



# PHÁT TRIỂN PHẦN MỀM HƯỚNG ĐỐI TƯỢNG

GV Hướng Dẫn: ThS. Nguyễn Ngọc Long  
SV Thực Hiện: Tạ Hoàng Thảo Vy - 1511353

## MỤC LỤC

1.	Giới thiệu .....	2
2.	Kiến trúc phần mềm.....	3
3.	Ứng dụng .....	4
4.	Tài liệu tham khảo .....	9

# 1. Giới thiệu

## a. Nội dung đề tài

Đề 06: Thiết kế và cài đặt một ứng dụng cho phép “chạy” trực quan một giải thuật viết bằng ngôn ngữ C hoặc Java. Giới hạn trong các giải thuật tìm kiếm, sắp xếp. Cài đặt ít nhất hai giải thuật sắp xếp, ít nhất 2 giải thuật tìm kiếm.

## b. Phương án thực hiện:

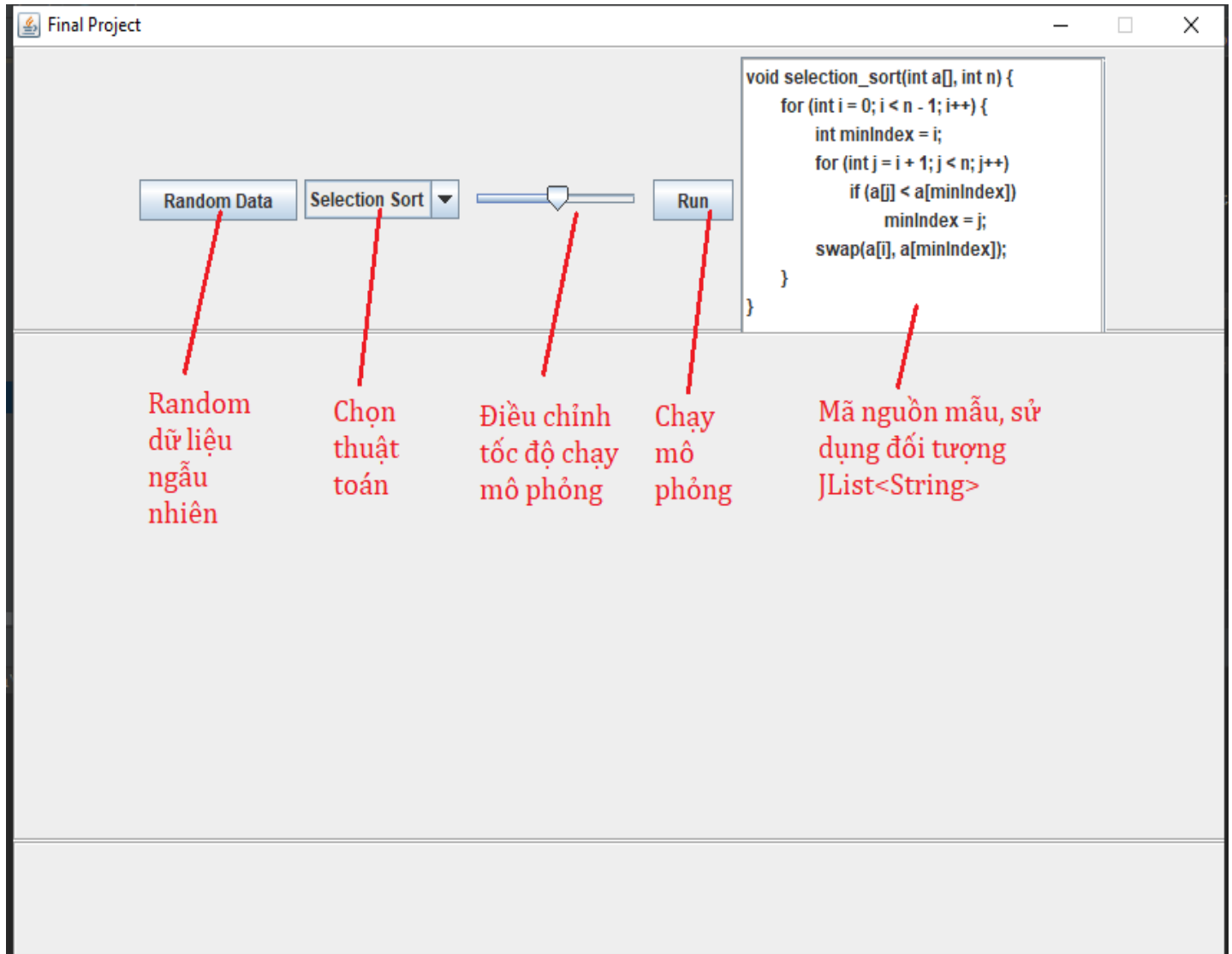
- ❖ Framework: Java Swing
- ❖ Ngôn ngữ: Java
- ❖ Demo tham khảo: <https://www.youtube.com/watch?v=Sp3gmL3cwqM>
- ❖ Thiết kế:
  - Mỗi phần tử trong mảng ứng với một đối tượng JLabel
  - Class ArrayGUI lưu trữ thông tin của mọi phần tử
  - Class SortAlgorithm thực hiện giải thuật sắp xếp, chứa đối tượng ArrayGUI. Khi cài đặt giải thuật sắp xếp sẽ thực hiện kế thừa từ lớp này. Giải thuật sắp xếp cài đặt: Selection Sort, Quick Sort
  - Class SearchAlgorithm thực hiện giải thuật tìm kiếm, chứa đối tượng ArrayGUI. Khi cài đặt giải thuật tìm kiếm sẽ thực hiện kế thừa từ lớp này. Giải thuật tìm kiếm cài đặt: Linear Search, Binary Search.
- ❖ Các tính năng kỹ thuật:
  - Class ArrayGUI chứa các phương thức cho phép hiển thị trực quan khi chọn một phần tử bất kỳ hay trao đổi hai phần tử nào đó (sử dụng multi-thread)
  - Cho phép người dùng chọn số lượng phần tử (tối thiểu: 5, tối đa: 15), đồng thời các giá trị của phần tử sẽ được random ngẫu nhiên hoặc do người dùng nhập/thay đổi.
  - Cho phép người dùng thay đổi tốc độ của quá trình “chạy” thuật toán
  - Cho phép người dùng chọn các giải thuật tương ứng, nhấn Bắt đầu để thực hiện “chạy” thuật toán.
  - Đối với mỗi thuật toán sẽ có một đoạn mã nguồn C/C++, trong khi thuật toán đang chạy, bên cạnh việc hiển thị trực quan về các phần tử, sẽ hiển thị thuật toán hiện đang chạy đến bước nào của đoạn mã nguồn đó.

## 2. Kiến trúc phần mềm

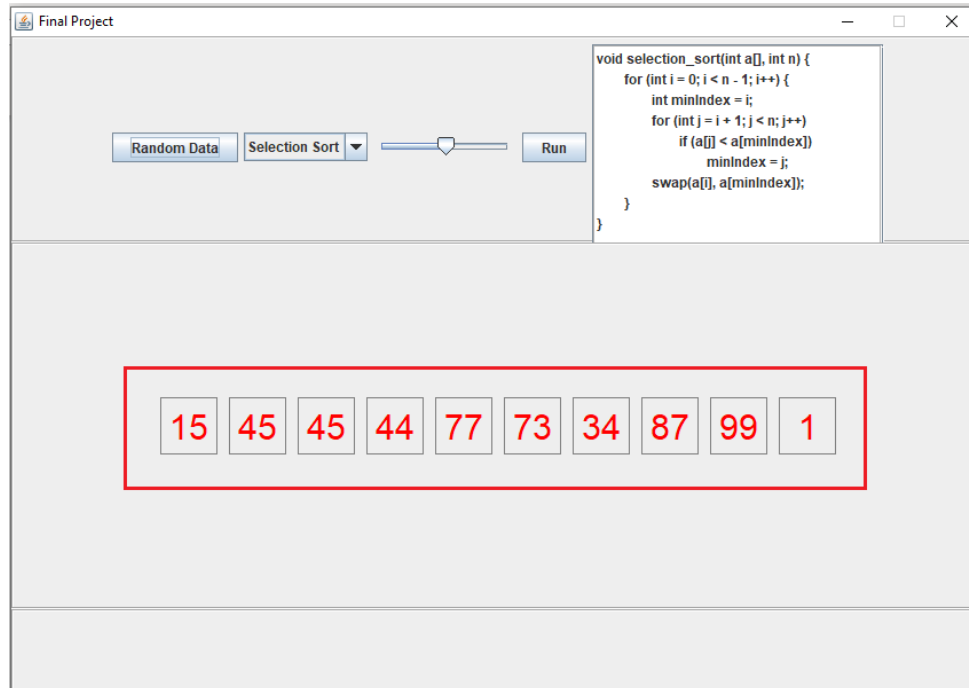
- Project sử dụng framework Java Swing.
- Class MainForm kế thừa từ JFrame, là form chính của chương trình.
- Class ArrayGUI chứa thông tin dữ liệu của một mảng dữ liệu một chiều (bao gồm thông tin về UI – sử dụng JLabel của Java Swing). ArrayGUI cung cấp các phương thức thao tác trên dữ liệu cũng như về UI, một số hàm như swapElement để trao đổi hai phần tử, highlightElement và unHighlightElement để đánh dấu cho một phần tử ở vị trí bất kỳ, hay randomData để tạo dữ liệu mảng random.
- Class Algorithm dùng để chạy mô phỏng thuật toán, thực hiện đọc mã nguồn mẫu từ file .txt thông thường. Các class SearchAlgorithm và SortAlgorithm kế thừa từ lớp này. Tiếp đến các thuật toán sắp xếp (cụ thể là SelectionSort và QuickSort) và các thuật toán tìm kiếm (cụ thể là LinearSearch và BinarySearch) sẽ kế thừa từ các lớp này.
- Class Algorithm có hai phương thức quan trọng là initialize() để cấu hình một vài thông số cần thiết trước khi chạy mô phỏng và execute() để thực hiện chạy mô phỏng. Lớp Algorithm cung cấp một số phương thức (như di chuyển JLabel, ...) để dễ dàng hơn trong việc hiển thị trực quan thuật toán.
- Ngoài ra, class Algorithm còn quản lý một đối tượng JList<String> để hiển thị mã nguồn thuật toán, cung cấp phương thức chooseLine để chọn một dòng bất kỳ (để hiển thị xem trình mô phỏng đã chạy được tới bước nào của mã nguồn).
- Mã nguồn mẫu được đặt trong thư mục Algorithm.
- Khi tiến hành chạy mô phỏng sẽ chạy trên thread riêng so với thread chính của chương trình.
- Đối với thuật toán BinarySearch, trước khi tiến hành chạy mô phỏng sẽ có thao tác sắp xếp lại dữ liệu.
- Lớp Configuration sử dụng mẫu thiết kế Singleton, ở đây có thể thay đổi các giá trị tốc độ chạy mô phỏng hay kích thước của các phần tử JLabel, ...

### 3. Ứng dụng

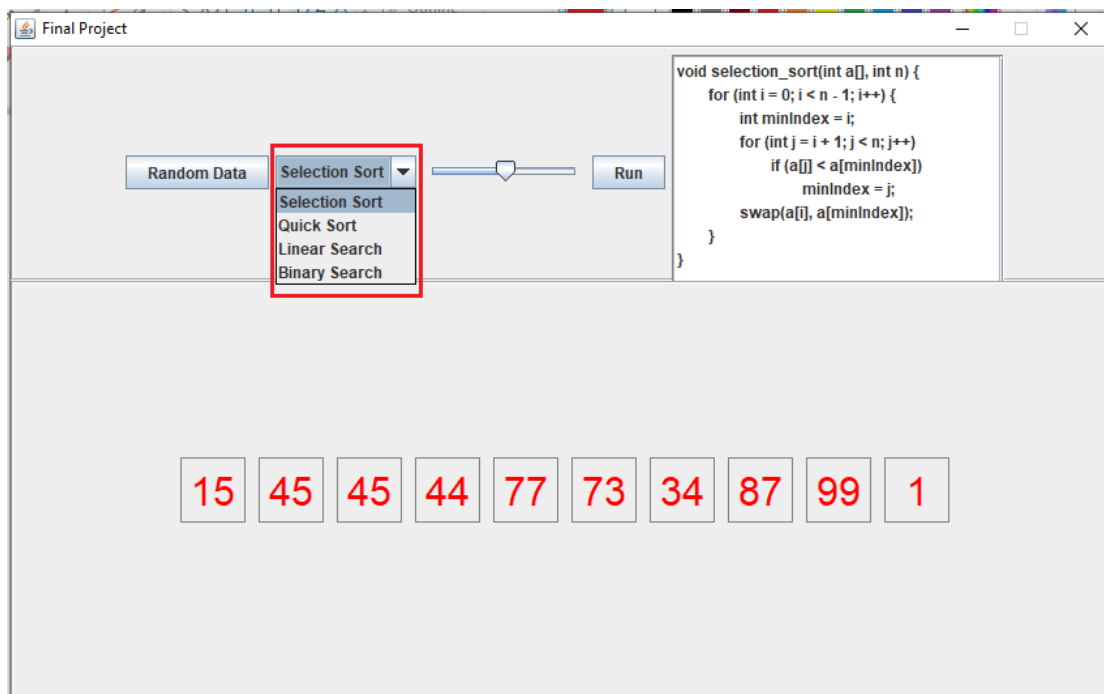
- Màn hình chính của ứng dụng



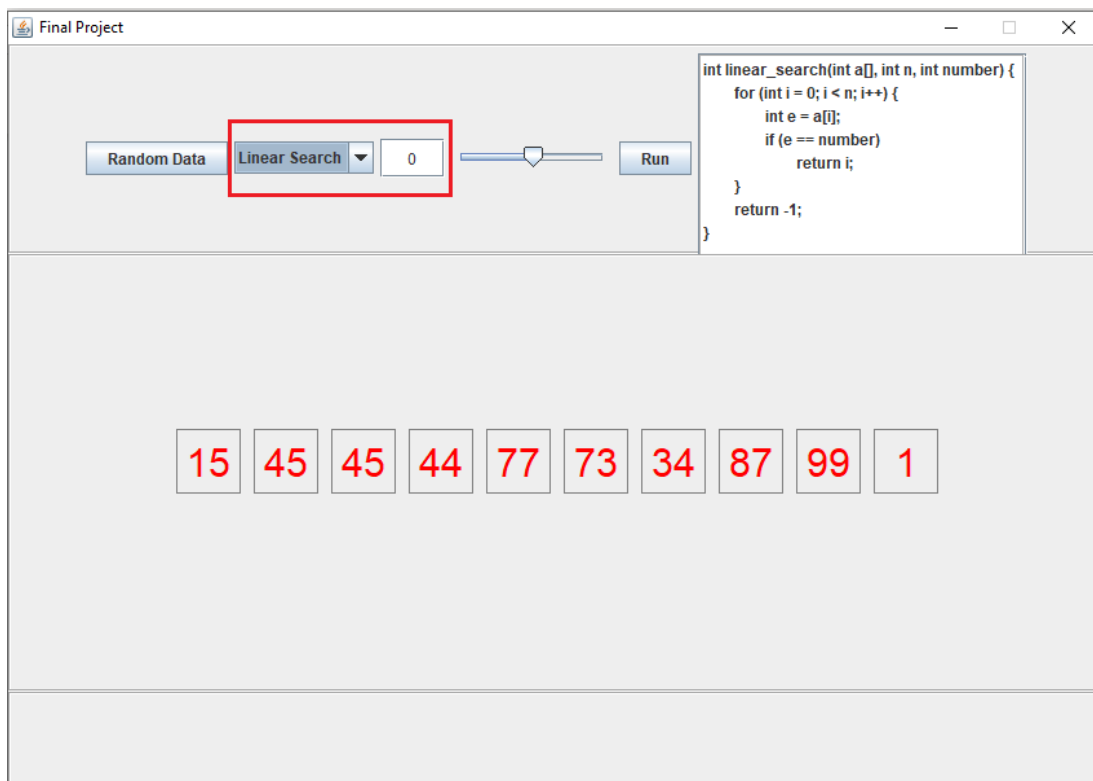
- Random dữ liệu ngẫu nhiên (khi nhấn JButton “Random Data”)



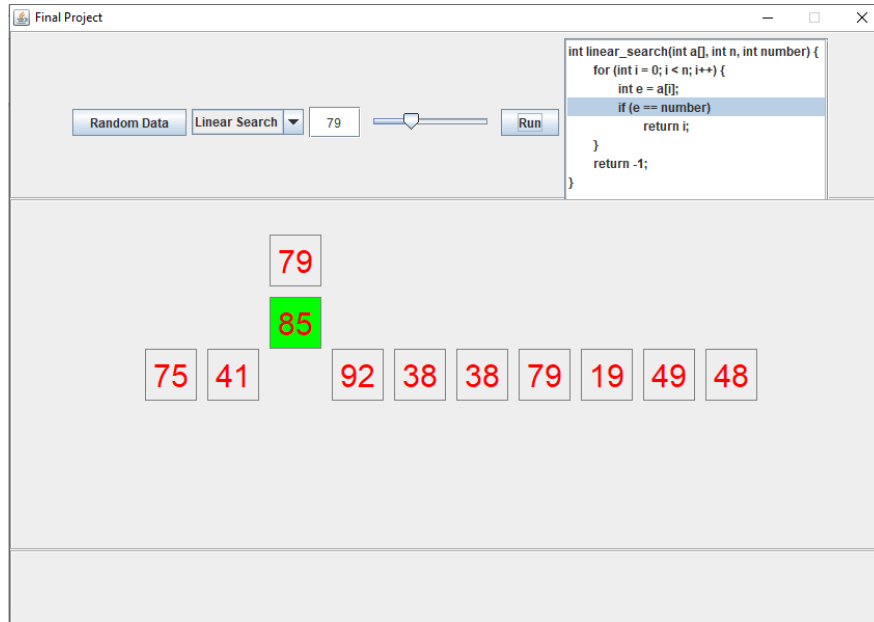
- Chọn thuật toán mô phỏng



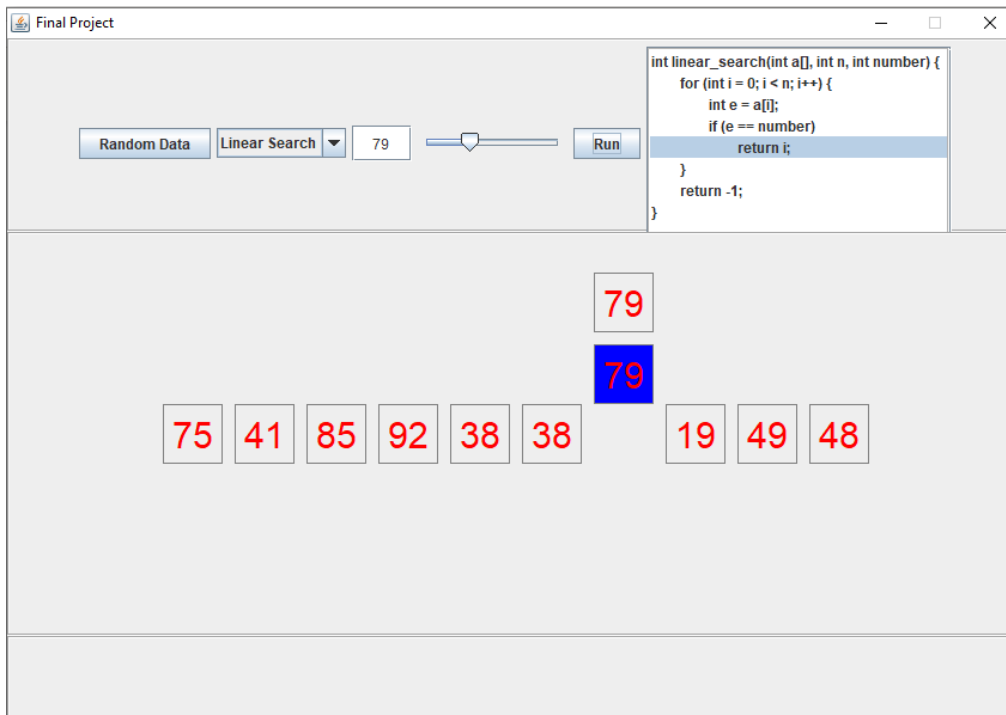
- Khi chọn các thuật toán tìm kiếm, sẽ hiển thị thêm một ô Text Box để nhập giá trị của phần tử cần tìm kiếm



- Giao diện chạy thuật toán LinearSearch
  - Đang trong quá trình chạy thuật toán, vùng mã nguồn đang chọn tới dòng so sánh



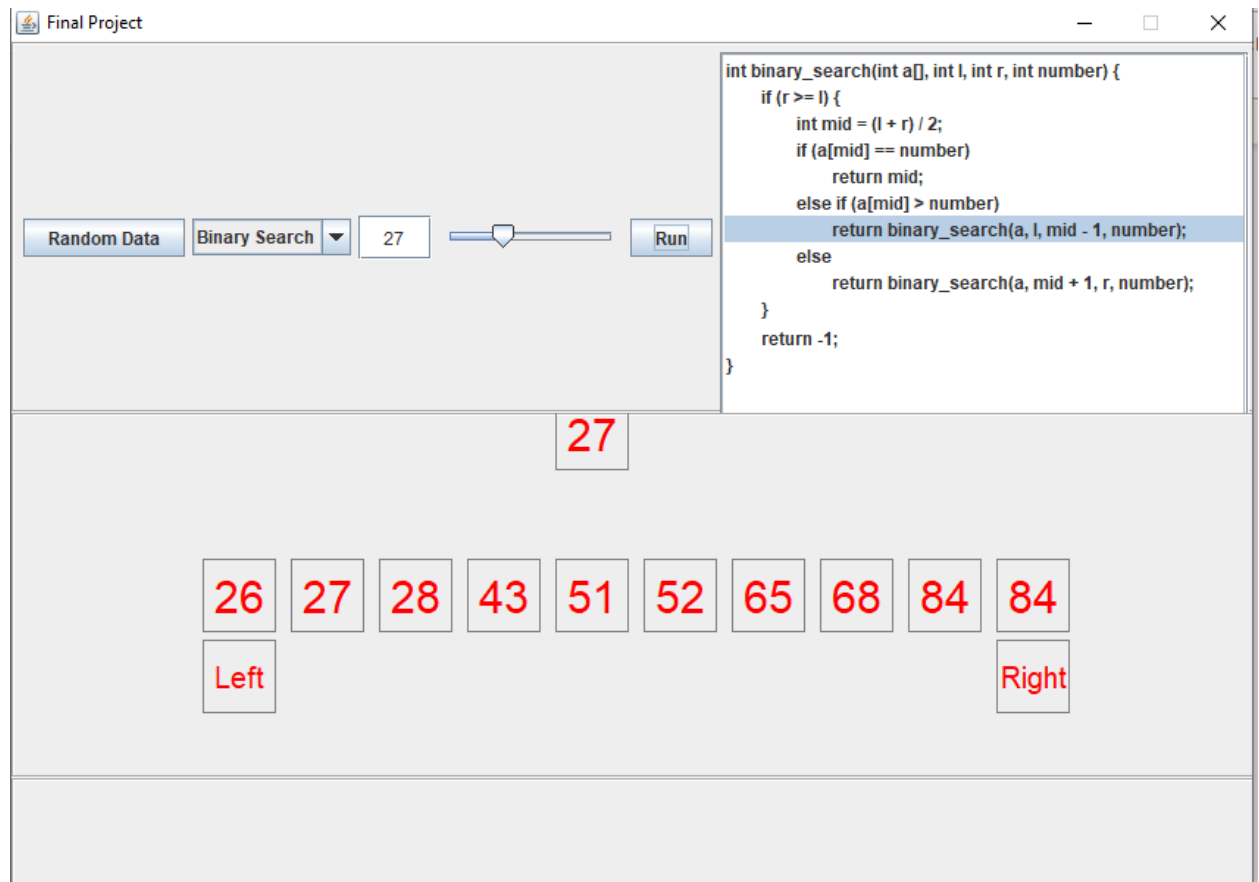
- Đã tìm thấy phần tử, vùng mã nguồn đang chọn tới dòng “return i;”





- Giao diện chạy thuật toán BinarySearch

Bên cạnh một đối tượng JLabel hiển thị số cần tìm kiếm, còn có các đối tượng JLabel khác để hiển thị các giá trị Left, Right hiện tại



## 4. Tài liệu tham khảo

- [1]. N.N.Long, Slide bài giảng Phát triển phần mềm hướng đối tượng.
- [2]. Demo mô phỏng các thuật toán sắp xếp bằng C#, Youtube.
- [3]. Linear Search, <https://www.geeksforgeeks.org/linear-search/>
- [4]. Binary Search, <https://www.geeksforgeeks.org/binary-search/>
- [5]. Java Swing, <https://www.javatpoint.com/java-swing>

HẾT.