

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**



# **BÁO CÁO**

**MÔN HỌC: ĐỒ ÁN 2**

**ĐỀ TÀI: XÂY DỰNG PHẦN MỀM GIẢI TRÒ  
CHƠI SUDOKU TRÊN NỀN TẢNG LINUX**

**Giáo viên hướng dẫn:**

**Ths. Nguyễn Trần Thi Văn**

**Sinh viên thực hiện:**

**Nguyễn Anh Tuấn - 16110243**

*TP. Hồ Chí Minh, ngày 10 tháng 05 năm 2019*

## NHẬN XÉT

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

# MỤC LỤC

LỜI MỞ ĐẦU.....	4
A. GIỚI THIỆU TỔNG QUAN.....	5
I. <i>Đặc tả đề tài:</i> .....	5
II. <i>Nền tảng xây dựng ứng dụng:</i> .....	5
III. <i>Mục tiêu:</i> .....	5
IV. <i>Đối tượng sử dụng ứng dụng:</i> .....	5
B. NỘI DUNG .....	6
I. <i>Thiết kế giao diện ý niệm:</i> .....	6
II. <i>Thiết kế code:</i> .....	10

## LỜI MỞ ĐẦU

Để từng bước cải thiện đáp ứng nhu cầu của các doanh nghiệp sau khi tốt nghiệp. Vì những điều đó học phần Đồ án 2 được nhà trường đưa vào chương trình đào tạo. Thông qua đó ôn lại những kiến thức cũ sau các kì học đã qua, củng cố thêm những kĩ năng một cách tốt nhất cho đồ án tốt nghiệp ra trường.

Trò chơi Sudoku là trò chơi quen thuộc và dễ sử dụng, nhưng để giải được nó là cả một quá trình suy nghĩ và tốn khá nhiều thời gian. Trên cơ sở đó, em đã suy nghĩ và tìm hiểu, sau đó em chọn đề tài “xây dựng phần mềm giải trò chơi Sudoku trên nền tảng hệ điều hành Linux” sử dụng ngôn ngữ Java, nhằm giúp những người chơi rút ngắn được thời gian để giải trò chơi này.

Em thực hiện dưới sự hướng dẫn của Ths. Nguyễn Trần Thi Văn. Em chân thành cảm ơn thầy vì những giải đáp và góp ý giúp em hiểu đúng yêu cầu mà thầy đã đặt ra và hoàn thành nó một cách tốt nhất đề tài này!

## **A. GIỚI THIỆU TỔNG QUAN**

### ***I. Đặc tả đề tài:***

Viết phần mềm giả trò chơi Sudoku trên nền tảng Linux.

- Phần mềm giải game Sudoku giao diện gồm 3 button Reset, Solve, Close.
- Giao diện bàn giải game gồm có 9x9 ô số là 9 hàng và 9 cột, được chia làm 3x3 vùng, mỗi vùng gồm các số từ 1 tới 9.
- Khi người chơi mở ứng dụng lên và nhập các số trên đề vào các ô tương ứng trên ứng dụng (các số nhập vào sẽ có màu đỏ).
- Khi người chơi đã nhập (từ bàn phím) xong các số trên đề bài vào các ô tương ứng thì sẽ nhấn vào button Solve thì kết quả sẽ hiện ra trên các ô còn lại của ứng dụng (các số kết quả có màu đen).
- Khi người chơi muốn xóa hết các ô cùng một lúc thì sẽ nhấn vào button Reset.
- Người chơi không muốn sử dụng ứng dụng nữa có thể nhấn vào button Close.

### ***II. Nền tảng xây dựng ứng dụng:***

- Xây dựng trên nền tảng: Linux (Ubuntu).
- Xây dựng trên ngôn ngữ lập trình: Java.
- Các kiến thức áp dụng: kiến thức về cấu trúc dữ liệu và giải thuật.
- Thuật toán áp dụng: Quay lùi, vét cạn.

### ***III. Mục tiêu:***

Xây dựng được phần mềm giải game Sudoku để giúp người chơi game Sudoku tiết kiệm được thời gian, và suy nghĩ.

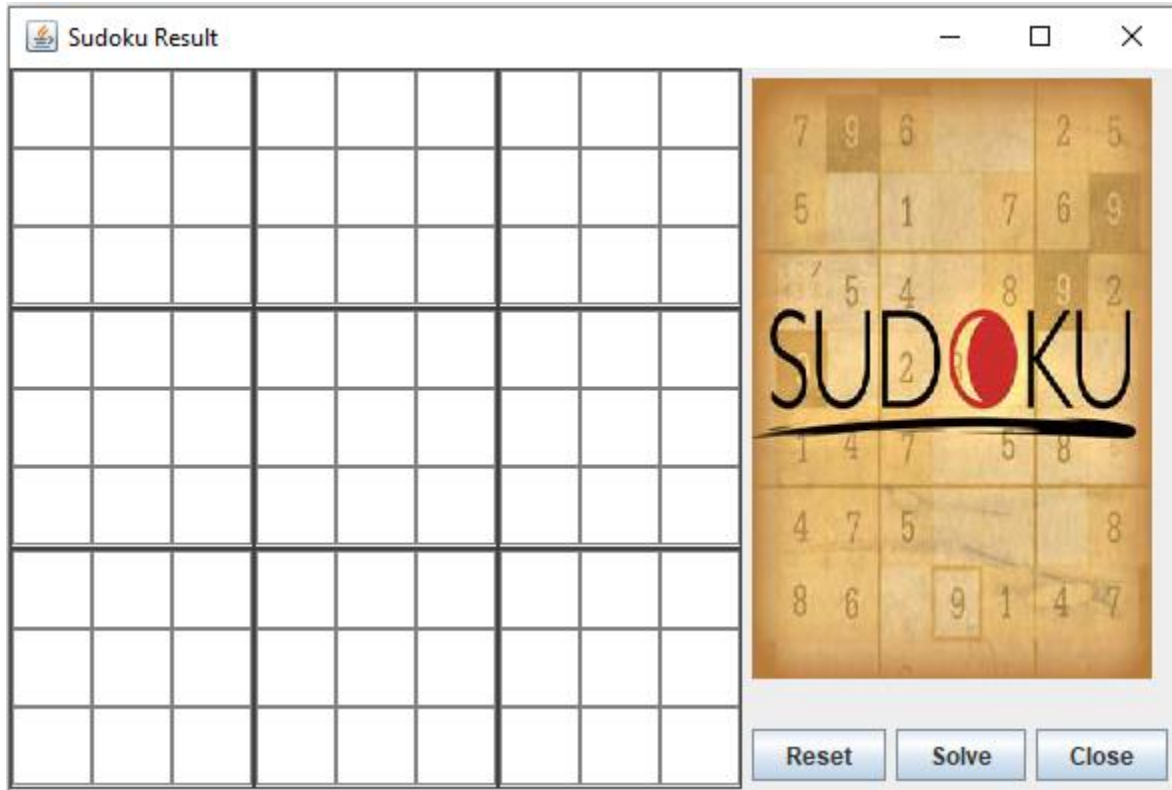
### ***IV. Đối tượng sử dụng ứng dụng:***

Giành cho những người chơi game Sudoku mà không muốn suy nghĩ nhiều hoặc đề bài quá khó.

## B.NỘI DUNG

### I. Thiết kế giao diện ý niệm:

#### 1. Form chính:



Hình 1: Form chính

Khi người dùng mở ứng dụng lên thì giao diện trên chính là giao diện chính của ứng dụng. và ứng dụng cũng chỉ có một màn hình giao diện trên. Mọi thao tác của người dùng đều được thực hiện trên màn hình giao diện trên.

## 2. Form sau khi nhập dữ liệu:

The screenshot shows a window titled "Sudoku Result". On the left is a 9x9 grid. The grid contains the following numbers in red: Row 2, Column 2 is 1; Row 2, Column 5 is 2; Row 4, Column 5 is 3; Row 8, Column 2 is 6; Row 8, Column 8 is 5. On the right side of the window is a decorative panel with a wooden texture. It features the word "SUDOKU" in a stylized font, with a red circle replacing the letter 'O'. Above the text is a small 3x3 grid of numbers. Below the text is another 3x3 grid of numbers. At the bottom of the panel are three buttons: "Reset", "Solve", and "Close".

Hình 2: Form sau khi nhập dữ liệu

Dữ liệu được người dung nhập vào bằng bàn phím sẽ có màu đỏ.

### 3. Form kết quả:

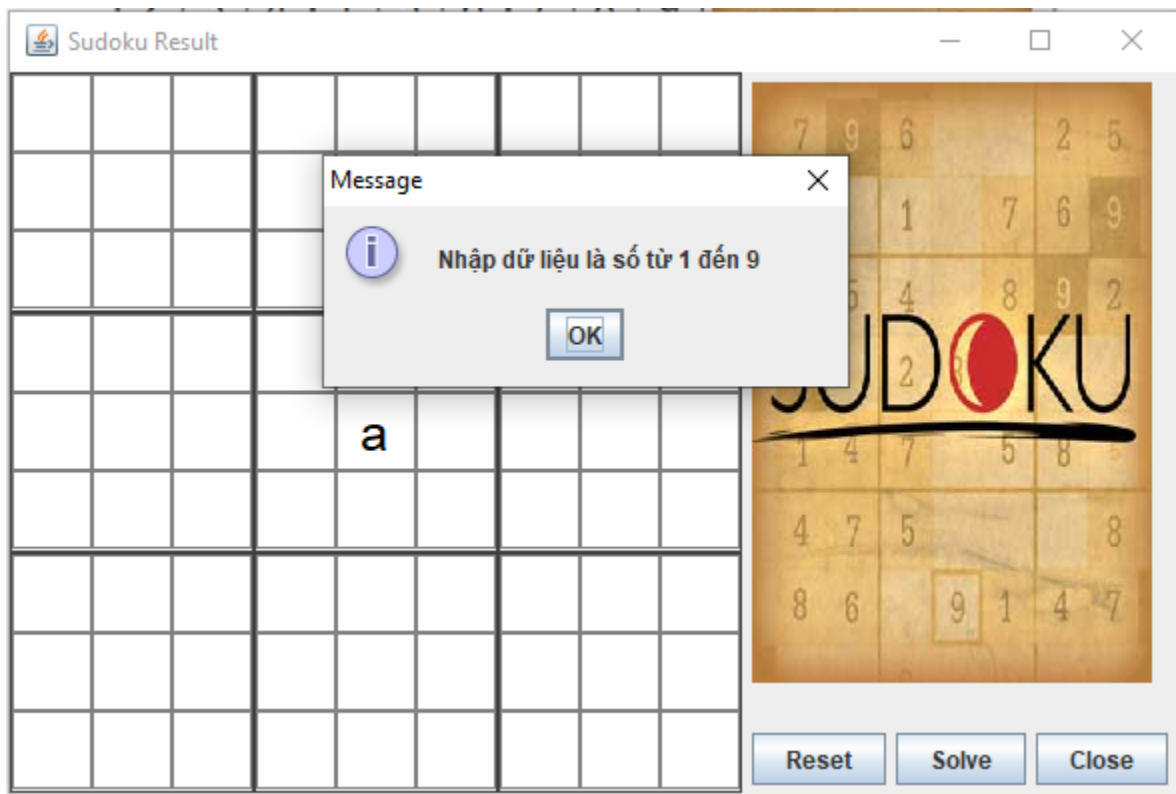
2	3	4	1	5	6	7	8	9
5	1	7	8	2	9	3	4	6
6	8	9	3	4	7	1	2	5
1	2	3	4	6	5	8	9	7
4	7	5	9	3	8	2	6	1
8	9	6	2	7	1	5	3	4
3	4	1	5	9	2	6	7	8
9	6	8	7	1	3	4	5	2
7	5	2	6	8	4	9	1	3

Hình 3: Form kết quả

Sau khi nhập dữ liệu (các số màu đỏ) xong. Khi người dùng nhấn vào button “Solve” kết quả sẽ hiện ra. Các số có chữ màu đỏ là đề bài. Còn các chữ số màu đen là đáp án ta cần tìm.

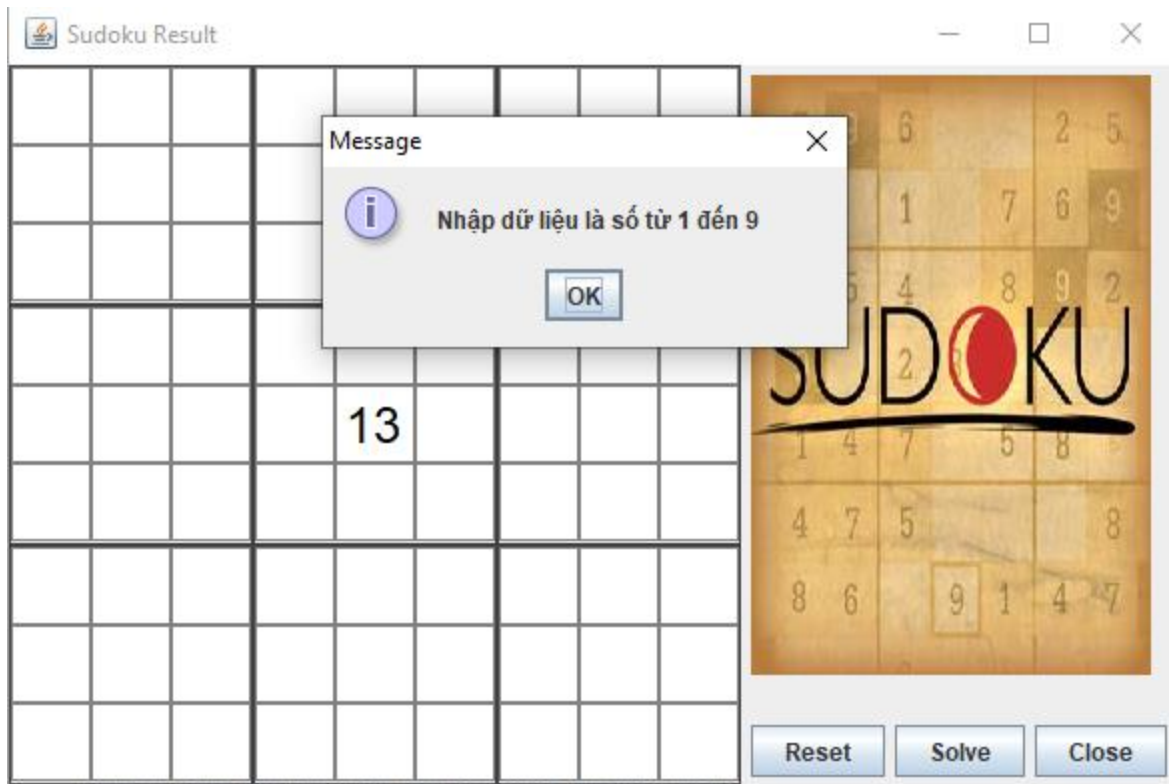


#### 4. Form nhập sai dữ liệu:



Hình 4: form nhập chữ

Sau khi nhập dữ liệu là 1 chữ cái thì kết quả sẽ trả về 1 thông báo “nhập dữ liệu là số từ 1 đến 9” sau khi bấm “OK” thì ô chứa chữ cái sẽ bị xóa.



Hình 5: Form nhập số khác từ 1 tới 9

Sau khi nhập dữ liệu là 1 số lớn hơn 9 hoặc nhỏ hơn 1 thì sẽ hiện ra 1 thông báo cảnh báo “nhập dữ liệu là số từ 1 đến 9” và sau khi bấm “OK” thì ô chứa số bị sai sẽ bị xóa trống.

## II. Thiết kế code:

Bảng 1: Bảng danh mục các hàm của class *JframeSudoku*

STT	Tên hàm	Sinh viên phụ trách	Mục đích
1	Public JframeSudoku()		Tạo một Jframe mới tên “Sudoku Result”. Với chiều dài 600px và chiều rộng 400px.

2	Public void addPanelEAST(Jpanel jp)		Thêm 1 Jpanel vào bên phải JFrame.
3	Public void addPanelCENTER(Jpanel jp)		Thêm 1 Jpanel vào chính giữa JFrame.

*Bảng 2: Bảng danh mục các hàm của class MatrixPanel*

STT	Tên hàm	Sinh viên phụ trách	Mục đích
1	Public MatrixPanel()		Tạo một ma trận panel 3x3.
2	Public void CreateMatrixPanel()		Trong mỗi 1 panel tạo thêm 3x3 panel nhỏ (tạo thành ma trận 9x9).
3	Public void addTextField(JTextField jl)		Với mỗi panel thì chứa một Text Field. Kiểm tra dữ liệu đầu vào.
4	Public Boolean ktdulieuvao()		Kiểm tra xem dữ liệu nhập vào phải là số từ 1 đến 9 không.
5	Private Boolean kiemtratrungso()		Kiểm tra các số ở các ô ở cùng hàng, cùng cột và ma trận 3x3 không bị trùng nhau.

*Bảng 3: Bảng danh mục các hàm của class EASTPanel*

STT	Tên hàm	Sinh viên phụ trách	Mục đích
1	Public EASTPanel()		Tạo 2 panel ở trên và ở dưới ở panel bên phải.(ở trên chứa hình, ở dưới chứa 3 button chức năng).
2	Public void addpbc()		Thêm panel vào vị trí bên dưới của panel bên phải. (ở JFrame chia làm 2 panel bên trái và bên phải).
3	Public void addLabel(JLabel label)		Thêm label vào panel bên trên của panel bên phải. (ở JFrame chia làm 2 panel bên trái và bên phải).
4	Public void add plc()		Thêm panel vào vị trí bên trên của panel bên phải. (ở JFrame chia làm 2 panel bên trái và bên phải).

*Bảng 4: Bảng danh mục các hàm của class MatrixTextField*

STT	Tên hàm	Sinh viên phụ trách	Mục đích
1	Public MatrixTextField()		Định dạng font chữ, đường viền,... cho Text Field.
2	Public static JTextField[][] setMatrixTextField()		Tạo một ma trận Text Field 9x9.
3	Public static String[][] getMatrixTextField ()		Lấy giá trị của ma trận Text Field. Đồng thời kiểm tra màu chữ của các ô trong ma trận Text Field.

*Bảng 5: Bảng danh mục các hàm của class Butons*

STT	Tên hàm	Sinh viên phụ trách	Mục đích
1	Public Butons()		Khởi tạo 3 button: “Close”, “Solve”, “Reset”.
2	private void actionPerformed()		Tạo sự kiện click cho 3 button trên.
3	Public void ResetMatrix ()		Xóa tất cả các giá trị trong Text Field.

*Bảng 6: Bảng danh mục các hàm của class Labels*

STT	Tên hàm	Sinh viên phụ trách	Mục đích
1	Public Labels(ImageIcon icon, int center)		Thêm hình vào label, ở vị trí chính giữa label.
2	Public static JLabel createLabel()		Tạo một label với hình do mình cung cấp.

*Bảng 7: Bảng danh mục các hàm của class MatrixData*

STT	Tên hàm	Sinh viên phụ trách	Mục đích
1	Public MatrixData(String[][] data)		Khởi tạo một ma trận kiểu int từ một ma trận kiểu String.
2	private int[][] convertString2Int(String[][] str)		Đổi một ma trận từ kiểu String sang kiểu int.
3	Public String[][] convertInt2String(int[][] ints)		Đổi một ma trận từ kiểu int sang kiểu String.
4	Public int[][] getMatrix()		Trả về một ma trận kiểu int.

*Bảng 8: Bảng danh mục các hàm của class DataProcessing*

STT	Tên hàm	Sinh viên phụ trách	Mục đích
1	Public DataProcessing(int[][] matrix)		Khởi tạo một ma trận kiểu int từ một ma trận truyền vào.
2	public int CheckRowCol(int i, int j, int x)		Kiểm tra hàng, cột, trong ma trận 3x3 có số vừa truyền vào không.
3	Public Boolean CheckCoincideNumber(int x, int a, int b)		Kiểm tra xem hàng, cột đó có bị trùng số không.
4	Public int findCell0(int[] cell)		Tìm vị trí của ô Text Field có giá trị bằng 0.
5	Public int Solve()		Giải game bằng thuật toán quay lùi, vét cạn.
6	Public int[][] getMatrixInt()		Trả kết quả ma trận kiểu int về.

*Bảng 9: Bảng danh mục các hàm của class Main*

STT	Tên hàm	Sinh viên phụ trách	Mục đích
1	Public static void main(String[][] args)		Hàm chạy chương trình.

