Đại Học Bách Khoa Hà Nội

Trường Công nghệ thông tin và Truyền thông

A red and white flag with a yellow and red design

Description automatically generated

BÁO CÁO DỰ ÁN

Lập trình hướng đối tượng

Chủ đề 3:Demonstration of sorting algorithms on array (1)

Giáo viên hướng dẫn: Nguyễn Thị Thu Trang

Trợ giảng: Phạm Phan Anh

Thành viên nhóm 23

Họ và tên sinh viên : Lê Thái Sơn\_20225919

Họ và tên sinh viên: Chu Văn Thành\_20194673

Họ và tên sinh viên: Lê Đức Thắng\_20225925

Họ và tên sinh viên: Nguyễn Trung Sơn\_20226124

Họ và tên sinh viên: Nguyễn Thanh Tân\_20225923

**I.Phân công nhiệm vụ**

1)**Nhóm trưởng: Lê Thái Sơn\_20225919**

* Xây dựng khung của dự án, cụ thể bao gồm các lớp Sort,StateSorting,StateSwap và cách 3 controller tô màu
* Thiết kế giao diện dự án, cụ thể bao gồm các lớp Helper,HomeController và 2 file homepage.fxml và get\_help\_page.fxml
* Thực hiện đóng gói và các sourcecode thành 3 package model,view,controller
* Tham gia làm slide và viết báo cáo( chỉ viết báo cáo và làm slide phần mình đã làm nên đóng góp 20%)
* Tham gia đóng góp và tổng hợp ý kiến xây dựng use case diagram và class diagram

2)**Thành viên Nguyễn Trung Sơn \_20226124**

* Giao diện của sort,điều khiển, đầu vào mảng, sort trực tiếp.
* Tham gia đóng góp ý kiến xây dựng use case diagram và class diagram

3) **Thành viên Chu Văn Thành\_20194673**

* Chỉnh sửa UI giao diện, thêm màu cho phần UI của các thuật toán sort để người dùng dễ hình dung các bước hơn.
* Tham gia đóng góp ý kiến xây dựng use case diagram và class diagram.

4) **Thành viên Nguyễn Thanh Tân\_20225923**

* Triển khai thuật toán MergeSort
* Làm khung slide
* Tham gia đóng góp ý kiến xây dựng use case diagram và class diagram

5)**Thành viên Lê Đức Thắng\_20225925**

* Triển khai thuật toán ShellSort (80%) (20% còn lại bạn Nguyễn Thanh Tân hỗ trợ fix phần tô màu)
* Tổng hợp viết báo cáo report
* Tham gia đóng góp ý kiến xây dựng use case diagram và class diagram.

|  |  |
| --- | --- |
| Lê Thái Sơn | 30% |
| Chu Văn Thành | 20% |
| Nguyễn Trung Sơn | 17,5% |
| Nguyễn Thanh Tân | 17,5% |
| Lê Đức Thắng | 15% |

1. **Mô tả dự án**
2. Mô tả chi tiết về yêu cầu của dự án

Mảng là cấu trúc cơ bản nhất của Khoa học máy tính. Hầu hết các thao tác cũng như các cấu trúc dữ liệu khác đều được xây dựng và thực hiện trên mảng. Trong dự án này, nhóm 23 tạo một ứng dụng để giải thích ba thuật toán sắp xếp trên mảng là : Selection Sort,Merge Sort và Shell Sort.

1. Sơ đồ trường hợp sử dụng và giải thích

A diagram of a diagram

Description automatically generated

1. **Thiết kế**
2. Một sơ đồ lớp chung

A close-up of a yellow paper

Description automatically generated

1. Một số sơ đồ lớp cho mỗi gói hoặc một số gói với các thuộc tính/ hoạt động chi tiết cho mỗi lớp

2.1.Sơ đồ lớp tổng quát:

Các gói:

1. **model**: Chứa các lớp liên quan đến thuật toán và dữ liệu.
2. **view**: Chứa các lớp giao diện người dùng.
3. **controller**: Chứa lớp điều khiển, chịu trách nhiệm kết nối giữa **model** và **view**.

**Chi tiết các lớp:**

**1. Gói model**

* **Lớp Sort (lớp cha):**
  + **Thuộc tính:**
    - int[] array: Mảng cần sắp xếp.
    - int size: Kích thước của mảng.
  + **Phương thức:**
    - void setArray(int[] array): Gán giá trị cho mảng.
    - int[] getArray(): Trả về mảng hiện tại.
    - abstract void sort(): Phương thức trừu tượng, thực hiện thuật toán sắp xếp.
    - String getSteps(): Trả về các bước thực hiện thuật toán.
* **Lớp SelectionSort (kế thừa từ Sort):**
  + **Thuộc tính:** Kế thừa từ lớp cha.
  + **Phương thức:**
    - void sort(): Triển khai thuật toán sắp xếp chọn.
    - String getSteps(): Ghi lại từng bước sắp xếp.
* **Lớp MergeSort (kế thừa từ Sort):**
  + **Phương thức:**
    - void sort(): Triển khai thuật toán sắp xếp trộn.
    - String getSteps(): Ghi lại các bước.
* **Lớp ShellSort (kế thừa từ Sort):**
  + **Phương thức:**
    - void sort(): Triển khai thuật toán sắp xếp Shell.
    - String getSteps(): Ghi lại các bước.

**2. Gói view**

* **Lớp MainMenu:**
  + **Thuộc tính:**
    - JButton btnSelectionSort, btnMergeSort, btnShellSort: Nút chọn thuật toán.
    - JButton btnHelp, btnQuit: Nút trợ giúp và thoát.
  + **Phương thức:**
    - void display(): Hiển thị giao diện chính.
    - void showHelp(): Hiển thị hướng dẫn.
* **Lớp DemonstrationPanel:**
  + **Thuộc tính:**
    - JButton btnCreateArray, btnStart, btnBack: Các nút điều khiển.
    - JTextArea txtSteps: Hiển thị các bước thực hiện.
  + **Phương thức:**
    - void createArray(): Cho phép người dùng tạo mảng.
    - void updateSteps(String steps): Cập nhật các bước thực hiện.
    - void reset(): Đặt lại giao diện.

**3. Gói controller**

* **Lớp SortController:**
  + **Thuộc tính:**
    - Sort sortAlgorithm: Lưu trữ thuật toán đang được chọn.
    - MainMenu mainMenu: Tham chiếu đến giao diện chính.
    - DemonstrationPanel demonstrationPanel: Tham chiếu đến giao diện trình diễn.
  + **Phương thức:**
    - void setSortAlgorithm(Sort algorithm): Gán thuật toán sắp xếp.
    - void startSorting(): Kích hoạt sắp xếp và cập nhật giao diện.
    - void backToMenu(): Quay lại giao diện chính.

1. **Giải thích về thiết kế**
2. Mô tả mối quan hệ giữa các lớp

**1.Quan hệ kế thừa (Inheritance):**

Các lớp SelectionSort, MergeSort, và ShellSort kế thừa từ lớp Sort để tận dụng các thuộc tính và phương thức chung, đồng thời triển khai phương thức sort() theo thuật toán riêng.

**2.Quan hệ kết hợp (Association):**

* Lớp SortController tương tác với MainMenu và DemonstrationPanel để điều phối các thao tác từ người dùng và hiển thị giao diện tương ứng.
* Lớp SortController sử dụng lớp Sort(hoặc lớp con của nó) để thực hiện sắp xếp.

**3.Quan hệ thành phần (Composition):**

|  |
| --- |
| **StateSorting** |
| -iArg1: int |
| +StateSorting()  +StateSorting(int: int)  +getiArg1():int  +setiArg1(int:int): void |

|  |
| --- |
| **StateSwap** |
| -iArg1:int  -iArg2:int |
| +StateSwap()  +getiArg1():int  +getiArg2():int |

|  |
| --- |
| MergeSort |
| -currentSize:int  -leftStart:int  -stepMid:int  -stepEnd:int  -stepI:int  -stepJ:int  -stepK:int  -isMerging:boolean  -tempArray:int[]  +getSwapSorting():StateSwap |
| +MergeSort(int[]:int,int:int)  +sort():void  +getStateSorting():StateSorting  +displayFinalArray():void  +isSorted():boolean  +getArray():int[] |

|  |
| --- |
| ShellSort |
| -gap:int  -i:int  -j:int  -currentElement:int  -stepInitialized:boolean |
| +ShellSort(int[],int:int)  +getStateSorting():StateSorting  +getSwapSorting():StateSwap  +nextStep():void  +isStepInitialized():boolean  +setStepInitialized(boolean:int): void  +getI():int  +set(int:int): void  +getArray():int [] |

|  |
| --- |
| SelectionSort |
| -iArray:int[]  -iNbElement:int  -iSorted:int  -iTarget:int |
| +SelectionSort(int[]:int,int:int)  +sort():void  +isSorted():boolean  +getArray():int[]  +getStateSorting():StateSorting  +getSwapSorting():StateSwap  +getiSorted():int |

1. Một số phương pháp quan trọng
2. **Encapsulation ( đóng gói )**

Các thuộc tính của lớp StateSorting và StateSwap ( như iArg1,iArg2) được khai báo là private. Điều này bảo vệ dữ liệu bên trong và chỉ cho phép truy cập hoặc chỉnh sửa thông qua các phương thức công khai như getiArg1,getiArg2,setiArg.

1. **Abstraction ( trừu tượng hóa)**

Lớp Sort là abstract class, chứa các phương thức trừu tượng getStateSorting và getSwapSorting. Các lớp con phải ghi đè các phương thức này, cung cấp các cách triển khai riêng theo thuật toán của từng loại sắp xếp.

1. **Inheritance (kế thừa)**

Lớp InsertionSort,MergeSort and ShellSort đều kế thừa từ lớp cơ sở Sort, điều này giúp tái sử dụng các thuộc tính và phương thức cơ bản như iArray,iNbElement,bSortDone và các phương thức trừu tượng ( getStateSorting,getSwapSorting).

1. **Association ( kết hợp)**

Lớp SortController tương tác với MainMenu và DemonstrationPanel để điều phối các thao tác từ người dùng và hiển thị giao diện tương ứng.

Lớp SortController sử dụng lớp Sort ( hoặc lớp con của nó) để thực hiện sắp xếp.

1. **Polymorphism ( đa hình)**

Các phương thức getStateSorting() và getSwapSorting() được định nghĩa trừu tượng ( hoặc mặc định) trong lớp cha Sort và được ghi đè ( override) trong các lớp con SelectionSort, MergeSort và ShellSort.

1. **Composition ( thành phần)**

Lớp MainMenu và DemonstrationPanel chứa các thành phần giao diện như nút bấm và vùng hiển thị để quản lý trực tiếp giao diện người dùng.