc123510857)

[1. Ngôn ngữ Java 4](#_Toc123510858)

[2. Các hình mẫu thiết kế (Design Pattern) 5](#_Toc123510859)

[2.1 MVC 5](#_Toc123510860)

[2.2 Singleton 5](#_Toc123510861)

[2.3 Observer 5](#_Toc123510862)

[3. SQL Server 5](#_Toc123510863)

[**Chương III. Mô tả bài toán** 5](#_Toc123510864)

[1. Mô tả khái quát 5](#_Toc123510865)

[2. Mô tả chi tiết 6](#_Toc123510866)

[2.1 Thuật toán Dijkstra (Dijkstra Algorithm) 6](#_Toc123510867)

[2.2 Phân tích cấu trúc dữ liệu 6](#_Toc123510868)

[2.3 Các sơ đồ thiết kế 6](#_Toc123510869)

[2.3.1 Sơ đồ Usecase 6](#_Toc123510870)

[2.3.2 Sơ đồ Package 8](#_Toc123510871)

[2.3.3 Sơ đồ Class 9](#_Toc123510872)

[2.3.4 Component Diagram 11](#_Toc123510873)

[2.3.5 Sơ đồ thiết kế bảng dữ liệu trong SQL 12](#_Toc123510874)

[**Chương IV. Mô tả ứng dụng** 13](#_Toc123510875)

[1. Cài đặt ứng dụng 13](#_Toc123510876)

[2. Source Files 13](#_Toc123510877)

[3. Chạy chương trình 15](#_Toc123510878)

# Chương I: Giới thiệu chung

## 1. Đặt ra bài toán

Xây dựng ứng dụng “Tìm đường đi ngắn nhất” để giải quyết các vấn đề trong việc di chuyển từ hai địa điểm sao cho nhanh nhất và ít hao tốn chi phí nhất.

## 2. Phạm vi

Ứng dụng được sử dụng bởi bất kì ai mong muốn tìm ra con đường tối ưu nhất và ngắn nhất có thể.

# Chương II. Công nghệ sử dụng

## 1. Ngôn ngữ Java

- Java được biết đến là ngôn ngữ lập trình hướng đối tượng (OOP) phổ biến, dễ sử dụng và triển khai. Java được phát triển bởi Sun Microsystems, do James Gosling khởi xướng và ra mắt năm 1995. Java hoạt động trên rất nhiều nền tảng như Windows, Mac và các phiên bản khác nhau của UNIX. Thêm vào đó, Java còn hỗ trợ SWING – một bulit – in library dùng cho việc thiết kế giao diện ứng dụng (GUI).

- Ứng dụng của Java

+ Desktop App như media player, acrobat reader, antivirus, …

+ Web App như javatpoint.com, irctc. co.in, …

+ Enterprise App như một vài ứng dụng về xử lý các nghiệp vụ ngân hàng, …

+ Thiết bị Mobile như các ứng dụng IOS hay Android.

+ Hệ thống nhúng.

+ Smart Card.

+ Robot.

+ Game App.

- Ưu điểm của ngôn ngữ lập trình Java

+ Java là ngôn ngữ độc lập với nền tảng vì chúng ta có thể chạy mã Java ở bất kỳ máy nào mà không cần phần mềm đặc biệt.

+ Là ngôn ngữ hướng đối tượng làm tăng khả năng phát triển mã dễ dàng và tăng hiệu quả.

+ Là ngôn ngữ an toàn vì Java không sử dụng con trỏ.

+ Quản lý bộ nhớ hiệu quả.

+ Hỗ trợ đa luồng, có thể thực hiện cùng lúc nhiều chương trình.

+ Có nhiều tính năng: tự động thu gom rác, không sử dụng con trỏ, xử lý ngoại lệ…

-Giới thiệu về Swing

Java Foundation Classes (JFC) được được giới thiệu trong phiên bản 2.0 (Java Development Kit – JDK 2.0), là một framework hỗ trợ lậptrình giao diện đồ hoạ (Graphical Interface) với thư viện Swing

Swing là một framework được thiết kế theo mô hình MVC (Model ViewController), hỗ trợ công nghệ gọi là “Pluggable-Look-And-Feel” cho phép các thành phần giao diện có thể được hiển thị trên bất ký hệ điều hành nào như Windows, Mac OS, Linux

## 2. Các hình mẫu thiết kế (Design Pattern) và cơ sở dữ liệu

### 2.1 MVC

MVC là viết tắt của cụm từ “Model-View-Controller“. Đây là mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia thành ba phần được kết nối với nhau và mỗi thành phần đều có một nhiệm vụ riêng của nó và độc lập với các thành phần khác, trong đó phần Model sẽ đảm nhiệm chức năng liên hệ với Database, View sẽ đảm nhiệm việc tương tác với người dùng, phần thực thi thuật toán Dijkstra sẽ giao cho Controller.

### 2.2 Singleton

Đây là hình mẫu kinh điển cho những bài toán sử dụng Database. Khi đó, nó sẽ đảm bảo chỉ có một biến Connection kết nối đến Database được tạo ra trong suốt quá trình chạy của chương trình.

### 2.3 Observer

Observer pattern là một hình mẫu thiết kế phần mềm mà một đối tượng, gọi là subject, duy trì một danh sách các thành phần phụ thuộc nó, gọi là observer, được sử dụng để theo dõi trạng thái của một object nhất định, hầu hết thời gian và trạng thái thay đổi của một object sẽ ảnh hưởng đến trạng thái của phần còn lại. Vì vậy, phải có một hệ thống ghi nhận lại những thay đổi, và thông báo cho các object khác.

## 3. SQL Server

SQL server là một loại phần mềm đã được phát triển bởi Microsoft và nó được sử dụng để có thể dễ dàng lưu trữ dữ liệu dựa theo tiêu chuẩn RDBMS. Tuy nhiên, trong các bài toán thiên về thuật toán như thế này, nhóm chúng em nhận định không cần dùng quá nhiều đến Database, mục đích của Database chỉ cần để lưu trữ những thông tin cơ bản của đồ thị.

# Chương III. Mô tả bài toán

## 1. Mô tả khái quát

Xây dựng chương trình giải quyết bài toán tìm đường đi ngắn nhất trong một đồ thị: *Cho trước một đồ thị gồm V đỉnh, E cạnh và một hàm trọng số giá trị thực f: E → R. Nếu cho trước một đỉnh v E, hãy tìm đường P để đi từ v đến một đỉnh v’ nào đó sao cho: ∑ p∈P f (P) là nhỏ nhất trong số tất cả các con đường nối v với v’. Có thể sử dụng một trong các thuật toán sau để giải quyết bài toán này (thuật toán Dijkstra, thuật toán Bellman-Ford, hoặc thuật toán heuristics)*

Như vậy, để giải quyết bài toán đặt ra, chúng ta có nhiều cách chọn thuật toán để giải quyết, ở đây nhóm chúng em chọn ***Dijkstra*** để làm thuật toán chính trong đề tài này.

## 2. Mô tả chi tiết

### 2.1 Thuật toán Dijkstra (Dijkstra Algorithm)

Thuật toán Dijkstra, mang tên của nhà khoa học máy tính người Hà Lan EdsgerDijkstra vào năm 1956 và ấn bản năm 1959. Đây là thuật toán giúp duyệt cây đồ thị và tìm đường đi có giá cực tiểu từ một điểm đến một điểm bất kì khác. Thuật toán này có nhiều dạng biến thể khác nhau. Thuật toán gốc ban đầu sẽ cho biết đường đi ngắn nhất giữa 2 điểm cho trước còn biến thể phổ biến của nó là tìm đường đi ngắn nhất từ một điểm đến tất cả các điểm còn lại, tạo ra một cây lời giải chứa các đường đi đó (shortest – path tree). Thuật toán thường được sử dụng trong định tuyến với một chương trình con trongcác thuật toán đồ thị hay trong công nghệ Hệ thống định vị toàn cầu (GPS).

Ở đồ án của nhóm, chúng em thiên về việc tìm đường đi giữa chỉ 2 điểm cho trước mà không tạo ra toàn bộ cây lời giải đến tất cả các điểm khác, hay nói cách khác, đây chính là biến thể gốc ban đầu của Dijkstra.

### 2.2 Phân tích cấu trúc dữ liệu

Để giải quyết bài toán đồ thị vào trong lập trình, điều đầu tiên đó là biểu diễn được đồ thị đó thông qua các cấu trúc dữ liệu. Để biểu diễn một điểm trong đồ thị, ta sử dụng cấu trúc sau:

public class Node implements Comparable<Node> {

private final String name;

private int estimate;

private final HashMap<Node, Integer> adjacentNodes;

// methods below

}

Nguồn: Dijkstra\src\main\java\huce\Algorithm\Node\Node.java

Với:

* name: lưu trữ tên của đỉnh đó
* estimate: dùng cho thuật toán Dijkstra, biểu diễn cho chi phí từ điểm Source đến chính nó bằng bao nhiêu.
* adjacentNodes: lưu trữ các Node là con (direct – children) của nó.

Với cách xây dựng như thế này, ta sẽ có một đồ thị một chiều (directed – graph), tức là nếu B là con của A, sẽ có đường nối từ A đến B, điều ngược lại chưa chắc đúng.

### 2.3 Các sơ đồ thiết kế

#### 2.3.1 Sơ đồ Usecase

Diagram

Description automatically generated

#### 2.3.2 Sơ đồ Package

Diagram

Description automatically generated

#### 2.3.3 Sơ đồ Class

Diagram, engineering drawing, schematic

Description automatically generated

Trong đó:

* *Dijkstra:* là class chứa các logic thuật toán Dijkstra
* *Parser* : Dùng để biến đổi dạng file \*.dot (một file dùng để biểu diễn đồ thị) sang thành một dãy các *Node* có mối quan hệ tương đương.
* *AppDB:* thực hiện kết nối đến cơ sở dữ liệu. Class này được implement theo mô hình thiết kế Singleton.
* *GraphvizFileFormatException:* một exception, xảy ra nếu file \*.dot đầu vào có dữ liệu không hợp lệ.

Dữ liệu đầu vào là hợp lệ khi nó có dạng:

Table

Description automatically generated

Với *graph* là tên loại kiểu đồ thị, nó có thể là graph (hai chiều) hay digraph (một chiều). Tìm hiểu thêm về file \*.dot tại: [DOT (graph description language) - Wikipedia](https://en.wikipedia.org/wiki/DOT_(graph_description_language))

* *NoDataException:* Gần giống chức năng như *GraphvizFileFormatException* nhưng nó sẽ báo lỗi khi file đầu vào không có dữ liệu hoặc graph rỗng.
* *PathNotFoundException:* Nếu trong quá trình tìm đường đi từ S tới G, khi duyệt hết tất cả phần tử mà không tìm thấy đường nối giữa S và G thì sẽ sinh ra Exception này.
* *Controller:* Một abstract class dùng để them các sự kiện người dùng cho dứng dụng chính, từ mẫu Controller này sẽ viết các class khác kế thừa từ nó để “control” cho một sự kiện cụ thể.
* *Subject và Observer:* Hai interface chính cho hình mẫu Observer (Observer Pattern). Việc thiết kế hình mẫu này là để kết nối dữ liệu giữa 2 JFrame: *MainApp* và *FileSelectorView.* MainApp sẽ implement Subject và FileSelectorView sẽ kế thừa Observer. Khi người dùng chọn được file mình mong muốn để làm dữ liệu đầu vào, FileSelectorView (Observer) sẽ kích hoạt hàm update của MainApp (Subject) và truyền file vừa chọn cho MainApp thực hiện việc đọc dữ liệu.

#### 2.3.4 Component Diagram

Diagram

Description automatically generated

#### 2.3.5 Sơ đồ thiết kế bảng dữ liệu trong SQL

Dữ liệu chỉ chứa trong một bảng duy nhất

* Sơ đồ ER

Diagram

Description automatically generated

* Sơ đồ ERD:

Table

Description automatically generated

# Chương IV. Mô tả ứng dụng

## 1. Cài đặt ứng dụng

* Ứng dụng được viết bằng ngôn ngữ Java, Jre18
* Yêu cầu để chạy:
* Compile và Run trên nền tảng IntelliJ IDEA.
* JRE18
* Database dựa theo sơ đồ thiết kế ở trên (mục I – 2.3.5).

## 2. Source Files

Có các cách để download Project online sau:

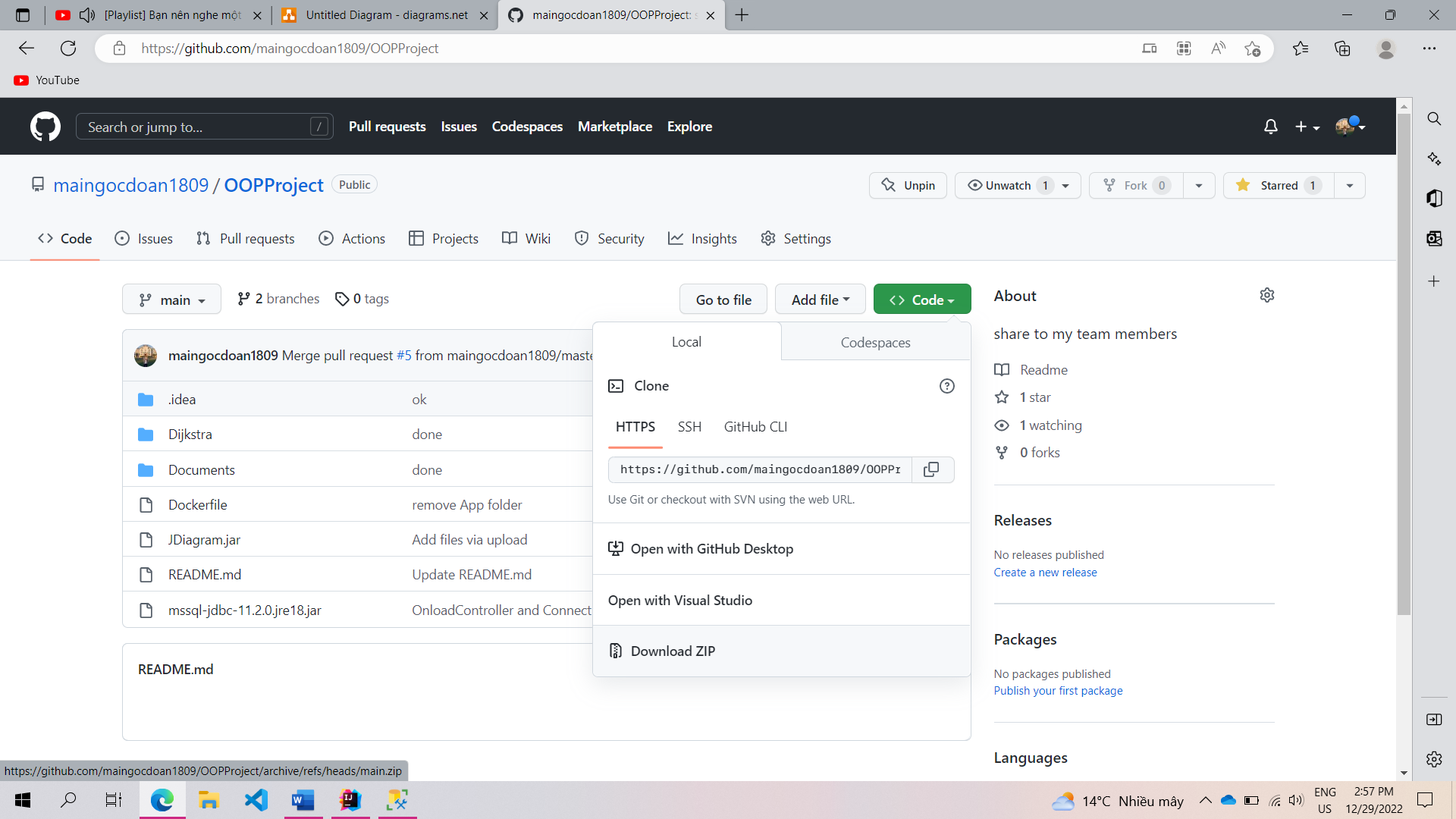
Cách 1: Dùng git và github:

* Tải file zip

B1, Truy cập: <https://github.com/maingocdoan1809/OOPProject>

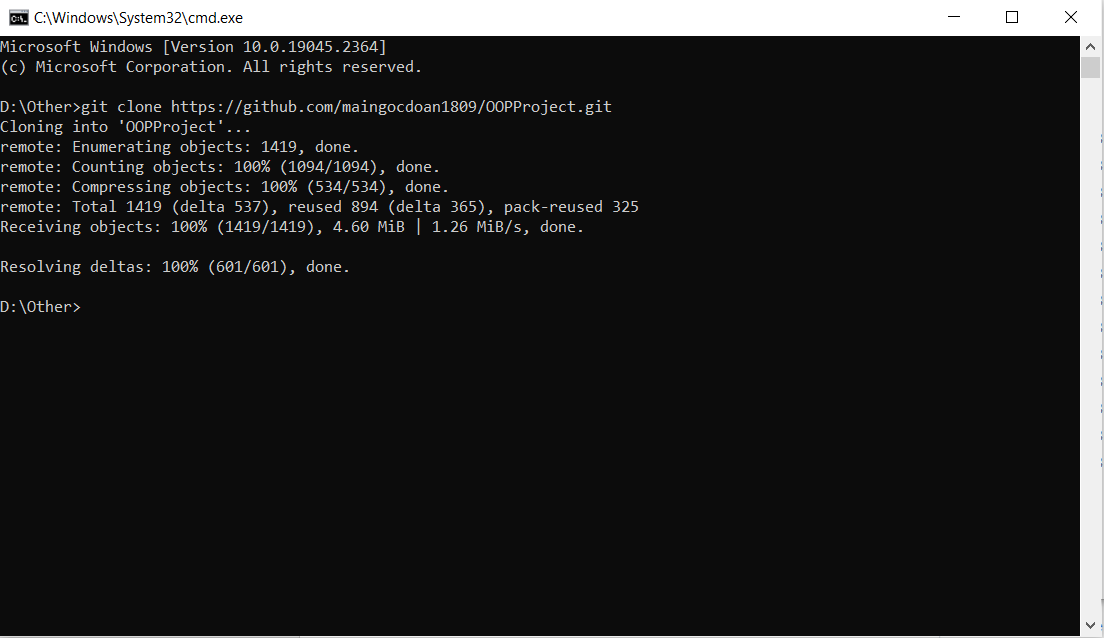
B2: Có 2 cách download:

+ Nhấn vào nút <> Code màu xanh, chọn Download ZIP và tải về máy:



* Clone bằng git local trên máy tính, vào command prompt trên máy tính, target đến thư mục cần lưu project và nhập:

git clone <https://github.com/maingocdoan1809/OOPProject.git>



Nếu màn hình xuất hiện như thế này thì là đã download thành công.

Cách 2: Download qua google Drive:

Truy cập vào <https://m5.gs/bE94M2> để tải.

## 3. Chạy chương trình

* Giao diện chính

Graphical user interface, application

Description automatically generated

-Giao diện chọn dữ liệu đầu vào:

Trong bảng Available testcases, chọn 1 graph đầu vào mà bạn đã insert sẵn trong SQL server.

Graphical user interface, text, application

Description automatically generated

Click vào *Choose* để thực hiện import.

- Sau khi chọn xong, giao diện kết quả là:

Graphical user interface

Description automatically generated with medium confidence

Ta có thể thực hiện thay đổi cách bố trí các node, màu sắc của chúng, cũng như độ thu phóng to nhỏ bằng thanh công cụ ở đây (Tùy ý):

Graphical user interface, text, application

Description automatically generated

* Sau khi điều chỉnh các thông số xong, tiến hành chọn điểm bắt đầu và điểm kết thúc ở panel bên trái, nếu thay đổi Source thì vào nút *From/Root,* nếu thay đổi điểm đích Goal/Dest thì vào *To*

Graphical user interface

Description automatically generated with medium confidence

* Nếu bạn muốn đồ thị không duyệt qua một nút bất kì thì chọn vào ô *Block*

Ví dụ bạn muốn bỏ qua node C, nhấn chọn nút C ở ô *Block* và nhấn nút Block ở dưới cùng, Node đó sẽ được thêm vào bảng. Bạn có thể chỉnh màu cho nó cho dễ phân biệt ở thanh công cụ.

A picture containing graphical user interface

Description automatically generated

* Nhấn start để bắt đầu tìm đường đi

A picture containing graphical user interface

Description automatically generated

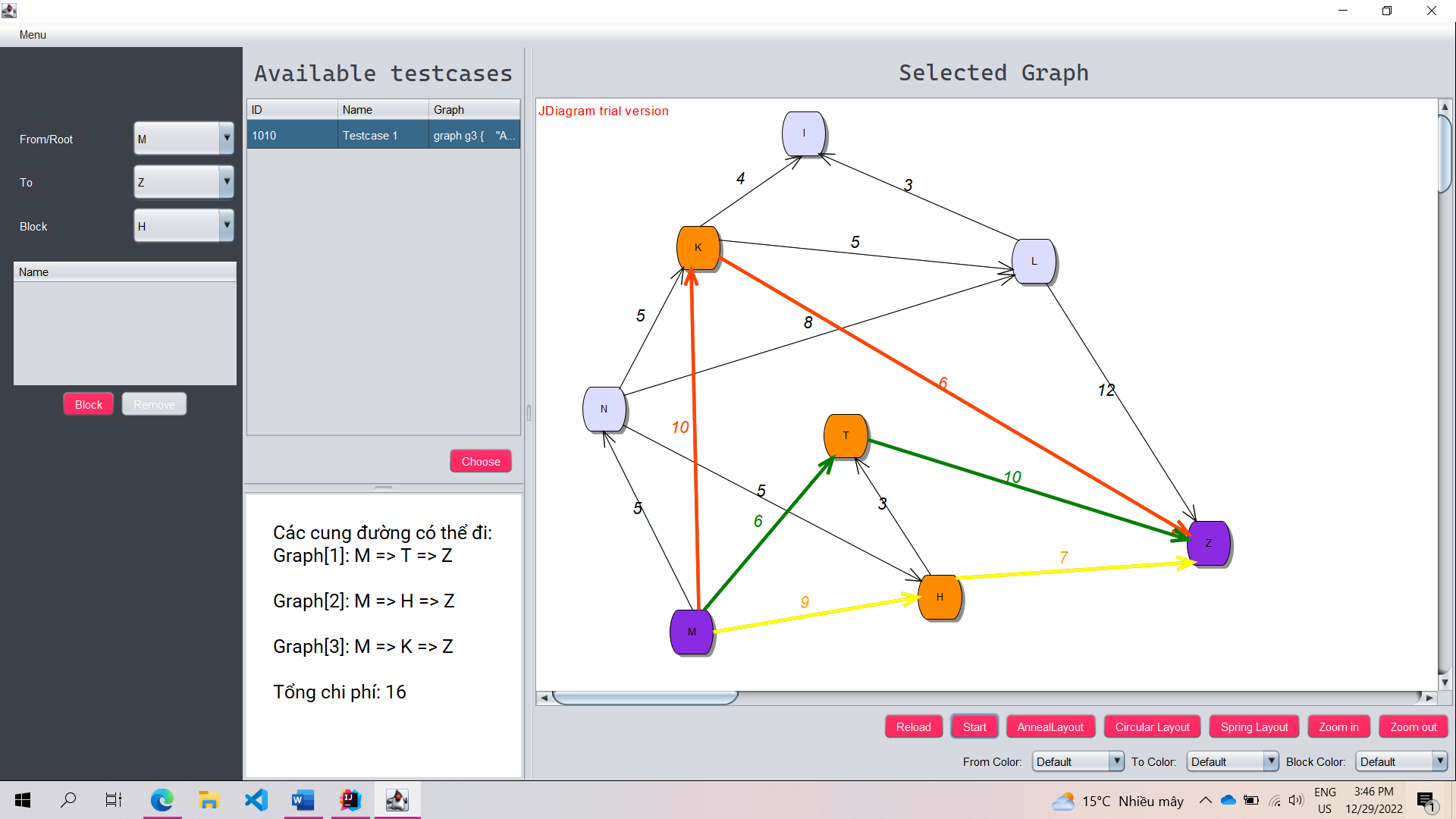
Đường đi sẽ hiện lên trên màn hình bằng đường nối có màu. Có thể có 3 màu, trong đó:

- Màu xanh biểu hiện cho đoạn đường tốt nhất, nhưng đi qua ít đỉnh nhất.

- Màu vàng biểu hiện cho đoạn đường tốt nhất, nhưng đi qua nhiều điểm hơn hoặc bằng màu xanh

- Màu vàng cam biểu hiện cho đoạn đường tốt nhất, nhưng đi qua nhiều hơn hoặc bằng màu vàng.

- Ví dụ trường hợp đó:



* Kết quả bài toán sẽ hiện ở góc dưới bên trái, cạnh panel bên trái

Graphical user interface, application

Description automatically generated

* Nếu bạn muốn thay đổi đích đến, chọn một đỉnh trong nút *To,* và không muốn nó đi qua đỉnh C:

A picture containing application

Description automatically generated

* Block C:

A picture containing application

Description automatically generated

* Start:

A picture containing graphical user interface

Description automatically generated

* Xóa C ra khỏi danh sách chặn:

Graphical user interface, application

Description automatically generated

* Thử nhấn start lại lần nữa:

A picture containing application

Description automatically generated  
Có thể thấy, nếu bỏ C ra khỏi danh sách chặn của A, đường đi bây giờ đã thay đổi.

* Nếu bạn muốn xem lại thì có thể nhấn nút Restart rồi Start, hoặc đơn giản là Start.

Nếu muốn thực hiện truy nhập bằng file, hãy chọn Menu ở phía trên góc trái màn hình:

A picture containing application

Description automatically generated

* Cửa sổ chọn file sẽ hiện ra, chọn đến file bạn muốn chọn và nhấn ok. Lưu ý, cửa sổ chọn file chỉ xuất hiện các file dạng .txt và .dot.

Graphical user interface, application, Word

Description automatically generated

* Nhấn open, ta thu được đồ thị của file đó, các bước thực hiện tiếp theo tương tự như chọn ở Database.

Application

Description automatically generated with low confidence

# V, Phân chia công việc trong nhóm

Bảng dưới đây thể hiện việc phân công công việc trong nhóm:

|  |  |  |
| --- | --- | --- |
| Công việc | Mai Ngọc Đoàn | Nguyễn Mai Nhật Linh |
| Code | x |  |
| Tạo testcases |  | x |
| Làm báo cáo | x | x |
| Usecase | x | x |
| Class Diagram | x | x |
| Other diagrams |  | x |