TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Object Oriented Programing Practice Class Mini-projects:Traveling Saleman Problem

GVHD: Ph.D.Bùi Thị Mai Anh

Mã HP: IT3120

Nhóm 5:

1. **Bùi Hoàng Hiệp**
2. **Lê Thanh Bảo**
3. **Phạm Minh Chiến**

Mục lục

1. Lời mở đầu

1.1. Nội dung đề tài

Mô phỏng lại các thuật toán giải quyết vấn đề Traveling Saleman Problems

1.2. Mục tiêu đề tài

Hiểu khái niệm phân tích và giải quyết bài toán lập trình hướng đối tượng. Xác định các đối tượng, lớp, mối quan hệ giữa các lớp trong một hệ thống thông tin.

Mục tiêu đề tài mục đích nhằm nghiên cứu môi trường phát triển, ngôn ngữ java xây dựng những ứng dụng cụ thể.

1.3. Công nghệ sử dụng

Đồ án được thiết kế theo phương pháp lập trình hướng đối tượng bằng ngôn ngữ Java. Chính vì vậy mà nó giải quyết được những vướng mắc gặp phải khi thiết kế theo phương pháp lập trình thủ tục thuần túy.

Đồ án còn sử dụng JavaFX làm thư viện GUI chuẩn cho JavaSE. Để giúp đồ án trở nên sinh động và dễ hiểu hơn.

1.4. Phạm vi nghiên cứu

Nghiên cứu môi trường phát triển, ngôn ngữ Java trong lập trình hướng đối tượng. Tìm hiểu cách xây dựng các phương thức, thuộc tính đối tượng trong Java.

1.5. Phân chia công việc

-Bùi Hoàng Hiệp: GUI module, GraphView module, Algorithm module

-Lê Thanh Bảo: Graph module, GraphView module

-Phạm Minh Chiến: làm Algorithm module , Step module

2. Nội dung báo cáo

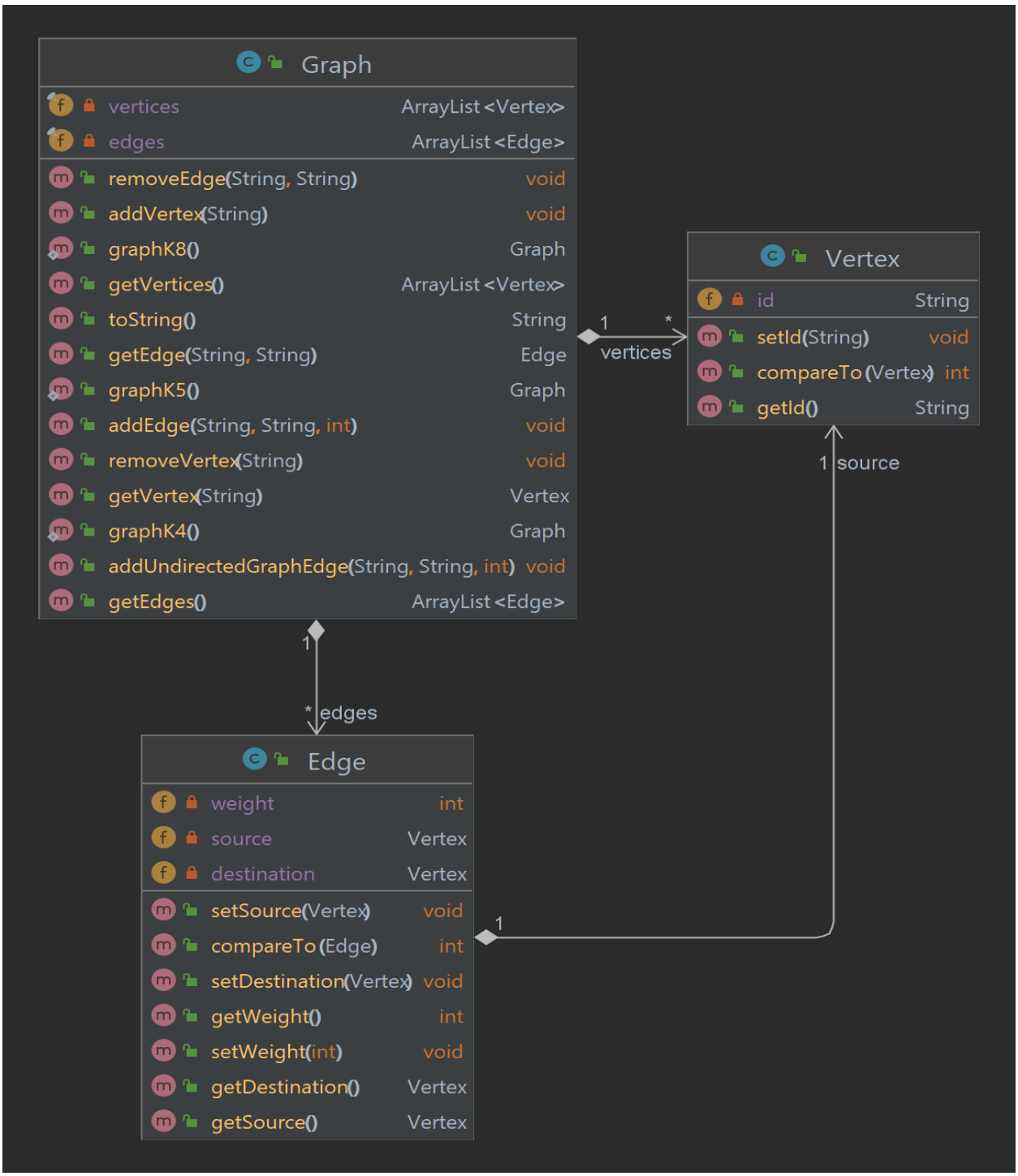
2.1. Tìm hiểu yêu cầu

Sử dụng lập trình hướng đối tưởng để giải quyết các bài toán về đồ thị

- Brute Force
- Dynamic Programing
- Approximation

2.2. Phân tích cấu trúc source code

a) Graph module

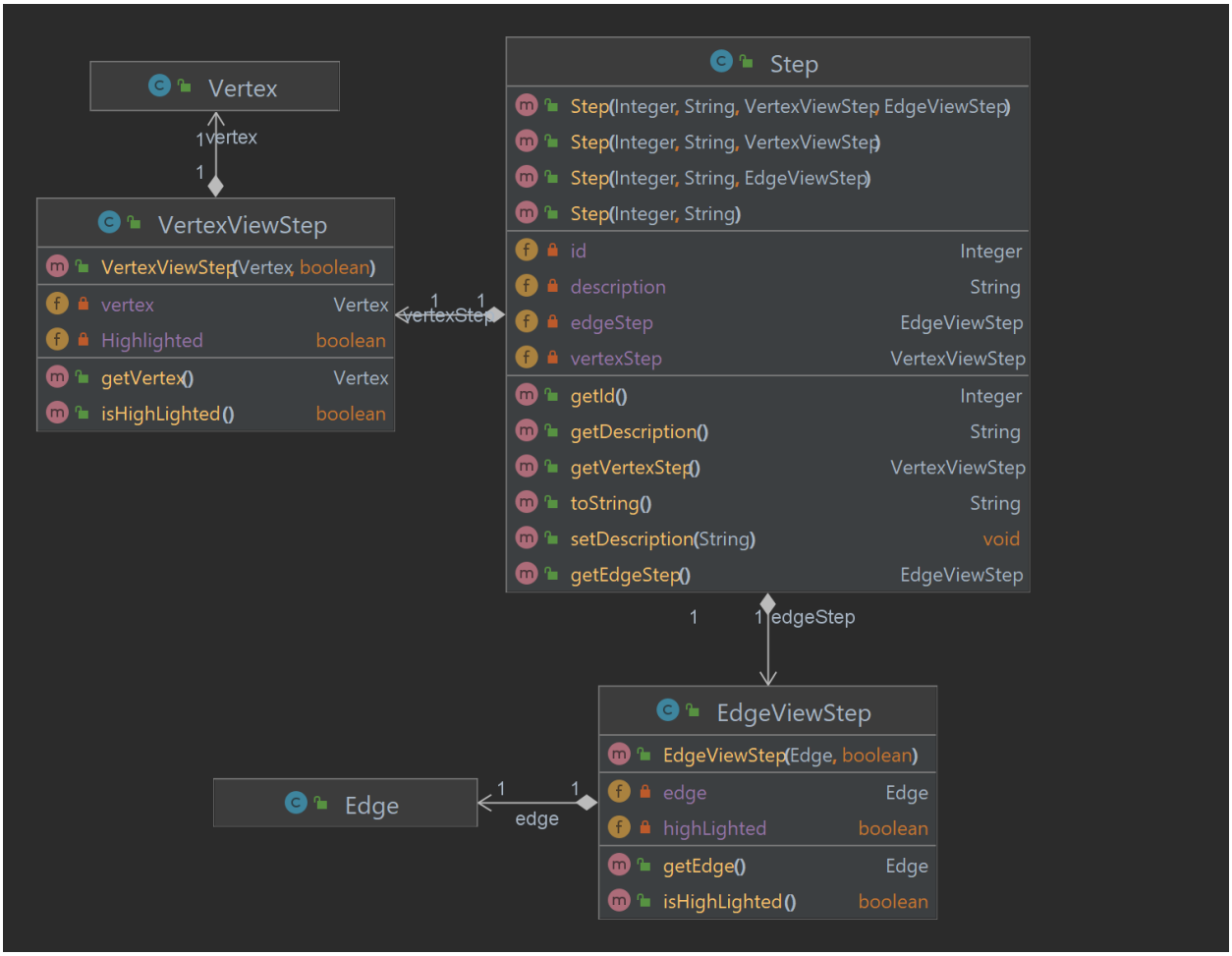


-Vertex là 1 điểm trên đồ thị nó chứa id là số thứ tự của đỉnh đó

-Edge là cạnh chứa 2 vertex(2 điểm)

-Graph là tập hợp các Vertex(ArrayList<Vertex> Vertices) và Edge(ArrayLists<Edge> Edges). Graph có chứa các method như: addVertex, addEdge, remove, get, các ví dụ Graph(K4, K5, K8)…

b) Step module



-Step module dùng để lưu lại các bước trong khi chạy thuật toán bao gồm các cạnh, các đỉnh được duyệt, thứ tự của bước trong thuật toán đang được xủ lí.

-VertexViewStep class chứa Vertex được duyệt và giá trị boolean HighLighted xem Vertex đó có được tô màu hay không.

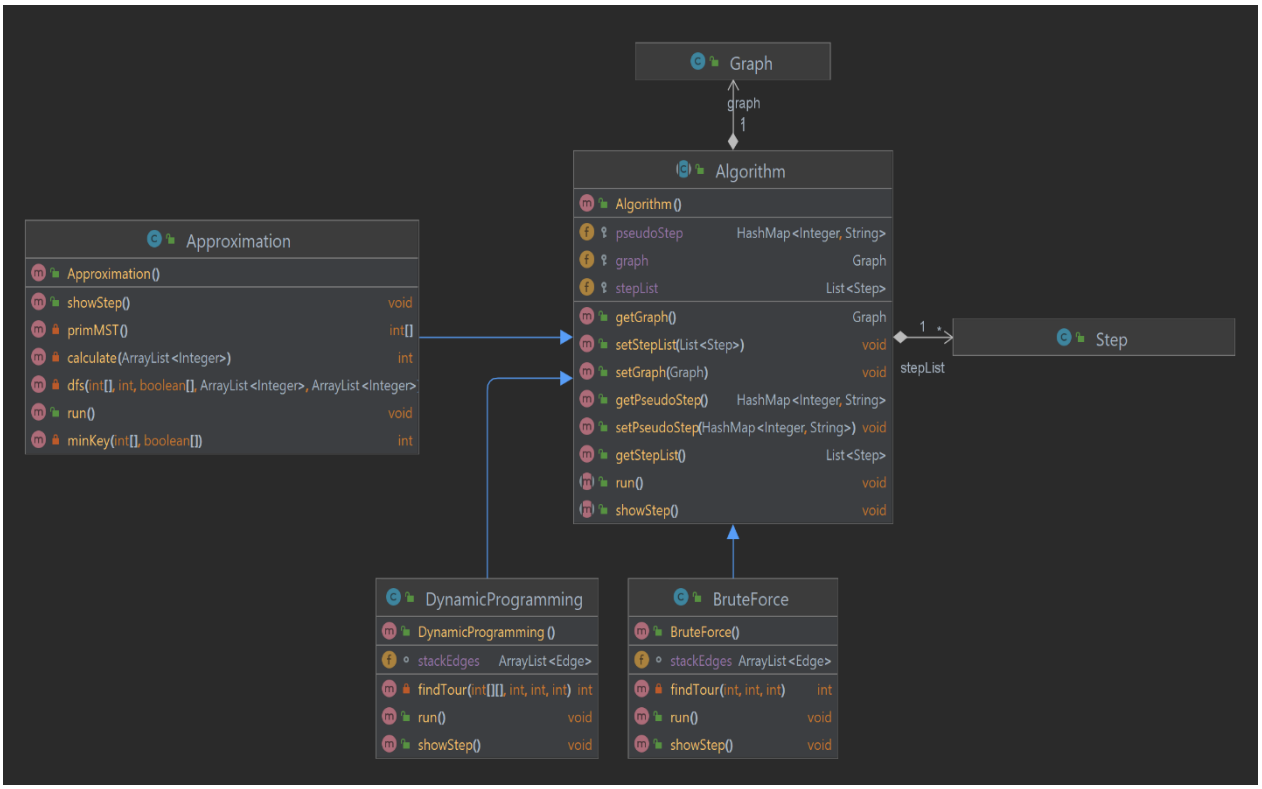
-EdgeViewStep class chứa Edge được duyệt và giá trị boolean HighLighted giống như VertexViewStep.

-Step class bao gồm VertexViewStep, EdgeViewStep, discription để chưa nội dụng các bước trong thuật toán và Id là stt của Step. Class này có nhiều Contructor để cho sử dụng cho nhiều trường hợp khác nhau như:

+ Nếu chỉ cần ghi nd, tô màu đỉnh: Step(Id, description, VertexViewStep)

+ Nếu chỉ cần ghi nd, tô màu cạnh: Step(Id, description, EdgeViewStep)

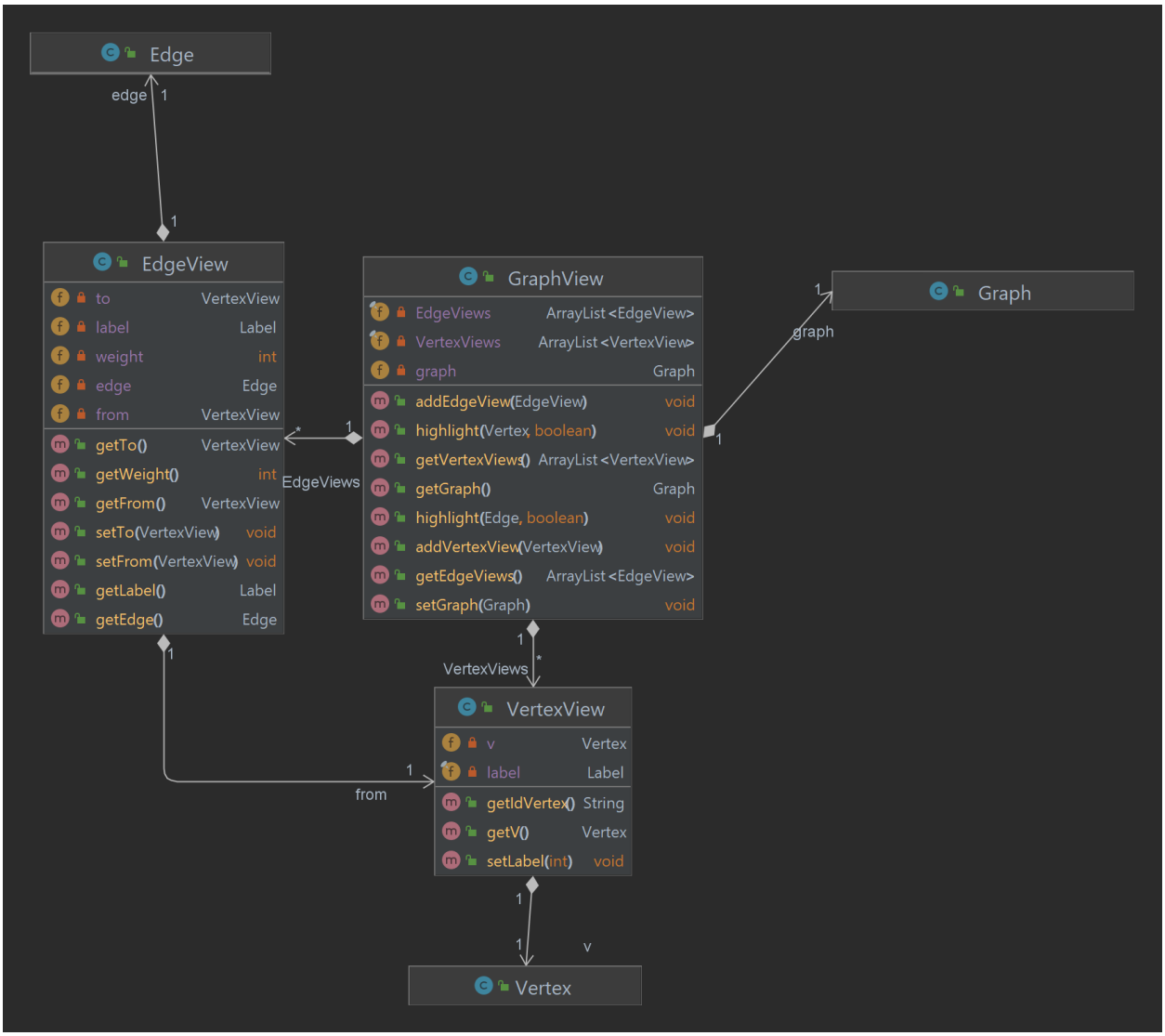
c) Algorithm module



Vì cả 3 thuật toán chính đều có những attribute(graph, stepList, pseudoStep), và method(run()) chung nên sẽ tạo thêm abstract class Algorithrm .Algorithm class sẽ sử thêm Graph module và Step module.

Cả 3 thuật toán chính còn lại sẽ kế thừa Algorithm class.

d) GraphView Module



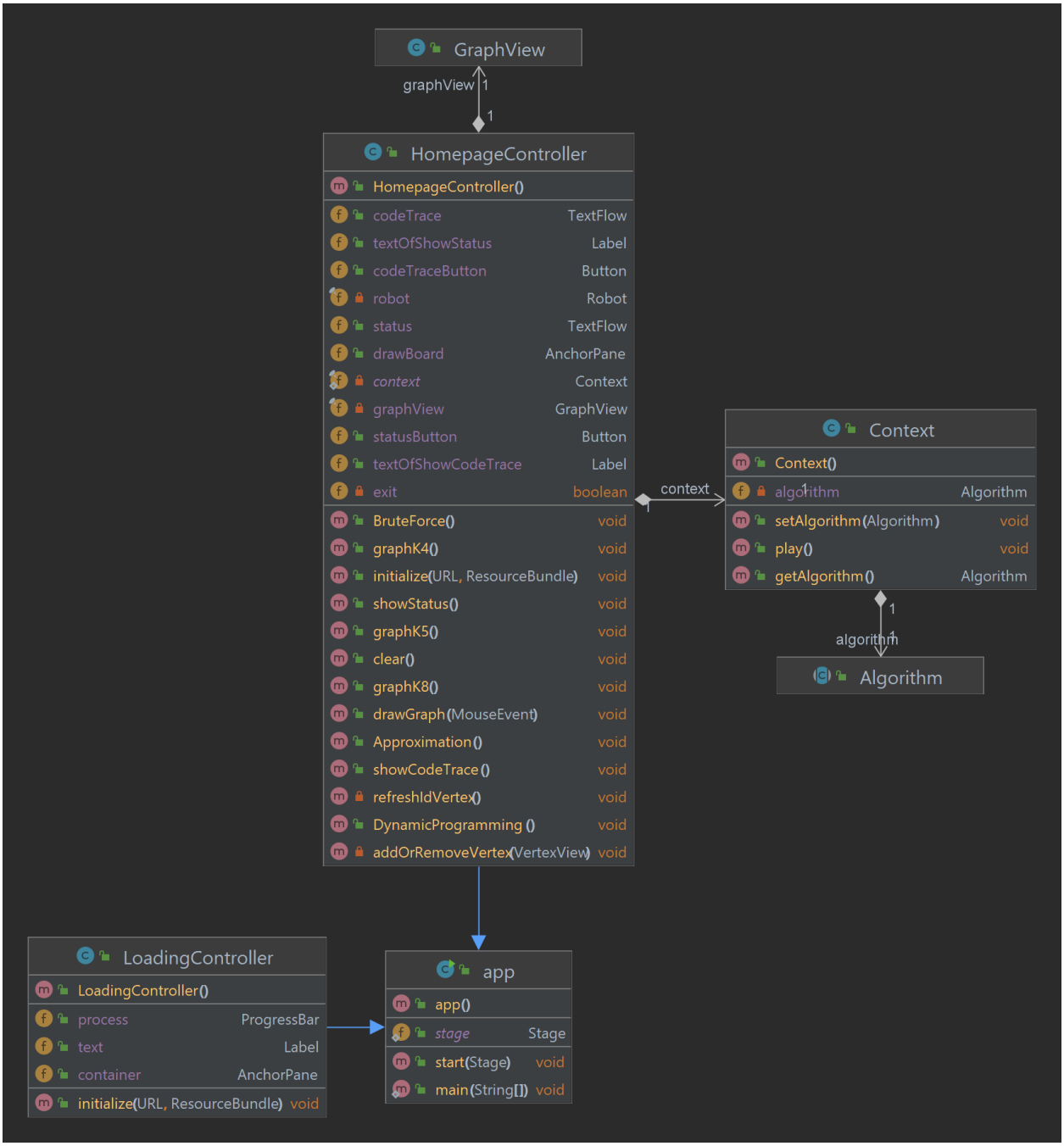
Module này sẽ sử dụng thêm Graph module để hiển thị và tạo đồ thị thông qua màn hình.

-VertexView class kế thừa StackPane(JavaFX) dùng đẻ hiện thị Vertex lên màn hình bằng cách tạo 1 Circle và 1 Label rồi cho vào StackPane. Đồng thời cũng tạo Vertex với Id giống như Label.

-EdgeView class kế thừa Line(JavaFX) đùng để hiện thị Edge bằng cách tạo Line giữa 2 VertexView. Đồng thời cũng tạo Edge dựa trên Id của 2 VertexView và weight là khoảng cách giữa VertexView trên màn hình.

-GraphView class chứa tập hợp các VertexView và EdgeView. Class có các method như: addVertexView, addEdgeView… Mỗi khi thêm, bớt 1 VertexView hay EdgeView thì cũng thêm, bớt Vertex hay Edge tương ứng vào Graph.

e) GUI module

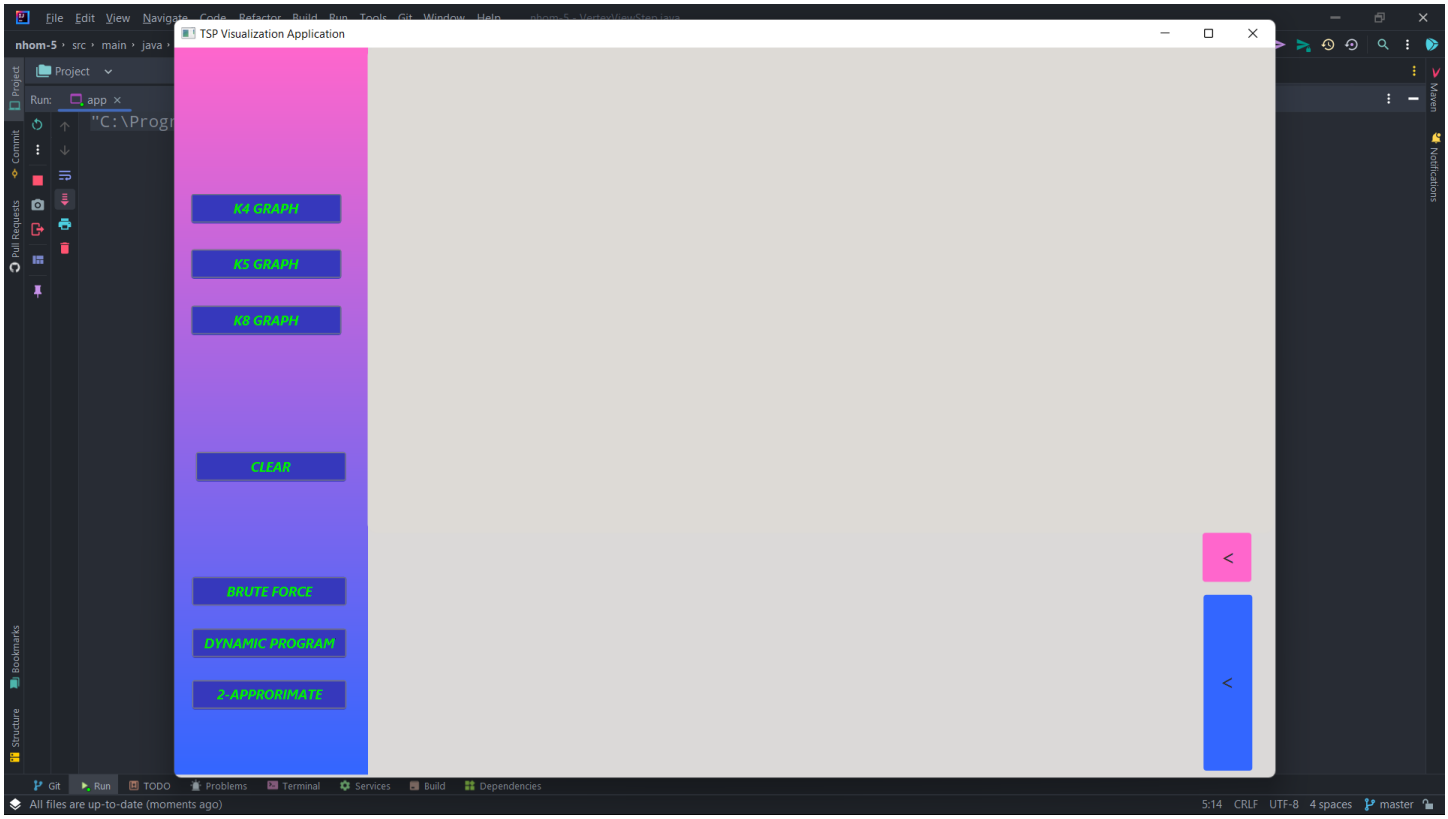


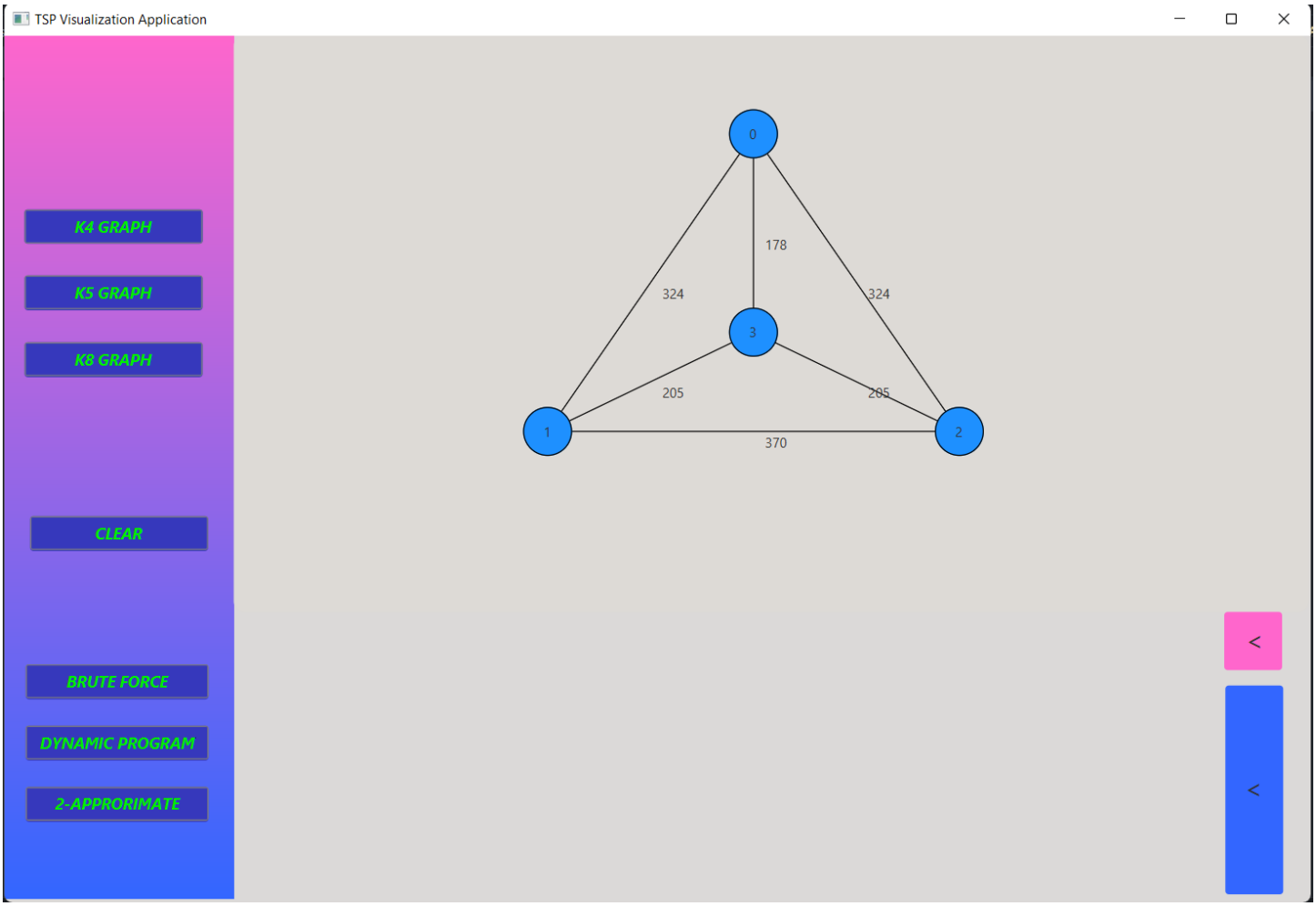
Main class là App, khi chạy thì App sẽ tải file loading.fmxl(giao diện loading cho ứng dụng) rồi tự động chuyển sang file Homepage,fxml chứa giao diện chính của app.

-LoadingController class dùng để điều khiển hành vi của file loading.fxml

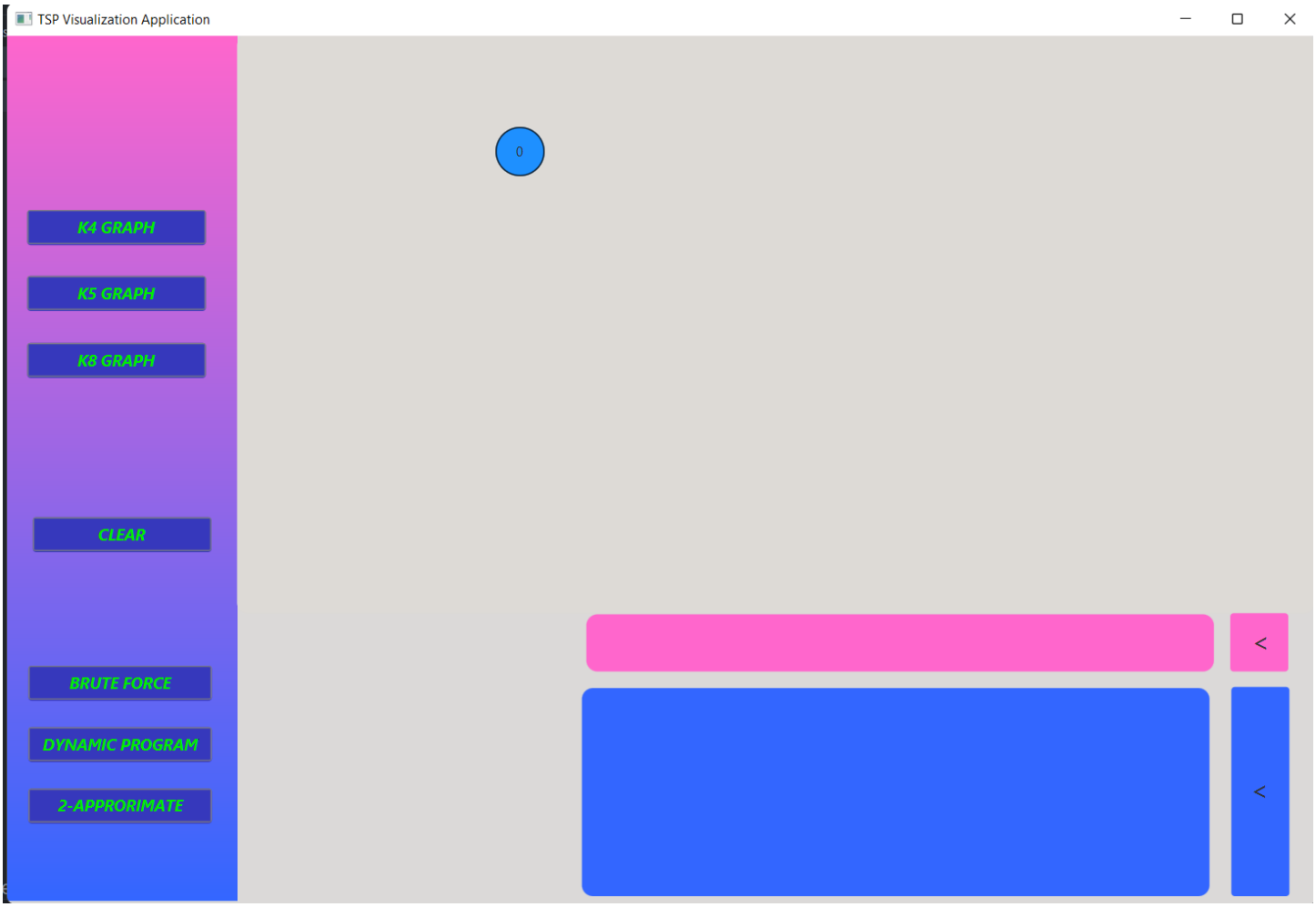
-HomepageController class dùng để điều khiển hành vi của file Homepage.fxml . Class sử dụng thêm Graphview để tạo và hiện thị Graph, sau đó thông qua class Context truy cập được Algorithm và bắt đầu chạy thuật toán. Các bước chạy sẽ được hiện thị dựa trên Step của Algorithm.

2.3 Giao diện và Demo ứng dụng

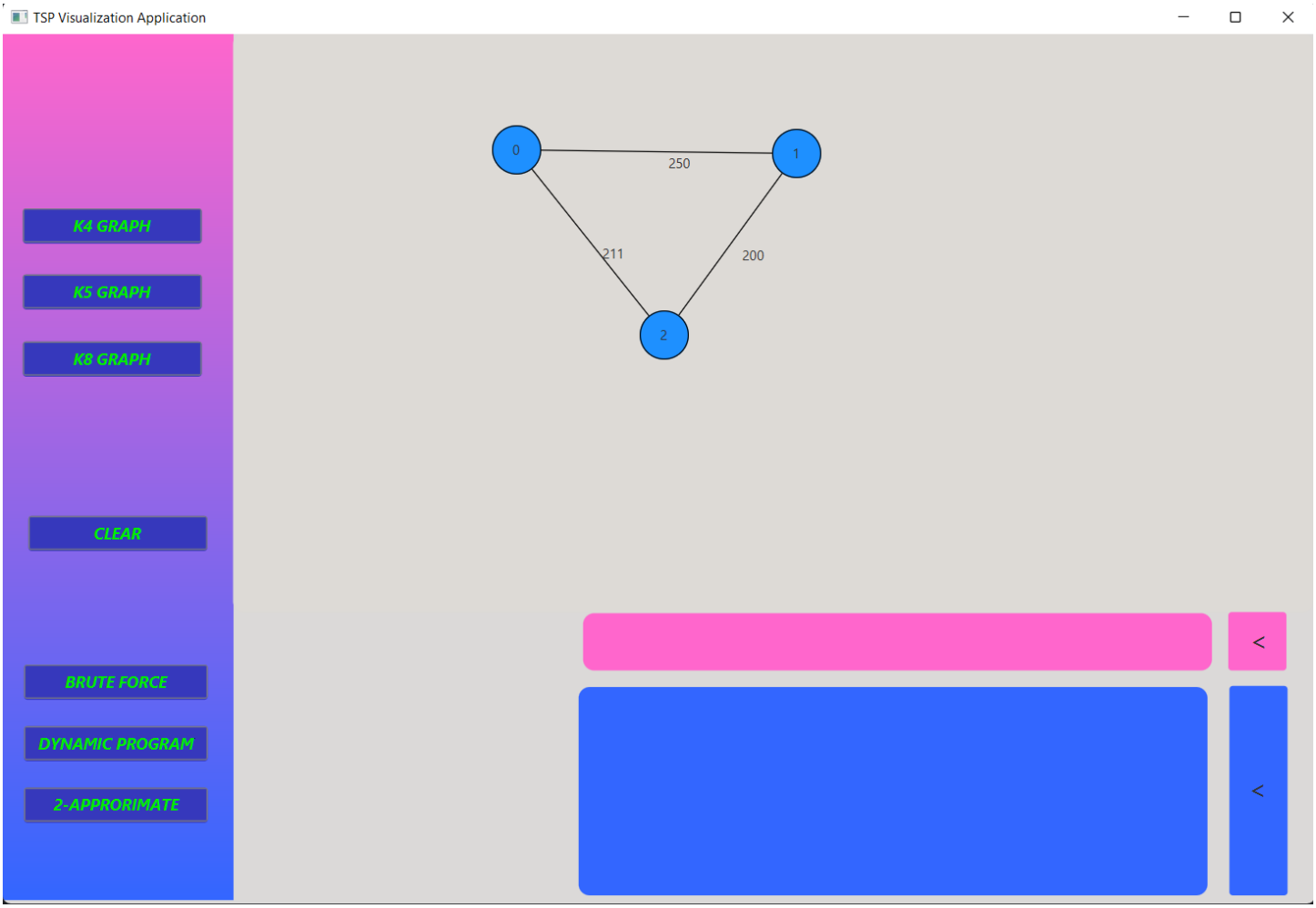




Các nút như K4, K5, K8GRAPH khi ấn vào sẽ tạo ra đồ thị tự động gồm lần lượt 4, 5, 8 đỉnh

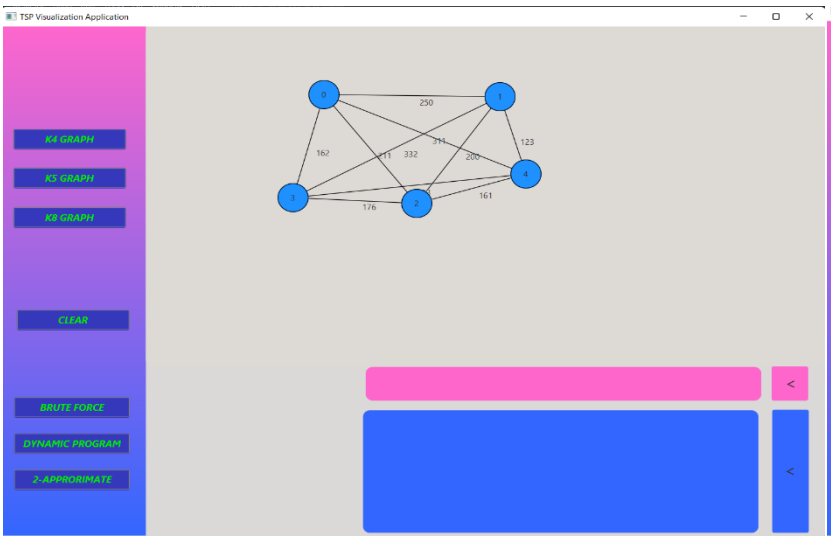


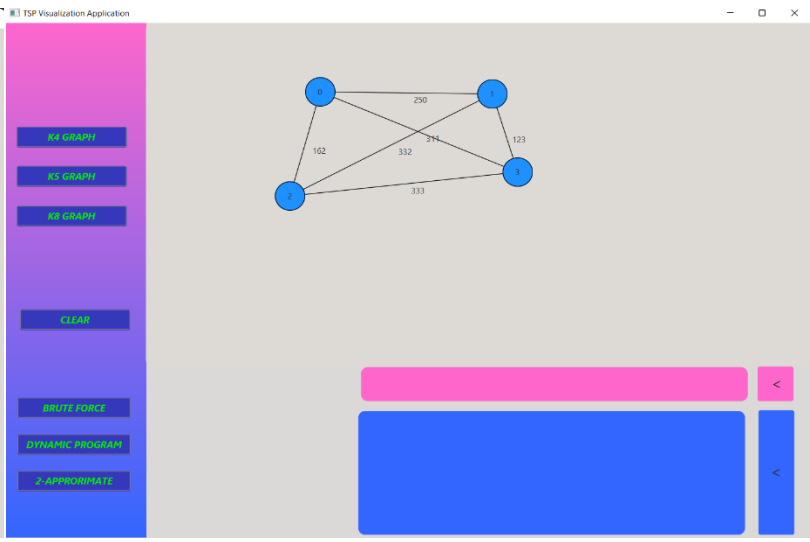
Khi ấn vào bảng trắng thì sẽ động tạo điểm theo vị trí ấn chuột.



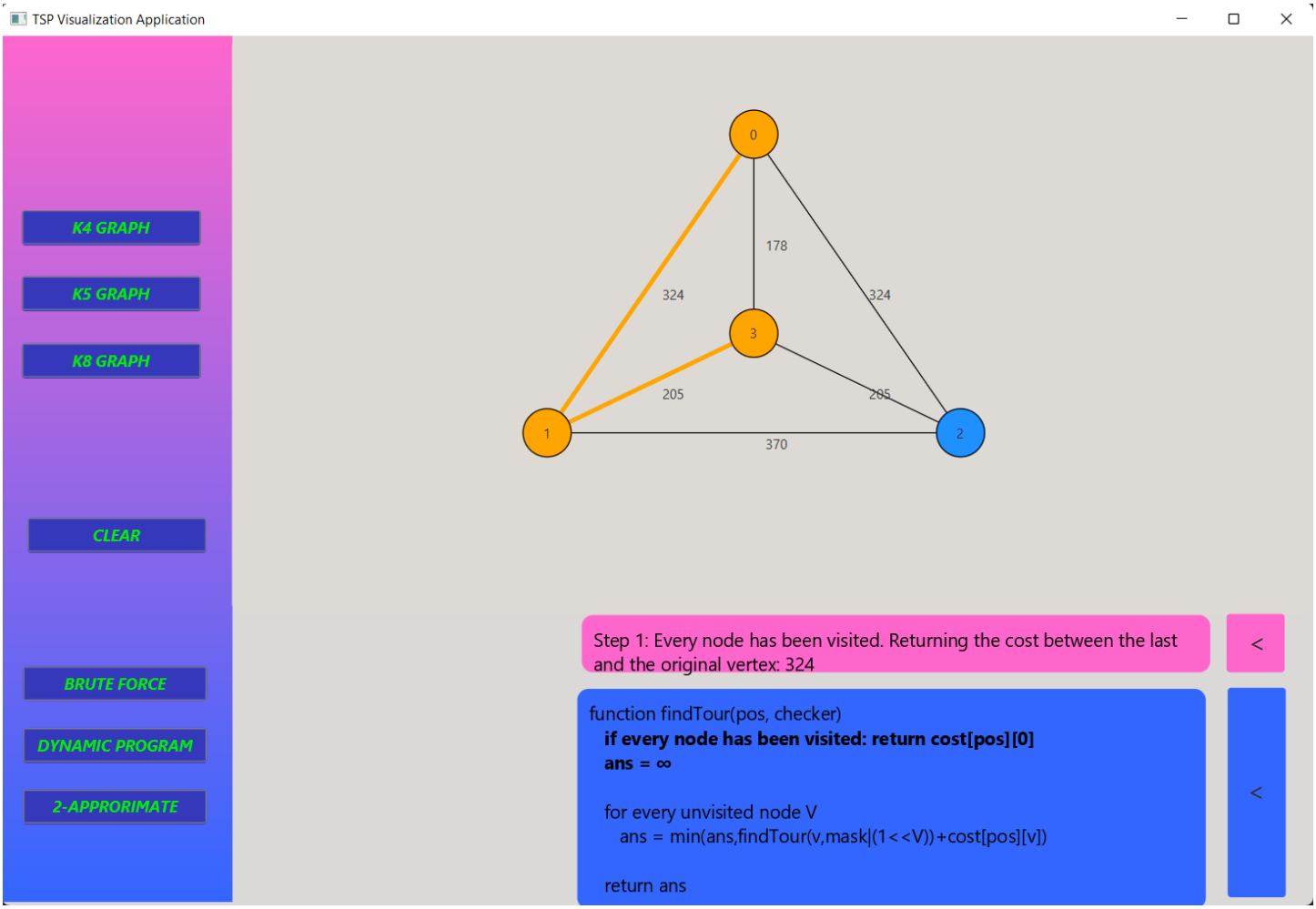
Khi có từ 2 đỉnh trơ lên thì sẽ tự động tạo thêm cạnh giữa các đỉnh với nhau.

Đưa chuột vào đỉnh ấn nut Delete trên bàn phím thì sẽ xóa đỉnh đó, stt các đỉnh sẽ được sửa lại





Nút CLEAR sẽ xóa sạch mọi thứ trên hình.



Các nút BRUTEFORCE, DYNAMIC PROGRAMING, 2-APPRORIMATE dùng để chạy các thuật toán như tên gọi, khi chạy sẽ hiện thi các step và tô màu đồ thị.

3.Kết luận

3.1 Những điều đạt được

-Hiểu và làm việc được với môi trường phát triển, lập trình hướng đối tượng với ngôn ngữ Java.

-Cơ hội tìm hiểu và phân tích yêu cầu của bài toan thực tế.

-Xây dựng chương trình cụ thể.

-Biết cách làm việc nhóm và phân chia công việc hiệu quả.

3.2 Những điều chưa đạt được

-Chưa làm được giống như web mẫu.

-Chưa có chức năng dừng và lùi các bước.