**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH  
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO  
NGÀNH CÔNG NGHỆ THÔNG TIN**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**BÁO CÁO ĐỒ ÁN 1**

**PHẦN MỀM MÔ PHỎNG TÌM ĐƯỜNG ĐI NGẮN NHẤT GIỮA 2 ĐIỂM**

**GVHD: ThS. Trần Công Tú**

**SVTH: Phạm Thế Anh - 15110009**

**Phan Lâm Trường Hải – 15110039**

**Tp. Hồ Chí Minh, tháng 12 năm 2017**

# Mục lục

[Mục lục 2](#_Toc501344898)

[1 Mô tả phần mềm đồ án 4](#_Toc501344899)

[1.1 Mục đích sử dụng của phần mềm: 4](#_Toc501344900)

[2 Quá trình xây dựng phần mềm 4](#_Toc501344901)

[2.1 Thiết kế giao diện 4](#_Toc501344902)

[3 Các lớp chính cho thiết kế code 7](#_Toc501344903)

[Table 3.1 Mô tả các lớp chính trong chương trình 8](#_Toc501344904)

[Table 3.2 Các phương thức hỗ trợ trong lớp DrawDoThi 9](#_Toc501344905)

[Table 3.3 Các phương thức hỗ trợ trong lớp ShapePoint 9](#_Toc501344906)

[Table 3.4 Các phương thức hỗ trợ trong lớp DataPerStepForRunAutomatically 9](#_Toc501344907)

[Table 3.5 Các phương thức hỗ trợ trong lớp Matrix 10](#_Toc501344908)

[Table 3.6 Các phương thức hỗ trợ trong lớp FunctionUtils 11](#_Toc501344909)

[4 Mô tả phân công công việc 11](#_Toc501344910)

[Table 4.1 Mô tả phân công công việc 13](#_Toc501344911)

[5 Kết luận 13](#_Toc501344912)

[5.1 Những khó khăn gặp phải và cách khắc phục 13](#_Toc501344913)

[5.2 Ưu và khuyết điểm của đồ án 14](#_Toc501344914)

[5.3 Mức độ hoàn thành và định hướng phát triển 14](#_Toc501344915)

[6 Phụ lục 15](#_Toc501344916)

[6.1 Hướng dẫn cách cài đặt và khởi chạy phần mềm (Trên windows) 15](#_Toc501344917)

[6.2 Hướng dẫn sử dụng 21](#_Toc501344918)

[6.3 Tài liệu tham khảo 24](#_Toc501344919)

# Mô tả phần mềm đồ án

## Mục đích sử dụng của phần mềm:

Phần mềm được xây dựng để trình bày, mô tả giải thuật tìm đường đi ngắn nhất của thuật toán Dijkstra giúp những người chưa biết, chưa hiểu về thuật toán Dijkstra có thể hiểu về nó – thuật toán thường được sử dụng trong định tuyến với một chương trình con trong các thuật toán đồ thị hay trong công nghệ Hệ thống định vị toán cầu (GPS).

# Quá trình xây dựng phần mềm

## Thiết kế giao diện

|  |  |  |  |
| --- | --- | --- | --- |
| Thứ tự | Màn hình/Cửa sổ/Dialog | Người thiết kế và giải thích thiết kế của màn hình | Mục đích thiết kế màn hình |
| 1 | Màn hình chính. | Người thiết kế: Phan Lâm Trường Hải.  Giải thích: Giao diện màn hình chính của phần mềm bao gồm các phím chức năng nằm bên trái, đồ thì phía phải, góc phải trên cùng là các phím đóng, thu nhỏ/ mở rộng, tắt phần mềm. | Thiết kế đơn giản, trực quan cho người dùng dễ quan sát, thao tác và dễ dàng hiểu được nguyên lý hoạt động. của phần mềm. |
| 2 | Phím chức năng/ thao tác. | Người thiết kế: Phan Lâm Trường Hải.  Giải thích: \_ Kiểu đồ thị: Đồ thị có hướng hoặc vô hướng. \_ Tự vẽ: Tự thiết kế đồ thị bằng cách thêm điểm, nối, độ dài của đường. \_ Đồ thị có sẵn: Gồm 3 độ thị có sẵn vô hướng và có hướng. \_ Đường đi: Chọn điểm đầu và điểm đích để tìm ra đường đi ngắn nhất. \_ Chạy 1 lần: Ra ngay kết quả đường đi ngắn nhất với đường đi được tô màu. \_ Chạy từng bước: Mô tả từng bước của thuật toán với việc tô màu từng điểm được chọn trên đường đi. \_ Chạy tự động: Tự động chạy thuật toán tìm ra đường đi với thời gian delay tùy chọn. | Thiết kế tối giản để người dùng dễ dàng nhất khi sử dụng, chức năng phong phú để người dùng lựa chọn giúp hiểu rõ về thuật rõ về thuật toán. |
| 3 | Ma trận đồ thị.    Ma trận thuật toán, kết quả đường đi và tổng độ dài. | Người thiết kế: Phan Lâm Trường Hải.  Giải thích: \_ Ma trận đồ thị: thể hiện dữ liệu của đồ thị bao gồm các điểm (Point) và các giá trị đồ dài của các điểm trên đồ thị  \_ Ma trận thuật toán: Thể hiện các bước tìm ra đường đi ngắn nhất từ trên xuống dưới, tô màu đường đi đi qua bao gồm điểm đi qua và tổng giá trị đường đi. | Ma trận đồ thị mà ma trận thuật toán được thiết kế đơn giản, rõ ràng để người dùng có thể dễ dàng quan sát, nắm bắt các bước chạy của thuật toán. |

# Các lớp chính cho thiết kế code

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên lớp | Tên các SV phụ trách viết (Nếu là code lấy từ nguồn khác thì phải chỉ rõ lấy từ nguồn nào). | Mục đích chính của lớp trong chương trình. |
| 1 | Dijkstra. | Thế Anh. | Xử lý thuật toán Dijkstra |
| 2 | DrawDoThi | Thế Anh | Xử lý các thao tác trên đồ thị |
| 3 | ShapeLine | Thế Anh | Cấu hình đương thẳng trong đồ thị |
| 4 | ShapePoint | Trường Hải | Cấu hình điểm trong đồ thị |
| 5 | DataPerStepForDijkstra | Thế Anh | Dữ liệu cho từng bước trong thuật toán Dijkstra |
| 6 | DataPerStepForRunAutomatically | Thế Anh | Dữ liệu cho từng bước trong thuật toán Dijkstra trong trường hợp chạy tự động. |
| 7 | Line | Trường Hải | Lưu trữ dữ liệu của 1 đường thẳng |
| 8 | MPoint | Trường Hải | Lưu trữ dữ liệu của 1 điểm |
| 9 | Matrix | Trường Hải | Lưu trữ thông tin 1 ma trận và chứa phương thức bỗ trợ cho thuật toán |
| 10 | Const | Trường Hải | Lưu trữ các thông số như đồ thị có sẵn, và các hằng số có sẵn như kích thước, màu sắc v. v |
| 11 | FunctionUtils | Trường Hải | Lớp chứa các phương thức giúp chuyển đổi, cập nhật, hỗ trợ cho việc sử lý data. |

*Table 3.1 Mô tả các lớp chính trong chương trình*

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên phương thức hỗ trợ. | Tên các SV phụ trách viết. | Mục đích chính của phương thức trong chương trình. |
| 1 | initPopupMenu | Thế Anh | Khởi tạo các Popup menu. |
| 2 | Draw | Thế Anh | Vẽ các đối tượng ShapeLine, ShapePoint lên panel |
| 3 | drawWithMode | Thế Anh | Vẽ theo chế độ có hướng hay vô hướng |
| 4 | initShapeData | Thế Anh | Khởi tạo giá trị các ShapeLine, ShapePoint |
| 5 | drawRoad | Thế Anh | Vẽ đường đi lúc chạy giải thuật |
| 6 | drawDoThiUnselected | Thế Anh | Vẽ đồ thị lúc chưa chạy thuật toán |

Table 3.2 Các phương thức hỗ trợ trong lớp DrawDoThi

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên phương thức hỗ trợ. | Tên các SV phụ trách viết. | Mục đích chính của phương thức trong chương trình. |
| 1 | Paint | Trường Hải | Tô màu các điểm trên đường đi ngắn nhất. |

Table 3.3 Các phương thức hỗ trợ trong lớp ShapePoint

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên phương thức hỗ trợ. | Tên các SV phụ trách viết. | Mục đích chính của phương thức trong chương trình. |
| 1 | getListPointOfTrueRoad | Trường Hải | Lấy danh sách các điểm trên đường đi ngắn nhất |

Table 3.4 Các phương thức hỗ trợ trong lớp DataPerStepForRunAutomatically

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên phương thức hỗ trợ. | Tên các SV phụ trách viết. | Mục đích chính của phương thức trong chương trình. |
| 1 | createColumnsName | Trường Hải | Tạo tên cột từ listPoint. |
| 2 | getListVariablesVoHuongFromListVariablesCoHuong | Trường Hải | Tạo danh sách các biến vô hướng từ danh sách các biến có hướng. |
| 3 | updateMaTranWithListPoints | Trường Hải | Cập nhật lại ma trận từ danh sách các điểm. |
| 4 | clearMaTran | Trường Hải | Xóa ma trận. |

Table 3.5 Các phương thức hỗ trợ trong lớp Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên phương thức hỗ trợ. | Tên các SV phụ trách viết. | Mục đích chính của phương thức trong chương trình. |
| 1 | convertListVariableToListPoint | Trường Hải | Chuyển đổi từ ma trận tham số 2 chiều thành list Mpoint. |
| 2 | convertListPointsToListVariables | Trường Hải | Chuyển đổi từ list Mpoint sang ma trận tham số 2 chiều. |
| 3 | convertListObjectToListInt | Trường Hải | Chuyển đổi ma trận 2 chiều từ kiểu object thành kiểu int. |
| 4 | convertListDatasToListShape | Trường Hải | Chuyển đổi list Mpoint, Shape thành ShapePoint, ShapeLine. |
| 5 | convertListShapeToListPoint | Trường Hải | Chuyển đổi từ list ShapePoint, ShapeLine thành Mpoint, Line. |
| 6 | updateHeightOfRowInTable | Trường Hải | Cập nhật chiều cao của 1 cái ô trong table. |
| 7 | updateTableWithSelectedCells | Trường Hải | Cập nhật các thông số cấu hình 1 ô trong Table khi mà ô được chọn. |
| 8 | getListPointCanConnected | Trường Hải | Lấy danh sách những điểm có đường đi từ 1 điểm cho trước. |

Table 3.6 Các phương thức hỗ trợ trong lớp FunctionUtils

# Mô tả phân công công việc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tuần | Phạm Thế Anh |  | Phan Lâm Trường Hải |  |
| 1,2,3 | Tìm hiểu về thuật toán Dijkstra và ứng dụng của nó trong việc viết phần mềm tìm đường đi ngắn nhất. | V | Tìm hiểu về thuật toán Dijkstra và ứng dụng của nó trong việc viết phần mềm tìm đường đi ngắn nhất. | V |
| 4 | Thiết kế khái quát khung cho phần mềm (lập danh sách các class và chứng năng của nó). | V | Thiết kế khái quát khung cho phần mềm. | V |
| 5 | Tạo class xử lý xử lý thuật toán Dijkstra trong package Algorithm. | V | Tạo các class đối tượng cần thiết trong package Model. | X |
|  | Tạo giao diện của phần mềm. | V |
| 6 | Xử lý thao tác: + load dữ liệu lấy được từ xử lý thuật toán vào table. + thao tác click 1 chạy lần. | V | Xử lý thao tác: + load ma trận vào table  + chuyển đổi mode của đồ thị (có hướng || vô hướng). + Chọn đường đi. | V |
| V | Tạo các class hỗ trợ cho xử lý dữ liệu + thuật toán trong package Utils. | V |
| 7 | Xử lý thao tác: + Thao tác click chạy từng bước. | V | Cập nhật: + Các đối tượng cần thiết trong package Model. | V |
| Tạo các class Vẽ đối tượng trong package Draw: + ShapeLine. | X | Tạo các class Vẽ đối tượng trong package Draw: + Mshape (Interface chung của các lớp vẽ). + ShapePoint. | X |
| 8 | Cập nhật: +ShapeLine. | V | Cập nhật: +ShapePoint. | V |
| Tạo panel DrawDoThi: Chứa các shape và xử lý khi thao tác chỉnh sửa trên đồ thị. | V |  |
| 9 | Xử lý thao tác: + Nối 2 điểm. + Thêm điểm. + Thay đổi giá trị (line và point). | V | Xử lý thao tác:  + Di chuyển điểm. + Xoá điểm. + Xoá đường thẳng. | V |
| Cập nhật đồ thị với thao tác: + Chạy từng bước. + Chạy 1 lần. | V | Cập nhật đồ thị với thao tác: + Chọn mode đồ thị (có hướng || vô hướng). + Chọn đồ thị (tự vẽ || có sẵn). | V |
| 10 | Thêm tính năng chạy từng bước tự động + có time delay. | V | Cập nhật lại giao diện cho hợp lý. | V |
| 11,12, … | + Tối ưu hoá code. + Fix 1 số tính năng nếu có bug phát sinh. | | | |

Table 4.1 Mô tả phân công công việc

# Kết luận

## Những khó khăn gặp phải và cách khắc phục

1. Khó khăn gặp phải:
   * Xử lý chạy tự động.
   * Hiển thị màu chọn đối với các ô (tương ứng với 1 bước chọn trong thuật toán) trong Table thuật toán.
2. Cách khắc phục:
   * Xử lý chạy tự động:
     + Thêm thời gian delay => sử dụng timer.
   * Hiển thị màu chọn đối với các ô (tương ứng với 1 bước chọn trong thuật toán) trong Table thuật toán:
     + Tạo class CellRendererOfTable để tô màu cho các ô được chọn.

## Ưu và khuyết điểm của đồ án

1. Ưu điểm:

* Đồ án ứng dụng thuật toán, kiến thức đã từng học ở môn “Lý thuyết đồ thị và giải thuật”.
* Phần mềm được thiết kế và xây dựng trên NetBeans.

1. Khuyết điểm:

* Còn thiếu 2 thuật toán xử lý khác là: Floyd và Bellman Ford.
* Chưa có chức năng tạm dừng cho tính năng chạy tự động.
* Chưa có các hướng dẫn cụ thể trong phần mềm.

## Mức độ hoàn thành và định hướng phát triển

1. Mức độ hoàn thiện:

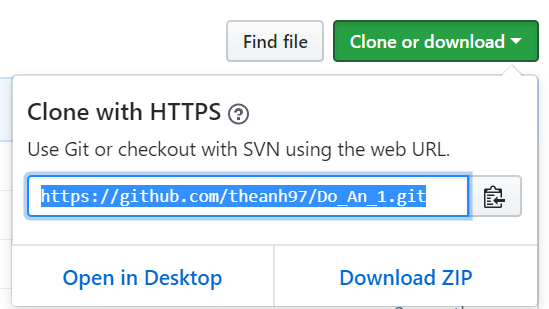
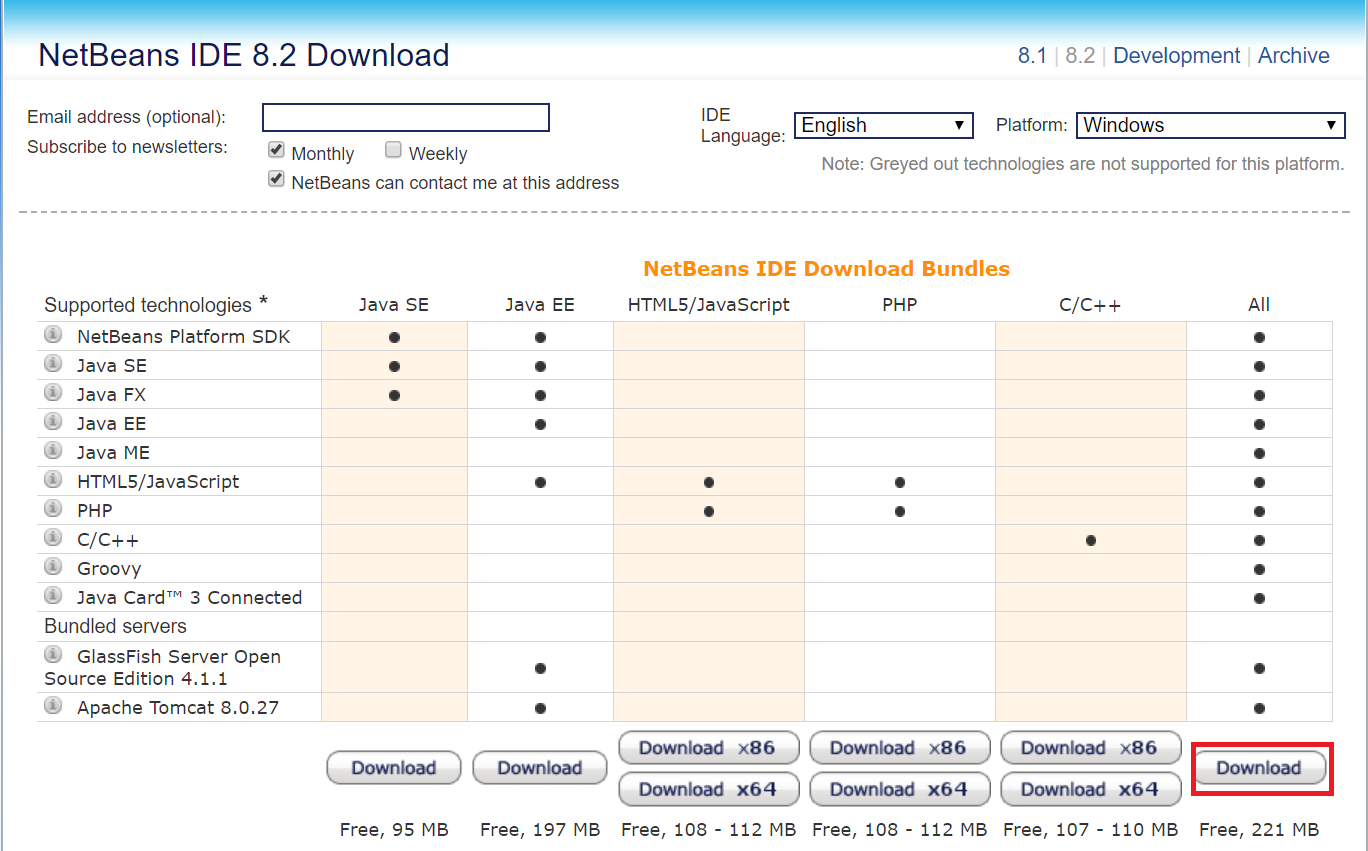
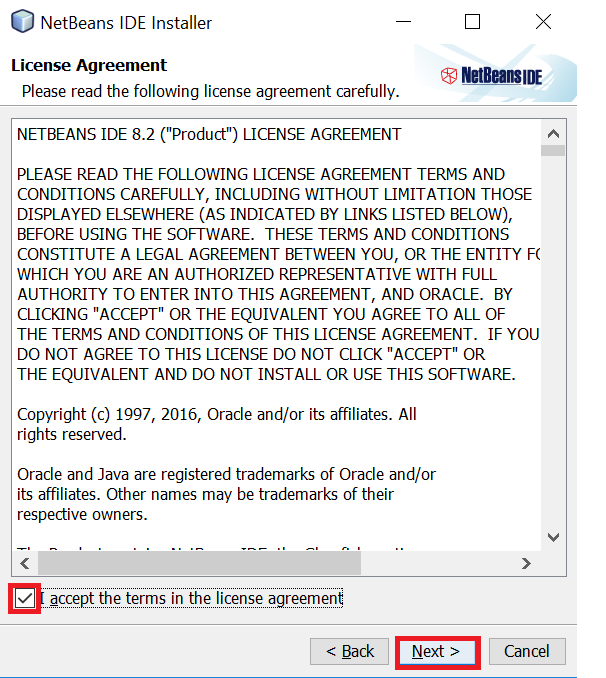
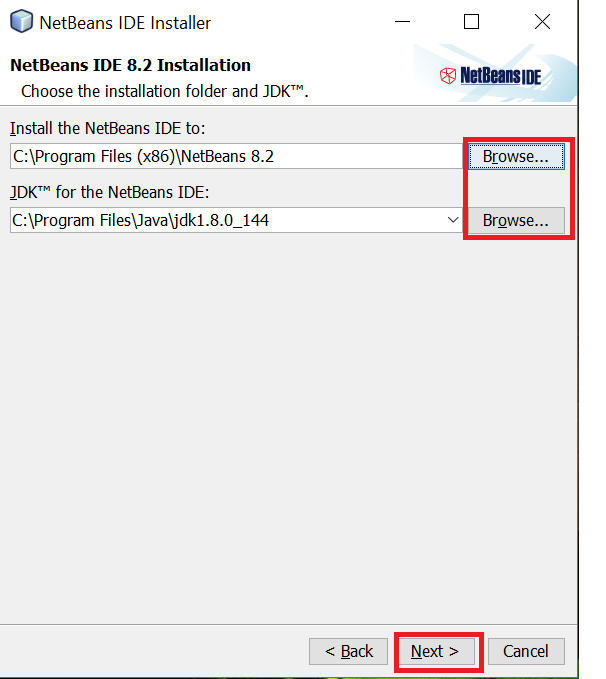
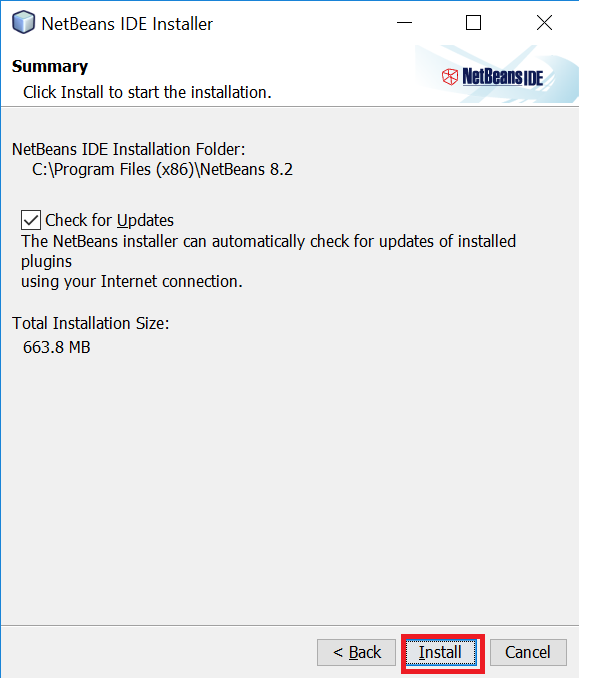
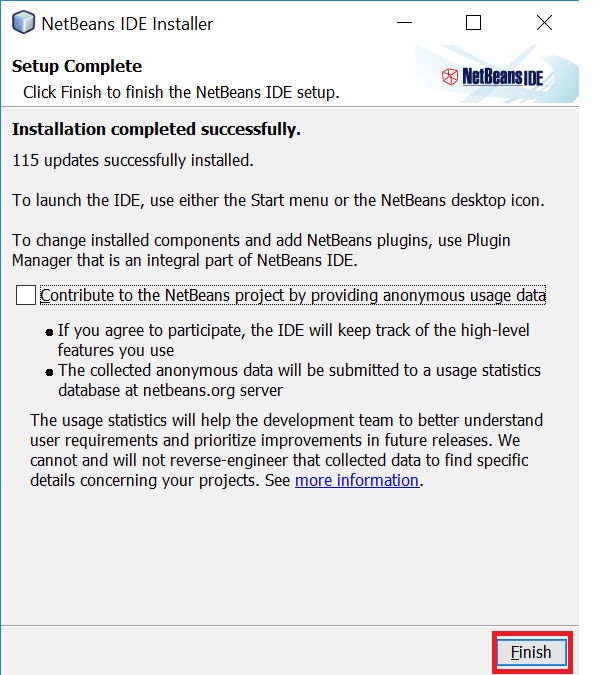
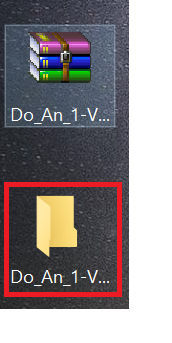
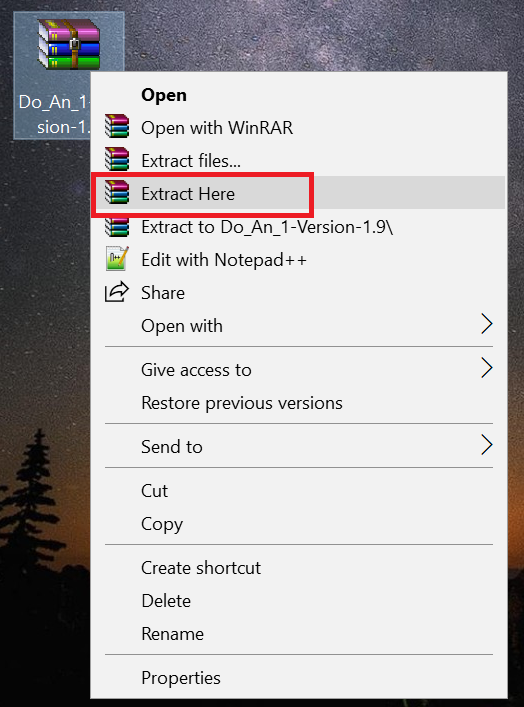
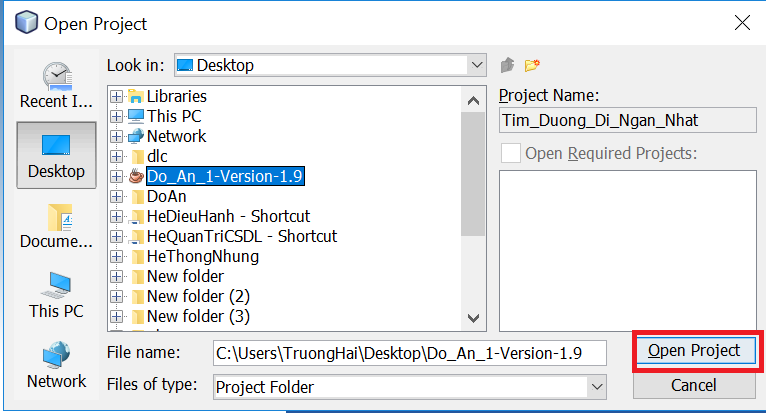
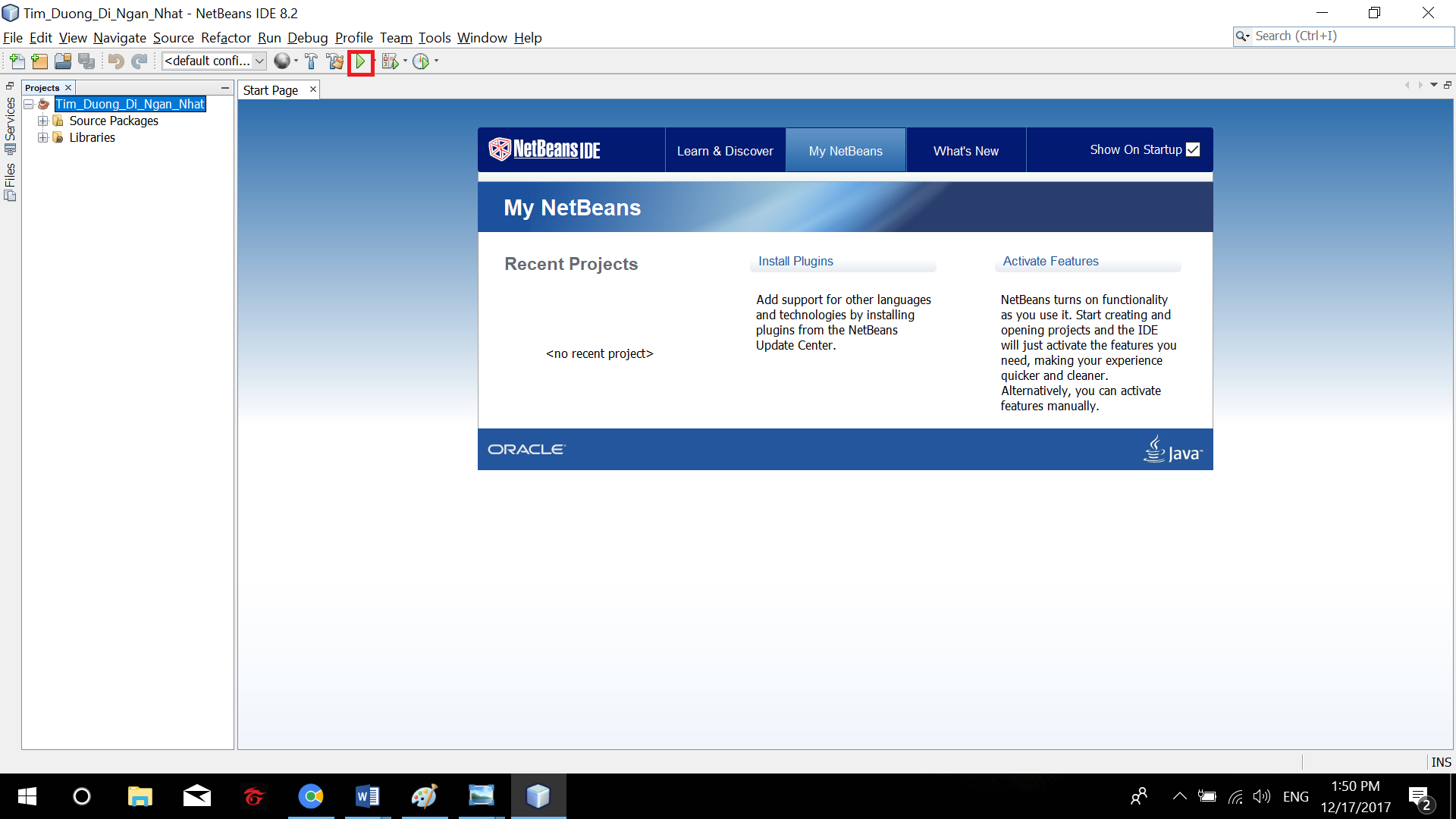
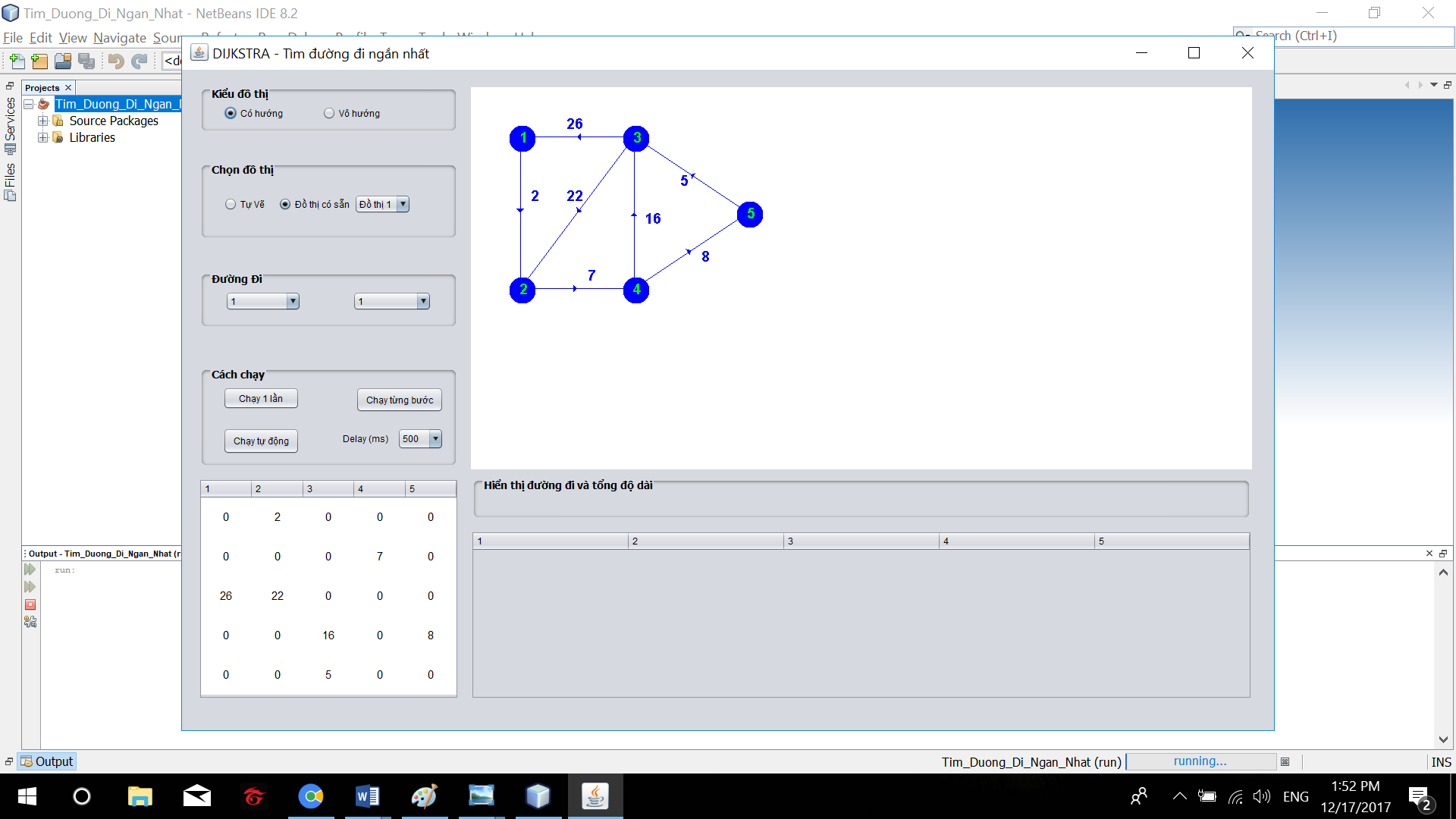
* Sản phẩm đã hoàn thiện được các tính năng chính, cơ bản của đề tài. Đáp ứng được một số yêu cầu do giảng viên hướng dẫn đề ra, nhưng vẫn còn nhiều thiếu sót, cần hoàn thiện hơn.

1. Định hướng phát triển:

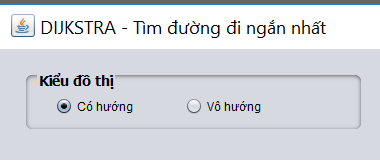
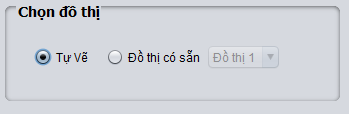
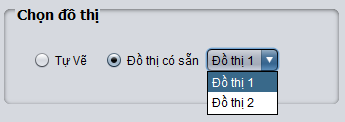
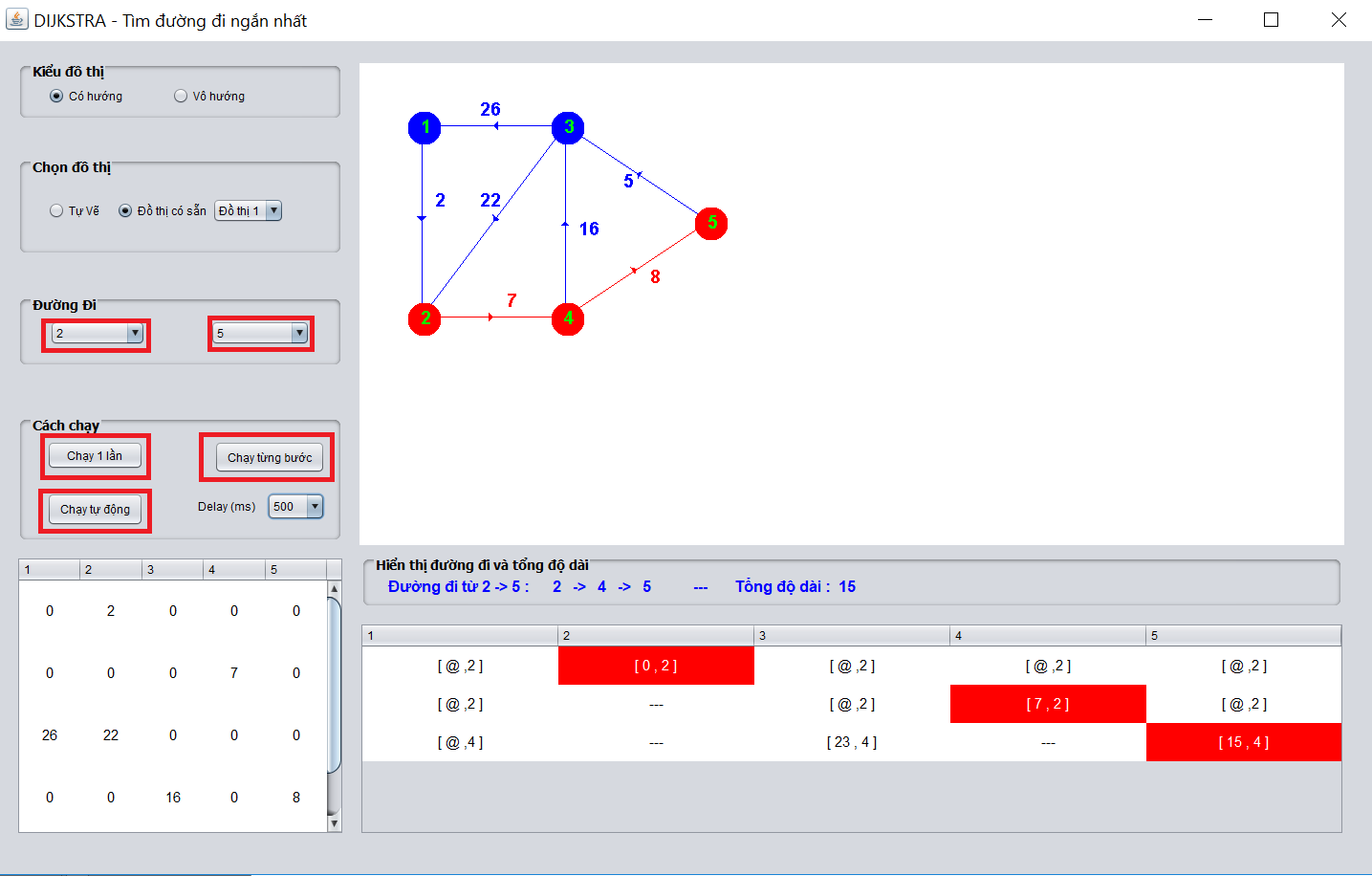
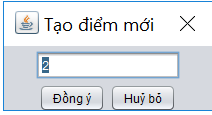
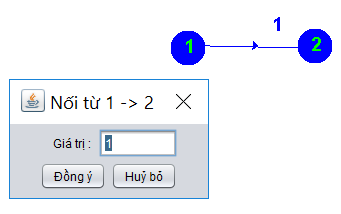
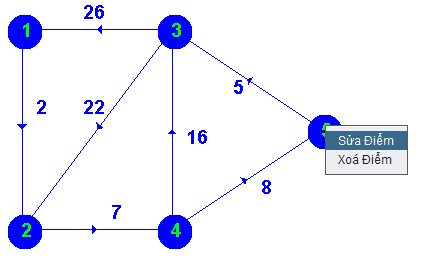
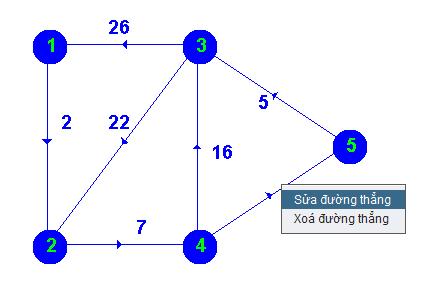
* Bỏ sung thêm các tính năng mới, khắc phục những khuyết điểm đã nêu ở mục 5.2.
* Tối ưu hóa thiết kế, giao diện cho đẹp mắt hơn.

# Phụ lục

## Hướng dẫn cách cài đặt và khởi chạy phần mềm (Trên windows)

* Tải dữ liệu trên đường dẫn website: <https://github.com/theanh97/Do_An_1>
* Download file Zip tại Clone or download: 
* Cài đặt NetBean IDE 8.2 (Free) tại: [https://netbeans.org/downloads/](https://netbeans.org/downloads/)
* Các bước cài đặt:
  + Bước 1:  
    
  + Bước 2:  
    
  + Bước 3: Chọn thư mục cài đặt NetBeans và JDK.  
    
  + Bước 4:  
    
  + Bước 5:  
    
* Khởi chạy phần mềm:
  + Bước 1: Mở chương trình NetBeans vừa cài đặt  
    
  + Bước 2: Giải nén thư mục Zip tải về trên Github, ta được:  
    
  + Bước 3: Mở chương trình NetBeans vừa cài đặt:  
    
  + Bước 4: Chọn **File -> Open project -> Chọn thư mục Zip vừa giải nén -> Open project**
  + Bước 5: Khởi chạy chương trình:  
    
  + Ta được: 

## Hướng dẫn sử dụng

* Đầu tiên ta chọn kiểu đồ thị (có hướng hoặc vô hướng) muốn hiển thị:  
  
* Tiếp theo, nếu muốn tự thiết kế độ theo ý muốn ta chọn tự vẽ  
  
* Hoặc có thể chọn 1 trong 2 đồ thị có sẵn trong chương trình  
  
* Muốn tìm các đường đi từ 1 điểm tới 1 điểm ta chọn điểm đầu và điểm điểm đích ở **Đường Đi** xong chọn tính năng chạy 1 lần (ra luôn kết quả) hoặc chạy từng bước (hiển thị từng bước chạy) hoặc chạy tự động (tự động chạy đến khi ra kết quả) với thời gian chờ (delay) tự chọn.  
  
* Các chức năng thao tắc với đồ thị gồm có:
  + Tạo điểm mới: Click chuột phải vào không gian trống trên đồ thị.  
    
  + Nối 2 điểm: Rê chuột phải từ điểm này đến vị trí điểm khác sau đó thả ra và nhập giá trị độ dài đường thẳng vừa nối.  
    
  + Di chuyển 1 điểm: Rê chuột trái tại điểm đó đến vị trí khác.
  + Sửa, xóa điểm: Click chuột phải vào 1 điểm sau đó thả ra rồi chọn chức năng muốn sử dụng.  
    
  + Sửa, xóa đường thẳng: Click chuột phải vào đường thẳng sau đó thả ra rồi chọn chức năng muốn sử dụng.  
    

## Tài liệu tham khảo

* <https://www.google.com/search>