BÁO CÁO THỰC HÀNH

**Môn học: Lập trình ứng dụng mạng**

**Buổi báo cáo: Lab 02**

*GVHD: Trần Mạnh Hùng*

*Ngày thực hiện: 2/4/2023*

**THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT109.O21.MMCL.

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Nguyễn Văn Anh Tuấn | 21522757 | [21522757@gm.uit.edu.vn](mailto:21522757@gm.uit.edu.vn) |

1. **ĐÁNH GIÁ KHÁC:**

|  |  |
| --- | --- |
| **Nội dung** | **Kết quả** |
| Tổng thời gian thực hiện bài thực hành trung bình | 4 ngày |
| Link Video thực hiện  *(nếu có)* |  |
| Ý kiến *(nếu có)*  + Khó khăn  + Đề xuất … | * Làm giao diện với awt mất khá nhiều thời gian. Bài 1 mọi biến số đều là Double nên in ra màn hình bị dư số 0. |
| Điểm tự đánh giá | 10/10 |

BÁO CÁO CHI TIẾT

**Bài 1: Calculator sử dụng java awt**

Tổng quan giao diện :

A screenshot of a calculator

Description automatically generated

Hình 1.1

Phân lớp giao diện:

A screenshot of a computer screen

Description automatically generated

Hình 1.2

Đầu tiên khởi tạo một đối tượng Frame sử dụng BorderLayout để chứa tất cả các components.

A black background with blue and pink text

Description automatically generated

Hình 1.3

Chia ra làm 3 phần chính là Panel1, Panel2 và MenuBar

A computer code on a black background

Description automatically generated

Hình 1.4

Panel1 : Chứa 2 label. Label1 để hiển thị đáp án và số nhập vào. Label2 dùng để hiện thị phép tính.

A screen shot of a computer program

Description automatically generated

Hình 1.5

Panel2 : Chứa tất cả các button của máy tính.

Trước tiên khởi tạo 2 list để chứa tên và buttons.

A computer screen with many white and blue text

Description automatically generated

Hình 1.6

Sử dụng vòng lặp for duyệt qua btnName và khởi tạo cho từng Button tương ứng:

 p2 = new Panel(new GridBagLayout());

        GridBagConstraints c = new GridBagConstraints();

        for (int i = 0; i < btnName.length; i++) {

            buttons[i] = new Button(btnName[i]);

            if (btnName[i].equals("0")) {

                c.fill = GridBagConstraints.BOTH;

                c.gridx = 0;

                c.gridy = 5;

                c.gridwidth = 2;

                c.gridheight = 1;

                c.weightx = 2;

                c.weighty = 1;

            } else if (btnName[i].equals("=")) {

                c.fill = GridBagConstraints.BOTH;

                c.gridx = 4;

                c.gridy = 4;

                c.gridwidth = 1;

                c.gridheight = 2;

                c.weightx = 1;

                c.weighty = 2;

            } else if (btnName[i].equals(".") || btnName[i].equals("+")) {

                c.fill = GridBagConstraints.BOTH;

                c.gridx = (i % 5) + 1;

                c.gridy = i / 5;

                c.gridwidth = 1;

                c.gridheight = 1;

                c.weightx = 1;

                c.weighty = 1;

            } else {

                c.fill = GridBagConstraints.BOTH;

                c.gridx = i % 5;

                c.gridy = i / 5;

                c.gridwidth = 1;

                c.gridheight = 1;

                c.weightx = 1;

                c.weighty = 1;

            }

            p2.add(buttons[i], c);

            buttons[i].addActionListener((ActionListener) this);

        }

Hình 1.7

Đối với 2 Button đặc biệt có kích thước khác với các Button khác là “0” và “=”, tạo điều kiện khi gặp btn “0” thì tăng width = 2, height =1 , gặp “=” thì tăng height = 2, width = 1

Sau đó, thêm button vào mảng buttons để quản lý và đồng thời addActionListener cho từng button để lắng nghe sự kiện khi được nhấn.

Tạo MenuBar với 3 MenuList là “View”, “Edit”, “Help”.

 menuBar = new MenuBar();

        Menu viewMenu = new Menu("View");

        Menu editMenu = new Menu("Edit");

        Menu helpMenu = new Menu("Help");

        menuBar.add(viewMenu);

        menuBar.add(helpMenu);

        menuBar.add(editMenu);

Hình 1.8

XỬ LÝ SỰ KIỆN:

Trước tiên, phải Implements ActionListener vào class.



Hình 1.9

Sau đó Override lại hàm actionPerformed() để xử lý sự kiện với từng nút.

A black background with purple and blue text

Description automatically generated

Hình 1.10

Khởi tạo các biến và danh sách cần thiết:

A computer code with colorful text

Description automatically generated with medium confidence

Hình 1.11

Num1,num2 để lưu giá trị cần tính toán

Biến isCalculated để kiểm tra xem button “=” đã được kích hoạt trước đó hay chưa

memoryList để chứa các giá trị muốn lưu vào Memory

Biến memoryListIsEmpty để kiểm tra xem memory có rỗng hay không

operatorQueue để chứa các toán hạng và toán tử.



Biến z để chứa chuỗi số và zt để chứa số hạng được nhập vào.

* Xử lý các nút số hạng: dưới đây là xử lý chung cho các số hạng

A screen shot of a computer

Description automatically generated

Hình 1.12 (Số hạng 0)

* Xử lý các toán tử cộng, trừ, nhân, chia, chia lấy dư: dưới đây là xử lý chung cho các toán tử.

A computer screen shot of a program code

Description automatically generated

Hình 1.13 ( Toán tử +)

Giải thuật: khi click vào button

Đầu tiên kiểm tra nếu operatorQueue rỗng thì thêm số hạng đầu tiên và toán tử

Lần lượt là operatorQueue[0] và [1].

Nếu operatorQueue không rỗng thì có nghĩa là user muốn cộng nhiều số hạng lại với nhau do đó ta tính kết quả của phép tính lưu ở trong Queue và lưu kết quả ra label2. Sau đó clear Queue.

* Xử lý Backspace:

A screen shot of a computer program

Description automatically generated

Hình 1.14

Giải thuật: Khi user muốn xóa 1 chữ số bị nhập sai.

Tạo newText để chứa dòng dữ liệu mới từ 0 –> n – 1 ( n là độ dài của dữ liệu cũ) sau đó hiển thị lại kêt quả vào l1.

* Xử lý CE:

A computer screen shot of text

Description automatically generated

Hình 1.15

Kiểm tra xem nếu như phép tính đã được tính thì clear l2, l1. Nếu chưa thì chỉ clear l1.

* Xử lý C:

A screen shot of a computer

Description automatically generated

Hình 1.16

Clear l1, l2, operatorQueue và memoryList.

* Xử lý 1/x:

A screen shot of a computer code

Description automatically generated

Hình 1.17

Tính kết quả 1/x sau đó lưu vào biến ans. Vì màn hình không thể hiển thị các số vô tỉ nên phải làm tròn kết quả để có thể hiện thị lên màn hình. Sử dụng DoubleRounder làm tròn xuống 6 số sau dấu thập phân.

* Xử lý ±:

A computer screen shot of a program code

Description automatically generated

Hình 1.18

Nếu l1 rỗng thì không làm gì. Nếu không rỗng thì nhân số cho -1 sau đó hiển thị ra l1.

* Xử lý căn bậc:

A computer screen shot of text

Description automatically generated

Hình 1.19

Sử dụng hàm Math.sqrt để tính căn bậc 2. Vì màn hình không thể hiện thị hết kết quả của số vô tỉ do đó sử dụng hàm DoubleRounder để làm tròn kết quả sau dấu thập phân.

* Xử lý MC:

A computer screen shot of a computer code

Description automatically generated

Hình 1.20

Sử dụng method clear() để clear memory và set memoryListIsEmpty = true

* Xử lý MR:

A computer screen shot of a program code

Description automatically generated

Hình 1.21

Kiểm tra nếu memoryList rỗng thì clear l1,l2. Nếu không thì in ra phần tử cuối cùng của memoryList.

* Xử lý MS:

A screen shot of a computer code

Description automatically generated

Hình 1.22

Thêm giá trị ở l1 vào cuối memoryList và set memoryListIsEmpty = false

* Xử lý M+:

A computer screen with text

Description automatically generated with medium confidence Hình 1.23

Kiểm tra xem nếu memoryList rỗng thì thêm giá trị của l1 vào memoryList. Nếu không thì cộng giá trị hiện tại của l1 với giá trị cuối cùng của memoryList.

* Xử lý M-:

A screen shot of a computer program

Description automatically generated

Hình 1.24

Kiểm tra xem nếu memoryList rỗng thì thêm giá trị hiện tại của l1 vào memoryList. Nếu không thì trừ giá trị hiện tại của l1 với giá trị cuối cùng của memoryList.

**Bài 2 : Viết một chương trình sử dụng java swing**

TO-DO APP

Tổng quan giao diện:

A screenshot of a computer

Description automatically generated

Hình 2.1

Giao diện bao gồm:

* Bao gồm một ô nhập liệu và nút "Thêm" để thêm công việc mới.
* Một danh sách hiển thị các công việc đã được thêm vào.
* Nút "Xóa" để xóa các công việc đã chọn từ danh sách.

Sử dụng ArrayList để lưu trữ danh sách các công việc:

A computer code with white and orange text

Description automatically generated

Hình 2.2

XỬ LÝ SỰ KIỆN:

* Xử lý nút Add:

A computer screen shot of text

Description automatically generated

Hình 2.3

Thêm task mới vào danh sách tasks và add task vào listModel để có hiện thị ra màn hình.

A computer screen shot of a program code

Description automatically generated

Hình 2.4

Nếu newTask không rỗng thì thêm newTask vào list và set taskInput về rỗng.

* Xử lý nút delete:

A computer code on a black background

Description automatically generated

Hình 2.5

Xóa task khỏi danh sách tasks và xóa task khỏi listModel để không hiển thị trên màn hình.

A screen shot of a computer code

Description automatically generated

Hình 2.6

Xóa task với index được chọn.

**HẾT**