Shape, square

Description automatically generated

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**UNIVERSITY OF TRANSPOST AND COMMUNICATIONS**

Logo

Description automatically generated

**BÁO CÁO BÀI TẬP LỚN**

**MÔN CÔNG NGHỆ JAVA**

***ĐỀ TÀI: Lập trình game cờ vua 2 người***

**Giảng viên hướng dẫn: Nguyễn Quốc Tuấn**

**Sinh viên thực hiện:**

**Lớp: CNTT6-K62**

**Năm 2023**

**MỤC LỤC**

[**I.** **Mở đầu: Giới thiệu mục đích của Bài tập lớn** 3](#_Toc134024095)

[**1.** **Kế thừa** 5](#_Toc134024096)

[**2.** **Bắt sự kiện trong Java AWT** 6](#_Toc134024097)

[**3.** **Giao diện tương tác với người dùng** 8](#_Toc134024098)

[**a.** **Gói Java AWT** 9](#_Toc134024099)

[**b.** **Gói Java Swing** 11](#_Toc134024100)

[**II.** **Thực thi và cài đặt** 13](#_Toc134024101)

[**1.** **Sơ đồ lớp** 13](#_Toc134024102)

[**2.** **Công nghệ được áp dụng** 15](#_Toc134024103)

[**a)** **Kế thừa:** 15](#_Toc134024104)

[**b)** **Giao diện tương tác với người dùng**: 15](#_Toc134024105)

[**c)** **Xử lí sự kiện:** 16](#_Toc134024106)

[**3.** **Cách chạy chương trình trên máy mới** 17](#_Toc134024107)

[**III.** **Kết luận** 17](#_Toc134024108)

1. **Mở đầu: Giới thiệu mục đích của Bài tập lớn**

Java là một công nghệ xây dựng các ứng dụng phần mềm có vị trí rất lớn trong những năm cuối thế kỉ 20, đầu thế kỉ 21. Nó được coi là công nghệ mang tính cách mạng và khả thi nhất trong việc tạo ra các ứng dụng có khả năng chạy thống nhất trên nhiều nền tảng mà chỉ cần biên dịch một lần.

Lần đầu tiên xuất hiện vào năm 1992 như là một ngôn ngữ dùng trong nội bộ tập đoàn Sun Microsystems để xây dựng ứng dụng điều khiển các bộ xử lý bên trong máy điện thoại cầm tay, lò vi sóng, các thiết bị điện tử dân dụng khác. Không chỉ là một ngôn ngữ, Java còn là một nền tảng phát triển và triển khai ứng dụng trong đó máy ảo Java, bộ thông dịch có vai trò trung tâm.

Sun, công ty đã phát minh ra ngôn ngữ Java, chính thức ban hành bản Java Development Kit 1.0 vào năm 1996 hoàn toàn miễn phí để các nhà phát triển có thể tải về, học Java, xây dựng các ứng dụng Java và triển khai chúng trên các hệ điều hành có hỗ trợ Java. Khi nhắc đến Java người ta không còn chỉ nhắc đến Java như là một ngôn ngữ mà nhắc đến Java như là một công nghệ hay một nền tảng phát triển.

Cũng như nhiều trường Đại học khác, Trường Đại Học Giao Thông Vận Tải từ lâu đã đưa Java vào chương trình giảng dạy cho sinh viên. Bài tập lớn này nhằm mục đích giúp sinh viên tìm hiểu và bước đầu làm quen với lập trình Java, lập trình những ứng dụng Java đơn giản có mục đích và tính hữu dụng trong thực tế.

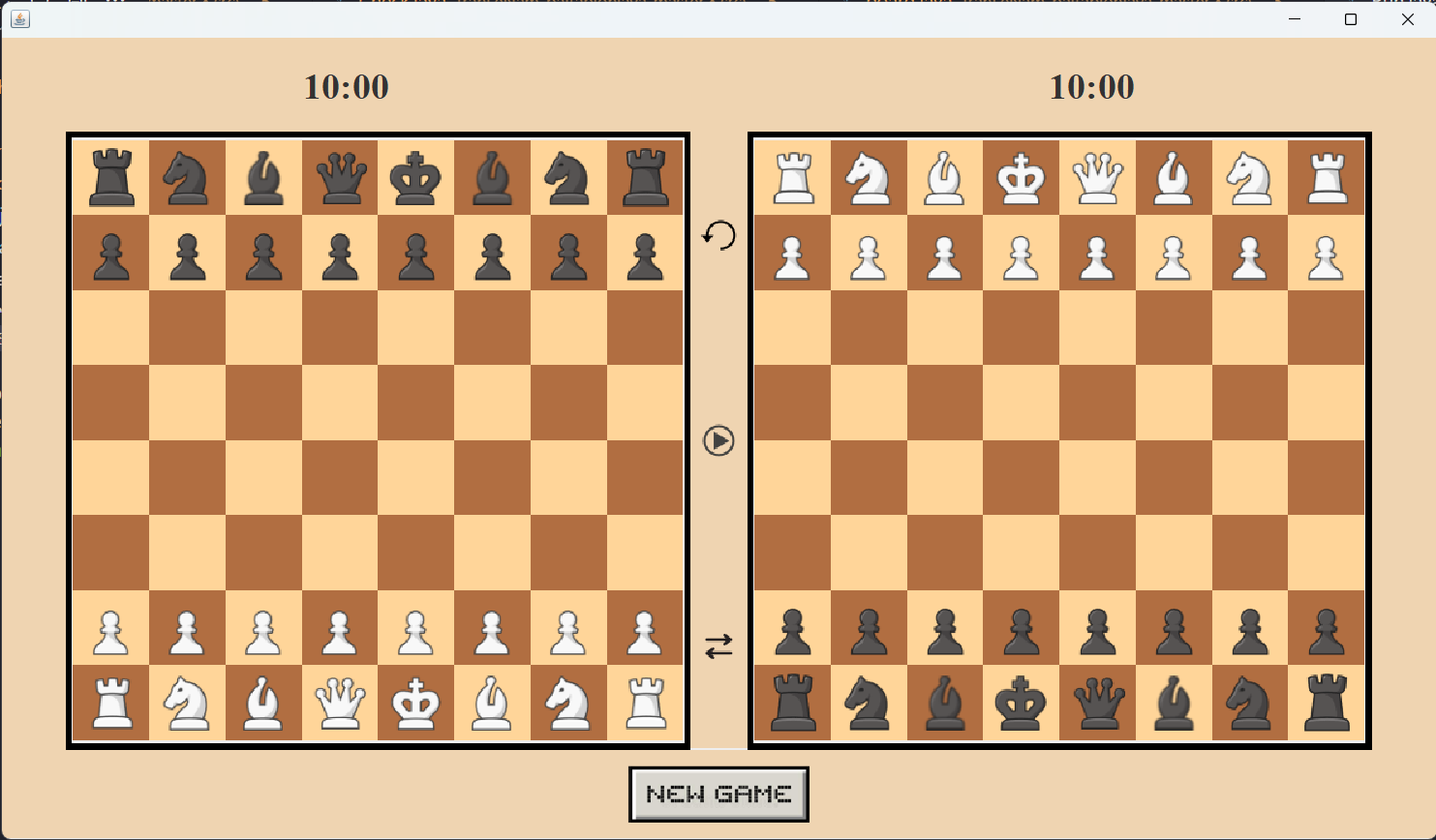
**2-Giới thiệu chương trình**

**Thông tin:**

Ứng dụng Game cờ vua 2 người được xây dựng nhằm mục đích làm quen với ngôn ngữ, các câu lệnh cơ bản và một số công nghệ hướng đối tượng tiên tiến.

Trò chơi cờ vua còn có rất nhiều lợi ích như:

* Rèn tính tập trung
* Giữ được sự bình tĩnh
* Rèn luyện trí não
* Khả năng tư duy cụ thể
* Kích thích trí tưởng tượng
* Tăng kỹ năng giải quyết vấn đề
* Khả năng phân tích tình huống
* Phát huy tính sáng tạo
* Rèn luyện tính kỉ luật



Hình ảnh của ứng dụng.

1. **Các công nghệ hướng đối tượng được được sử dụng trong ứng dụng**
2. **Kế thừa**

Tính kế thừa trong Java là môt kỹ thuật mà trong đó một đối tượng thu được tất cả thuộc tính và hành vi của đối tượng cha. Ý tưởng đằng sau tính kế thừa trong Java là bạn có thể tạo các lớp mới mà được xây dựng dựa trên các lớp đang tồn tại. Khi bạn kế thừa từ một lớp đang tồn tại, bạn có thể tái sử dụng các phương thức và các trường của lớp cha, và bạn cũng có thể bổ sung thêm các phương thức và các trường khác. Tính kế thừa biểu diễn mối quan hệ IS-A, còn được gọi là mối quan hệ cha-con.

Khi chúng ta nói về tính kế thừa, từ khóa thường xuyên nhất được sử dụng là **extends** và **implements**. Những từ khóa này có thể định nghĩa một kiểu là loại IS-A của loại khác. Sử dụng những từ khóa, chúng ta có thể tạo một đối tượng sử dụng thuộc tính của đối tượng khác. Chúng ta sử dụng từ khóa extends của lớp con để có thể kế thừa các thuộc tính của lớp cha trừ các thuộc tính private của lớp cha.

Cú pháp của Tính kế thừa trong Java

class ten\_lop\_con extends ten\_lop\_cha

{

//cac phuong thuc va cac truong

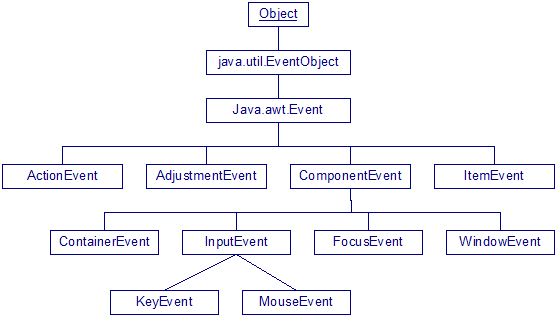
}

1. **Bắt sự kiện trong Java AWT**

Các hệ thống GUI xử lý các tương tác người dùng với sự trợ giúp của mô hình hướng sự kiện (event-driven).

Sự kiện trong Java:

* Sự kiện có thể được định nghĩa như là một kiểu tín hiệu tới chương trình thông báo một việc gì đó đã xảy ra.
* Sự kiện được sinh ra bởi các hành động của người dùng, ví dụ như di chuyển, click chuột, hay nhấn phím....



**Cây phân cấp các lớp của gói Event**

Việc xử lý những sự kiện này phụ thuộc vào ứng dụng. Abstract Windowing Toolkit (AWT) xử lý một vài sự kiện. Môi trường mà các ứng dụng này được thi hành ví dụ như trình duyệt cũng có thể xử lý các sự kiện khác. Người lập trình cũng cần phải xử lý những sự kiện nhất định và cần phải viết hàm xử lý các sự kiện đó.

Ứng dụng cần đăng ký một hàm xử lý sự kiện với một đối tượng. Hàm xử lý sự kiện này sẽ được gọi bất cứ khi nào sự kiện tương ứng phát sinh. JDK1.2 làm việc theo mô hình xử lý sự kiện này.

Trong quy trình này, ứng dụng cho phép bạn đăng ký các phương thức (handler), hay gọi là listener với các đối tượng. Những handler này tự động được gọi khi một sự kiện thích hợp phát sinh.

Một Event Listener lắng nghe một sự kiện nào đó mà một đối tượng đã thiết lập. Mỗi event listener cung cấp các phương thức xử lý những sự kiện này. Lớp thi hành listener cần phải định nghĩa những phương thức này. Để sử dụng mô hình này, bạn làm theo các bước sau:

* Cài đặt giao diện listener thích hợp. Cấu trúc như sau:

**public class MyApp extends Frame implements ActionListener**

* Xác định tất cả các thành phần tạo ra sự kiện. Các thành phần có thể là các button, label, menu item, hay window.

Cho ví dụ, để đăng ký một thành phần với listener, ta có thể sử dụng:

**exitbtn.addActionListener(This);**

Xác định tất cả các sự kiện được xử lý. Các sự kiện có thể là một ‘ActionEvent’ nếu một button được click hay một ‘mouseEvent’ nếu như chuột được kéo đi.

Thi hành các phương thức của listener và viết hàm xử lý sự kiện tương ứng với các phương thức.

|  |  |
| --- | --- |
| Lớp sự kiện | Mô tả |
| ActionEvent | Phát sinh khi một button được nhấn, một item trong danh sách chọn lựa được nhấn đúp (double-click) hay một menu được chọn. |
| TextEvent | Phát sinh khi giá trị trong thành phần textfield hay textarea bị thay đổi. |
| MouseEvent | Phát sinh khi chuột di chuyển, được click, được kéo hay thả ra. |
| KeyEvent | Phát sinh khi bàn phím ấn, nhả. |
|  |  |
|  |  |
| BẢNG 1: Các sự kiện khác nhau được sử dụng trong ứng dụng và mô tả về chúng | |

Các giao diện cần được cài đặt để xử lý một trong số những sự kiện này là:

* + ActionListener
  + TextListener
  + MouseListener
  + MouseMotionListener
  + KeyListener

Các giao diện định nghĩa một số phương thức để xử lý mỗi sự kiện. Những phương thức này sẽ được nạp chồng trong lớp mà cài đặt những giao diện này.

1. **Giao diện tương tác với người dùng**

Các ứng dụng phần mềm hiện nay rất thân thiện vì được trình bày nhiều màn hình giao diện đồ họa đẹp mắt. Các ngôn ngữ lập trình hiện nay cung cấp các đối tượng đồ họa, chúng có thể được điều khiển bởi người lập trình, hay bởi người sử dụng. Một trong số những kết quả quan trọng nhất chính là các ngôn ngữ hiện nay được dựa trên Giao diện người dùng đồ họa (Graphical User Interface - GUI).

  GUI cung cấp chức năng nhập liệu theo cách thân thiện với người dùng. GUI đa dạng từ ứng dụng đến ứng dụng và có thể chứa nhiều điều khiển như hộp văn bản, nhãn, hộp danh sách hay các điều khiển khác. Các ngôn ngữ lập trình khác nhau cung cấp nhiều cách khác nhau để tạo GUI. Các ngôn ngữ như VB hay VC++ có thể cung cấp chức năng kéo và thả trong khi đó phần mềm giống như C++ yêu cầu người lập trình phải viết toàn bộ mã để xây dựng GUI.

Một phần tử (element) GUI được thiết lập bằng cách sử dụng thủ tục sau:

* Tạo đối tượng
* Xác định sự xuất hiện ban đầu của đối tượng
* Chỉ ra nó nằm ở đâu
* Thêm phần tử vào giao diện trên màn hình

Một thành phần (component) GUI là một đối tượng trực quan. Người dùng tương tác với đối tượng này thông qua con trỏ chuột hay bàn phím. Các thành phần như là button, label v.v… có thể được nhìn thấy trên màn hình. Bất kỳ cái gì chung cho tất cả các thành phần GUI đều được tìm thấy trong lớp Component. Để tạo các đối tượng GUI chúng ta cần nhập gói **java.awt** hoặc gói **javax.swing**.

Giao diện tương tác với người dùng của ứng dụng game cờ Caro được xây dựng dựa trên các gói **java.awt** và **javax.swing**. Sau đây là các lớp được sử dụng trong ứng dụng từ 2 gói trên.

1. **Gói Java AWT**

* **Lớp Color**
* Màu được tạo từ 3 màu: đỏ , xanh lá cây hoặc xanh da trời. Mỗi màu được biểu diễn bởi một giá trị kiểu byte, từ 0 (tối nhất) đến 255 (sáng nhất) -> *RGB model*.

Color c = new Color(r, g, b);

* r, g, and b chỉ tới 1 màu đỏ, xanh lá cây, xanh da trời bởi các components của nó.

Ví dụ:

Color c = new Color(228, 100, 255);

* 13 màu chuẩn **(black, blue, cyan, darkGray, gray, green, lightGray, magenta, orange, pink, red, white, yellow)** được định nghĩa như hằng số trong java.awt.Color.
* Từ JDK 1.4: BLACK, BLUE, CYAN, DARK\_GRAY, GRAY, GREEN, LIGHT\_GRAY, MAGENTA, ORANGE, PINK, RED, WHITE, and YELLOW
* Thiết lập màu nền (background) và nổi (foreground) :

**setBackground(Color c)**

**setForeground(Color c)**

Ví dụ:

**jbt.setBackground(Color.yellow);**

**jbt.setForeground(Color.red);**

* **Font:**

Ví dụ:

**batdau.setFont(new java.awt.Font("Times New Roman", 0, 24));**

* **Lớp Graphics:**

Lớp trừu tượng Graphics cung cấp giao tiếp đồ hoạ độc lập với thiết bị để hiện thị hình minh hoạ và ảnh trên màn hình của các nền tảng khác nhau. Bất kể thành phần nào (button, label ... ) được hiển thị, một đối tượng Graphic sẽ được tạo trên nền tảng gốc (native). Có thể lấy về đối tượng này thông qua phương thức getGraphics(). Ví dụ, đề lấy về graphic của một đối tượng nhãn jlblBanner

Graphics graphics = jlblBanner.getGraphics();

Sau đó, có thể sử dụng các phương thức của lớp Graphics để vẽ lên nhãn này.

Phương thức paintComponent thuộc lớp JComponent :

**protected void paintComponent(Graphics g)**

Phương thức này được gọi mỗi khi hiển thị hoặc hiện thị lại.

1. **Gói Java Swing**

****

Cấu trúc cây của gói Swing

* **Frame:**
* cửa sổ không nằm bên trong cửa sổ khác; thành phần cơ bản để chứa các GUI khác.
* Lớp JFrame được dùng đề tạo cửa sổ.
* Với các chương trình GUI Swing -> sử dụng JFrame -> tạo cửa sổ.

Cách tạo một Frame:

Ví dụ:

import javax.swing.\*;

public class MyFrame {

public static void main(String[] args) {

JFrame frame = new JFrame("Test Frame");

frame.setSize(400, 300);

frame.setVisible(true);

frame.setDefaultCloseOperation(JFrame.

EXIT\_ON\_CLOSE);

}

}

Thêm một button vào frame

**frame.getContentPane().add(new JButton("OK"));**

* **Layout managers**
* Trình quản lý layout (sự bố trí) của Java cung cấp một mức độ trừu tượng hoá cho phép định hình giao diện lên tất cả các hệ thống cửa sổ.
* Các thành phần UI được đặt trong containe. Mỗi container có một trình quản lý layout để sắp xếp các thành phần này.
* Sử dụng phương thức setLayout(LayoutManager) của container.
* Grouplayout làm việc với các layout theo chiều ngang và chiều dọc riêng rẽ. Layout được xác định cho mỗi chiều một cách độc lập. Khi tập trung vào chỉ là một chiều, bạn chỉ có thể giải quyết một nửa vấn đề cùng một lúc. Điều này dễ dàng hơn so với việc xử lý cả hai chiều cùng một lúc. Điều này có nghĩa là, tất nhiên, rằng mỗi thành phần cần phải được định nghĩa hai lần trong layout. Nếu bạn quên làm điều này, GroupLayout sẽ tạo ra một ngoại lệ.
* **Panels:**
* Là Container con để nhóm các thành phần UI.
* Recomend: đặt component, panel -> panel -> frameIl.
* Để thêm thành phần -> JFrame: thêm vào thành nội dung của JFrame. Để thêm thành phần vào panel -> thêm trực tiếp vào panel bằng phương thức add.
* Tạo một Jpanel:
* Sử dụng new JPanel() : layout mặc định là FlowLayout
* Sử dụng new JPanel(LayoutManager).
* Sử dụng phương thức add(Component)

**JPanel p = new JPanel();**

**p.add(new JButton("OK"));**

* **Image Icon**
* Java sử dụng các lớp javax.swing.ImageIcon để biểu diễn cho một biểu tượng. Một biểu tượng là một hình ảnh kích thước cố định.
* Hình ảnh được thường được lưu trong tập tin hình ảnh. Bạn có thể sử dụng ImageIcon mới (tên tập tin) để xây dựng một biểu tượng hình ảnh.
* Ví dụ, câu lệnh sau đây tạo ra một Icon từ một file ảnh us.gif trong thư mục hình ảnh theo đường dẫn lớp hiện tại:

**Icon ImageIcon = new ImageIcon ("image / us.gif");**

1. **Thực thi và cài đặt**
2. **Sơ đồ lớp**

Gồm 10 lớp:



* Lớp Run : để chạy chương trình
* Lớp board : chứa giao diện game
* Lớp Check : chứa các thuật toán gọi kiểm tra quân cờ được nhấn vào và gọi nước đi.
* Lớp variable chứa dữ liệu của ván đấu và các hàm tạo giao diện cho bàn cờ.
* 6 lớp còn lại chứa thuật toán riêng biệt quy định nước đi hợp lệ của từng loại quân cờ.

1. Lớp Run: sẽ gọi đến hàm khởi tạo của board

public static *void* main(String[] *args*) {

**new** board();

    }

1. Lớp board : khởi tạo giao diện của trò chơi và tạo luồng thời gian đếm ngược.
2. Lớp Check\* :

Class Check

Hàm Message(String)

Hàm Check(int a,int b)

Hàm playSound()

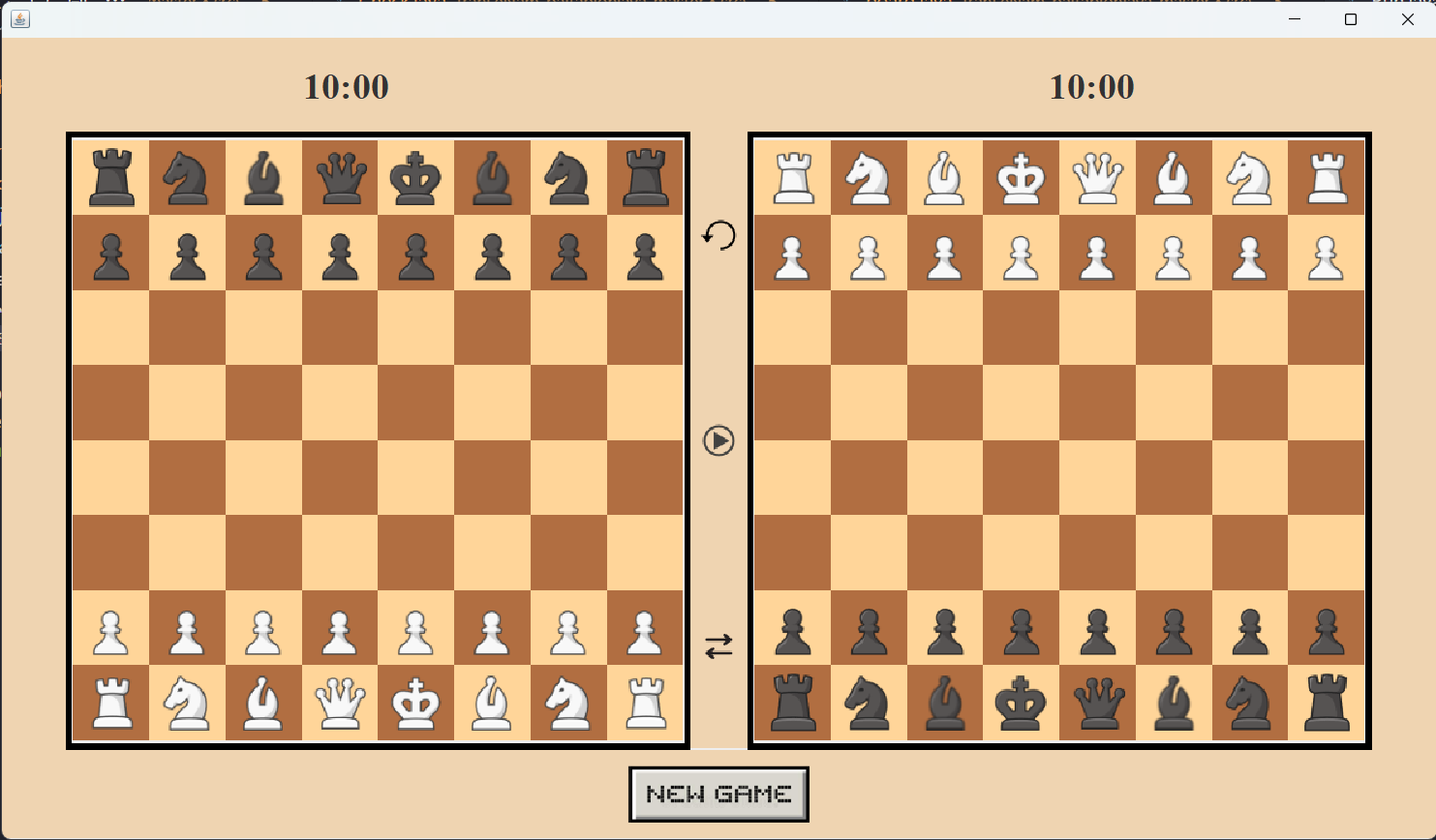
* hàm Check(int a, int b) : có tham số a b là tọa độ của ô đang xử lý trên bàn cờ. Hàm có chức năng nhận diện quân cờ và gọi các hàm của các quân cờ phù hợp.
* Hàm Message(String): Hàm truyền vào chuỗi kí tự để xuất ra màn hình khi một bên dành chiến thắng hoặc hai bên hòa cờ.
* Hàm playSound(): chức năng của hàm sau khi nước đi kết thúc sẽ phát ra âm thanh.

1. Lớp variable: chứa dữ liệu của ván đấu và có các hàm thay đổi trạng thái của ô cờ được gọi. Ngoài ra còn hàm Return() để in lại trạng thái bàn cờ ở nước trước đó.
2. Các lớp quân cờ: chứa các hàm active, off, run, an để thay đổi trạng thái ô cờ được gọi. Ngoài ra còn có các hàm đặc biệt để thực hiện các chức năng đặc biệt của mỗi loại quân cờ.
3. **Công nghệ được áp dụng**
4. **Kế thừa:**

Trừ lớp Run tất cả các lớp còn lại đều kế thừa lớp variable để sử dụng dữ liệu chung.

1. **Giao diện tương tác với người dùng**:

Sử dụng gói javax.swing và java.awt



Mỗi ô cờ là một jbutton. Đồng hồ sử dụng jlabel để in ra thời gian.

1. **Xử lí sự kiện:**

arr[i][j].addActionListener(

*e* *->* {

                        Check.check(g, h);}

                );

Pause.addActionListener(

*e* *->* {

                if(P){

                    timer1.stop();

                    timer2.stop();

                    Pause.setIcon(board.Pause);

                }else{

                    Pause.setIcon(Continue);

                    if(luot == "trang"){

                        timer1.start();

                    }else{

                        timer2.start();

                    }

                }

                P=!P;

                }

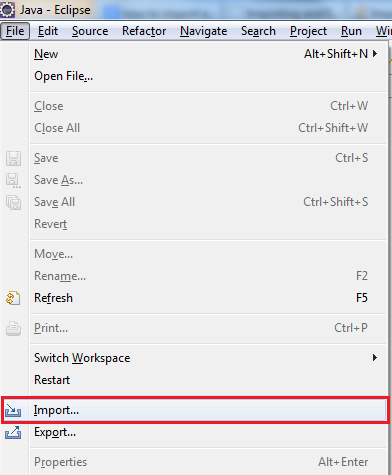
        );

Mỗi các phím bấm trong game đều được xử lí sự kiện click chuột.

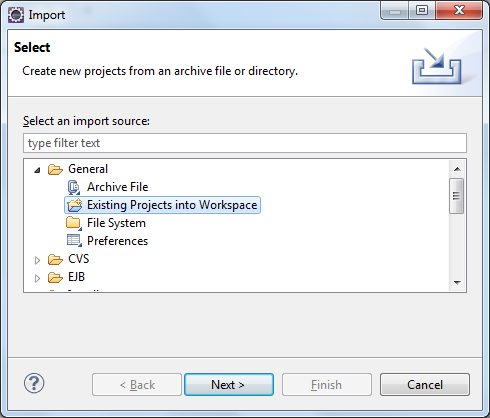
1. **Cách chạy chương trình trên máy mới**

Sử dụng các trình biên dịch java ví dụ như eclipse

Chọn File -> Import

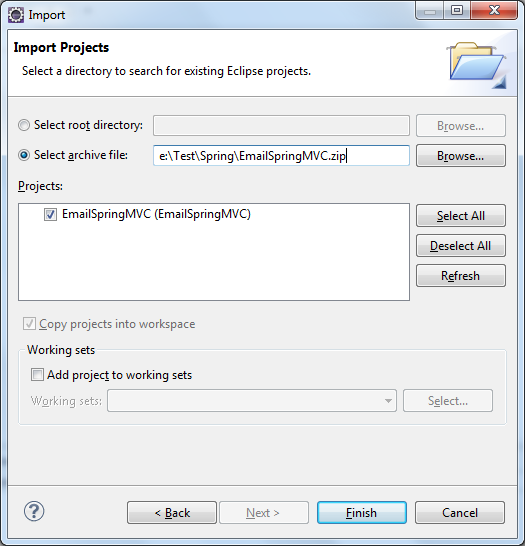
[](https://shareprogramming.net/wp-content/uploads/2021/03/Import-from-menu-File.png)

Sau đó Eclipse sẽ mở lên hộp thoại sau:

[](https://shareprogramming.net/wp-content/uploads/2021/03/Import-wizard-screen-1.png)

Chọn vào ***Existing Projects into Workspace*** và chọn ***Next****.* Hộp thoại tiếp theo sẽ mở ra.

Trong trường hợp bạn tải project ở dạng nén thì chúng ta có thể chỉ định trong phần **Select Archive file** trỏ đến file nén của bạn -> Chọn **Finish**.

[](https://shareprogramming.net/wp-content/uploads/2021/03/Import-wizard-screen-2.png)

Nếu là một thư mục thông thường thì chúng ta có thể chọn ***Select root directory*** và trỏ đến thư mục project. Chọn ***Finish***.

Ở file Run.java là file chứa hàm main có 2 cách để chạy chương trình

Cách 1: Ở trên thanh công cụ chọn Run -> Run (Phím tắt Ctrl + F11)  
  
Cách 2: Click biểu tượng Play trên thanh công cụ sau đó chọn OK

1. **Kết luận**

Hiện nay, lập trình game đang là một lĩnh vực mới thu hút nhiều lập trình viên. Việc xây dựng các ứng dụng vui chơi giải trí là rất cần thiết cho sự phát triển công nghệ. Trong phạm vi đề tài, chúng em chỉ trình bày những phần công nghệ cơ bản có liên quan đến ứng dụng như lập trình giao diện người dùng với Swing và AWT và bắt sự kiện với AWT.

Nhìn chung game tạo ra đáp ứng nhu cầu giải trí của người chơi. Giao diện thân thiện, đơn giản, dễ sử dụng và thao tác. Tuy nhiên vẫn còn một số lỗi như lỗi font chữ chưa được khắc phục, bàn cờ còn hạn chế, chưa đẹp mắt.

Tuy vậy, do những hạn chế về trình độ, thời gian cũng như thiết bị nên em không khỏi những vướng mắc, sai sót trong quá trình tìm hiểu và nghiên cứu. Em rất mong nhận được những đánh giá và chỉnh sửa từ các thầy cô hướng dẫn.

Em xin chân thành cảm ơn!

Mã nguồn dowloal: [https://github.com/tranlenam23/baitaplon.git](%20https:/github.com/tranlenam23/baitaplon.git)