

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

Khoa Công nghệ thông tin

Bộ môn Mạng và hệ thống thông tin

-----oOo-----



BÀI TẬP LỚN MÔN CÔNG NGHỆ JAVA

Chủ đề: Game Ai Là Triệu Phú

Giáo viên hướng dẫn: Vũ Huấn

Nhóm thực hiện: Nhóm 4-CNTT5

Thành viên nhóm:

211201935 Vũ Văn Hoàn

211204408 Hoàng Thị Hà

Hà Nội, ngày 2 tháng 5 năm 2023

Mục lục

I.	Tổng quan đề tài	3
1.	Giới thiệu đề tài	3
2.	Cách chơi	3
3.	Biểu đồ Use case.....	4
II.	Xây dựng chương trình.....	4
1.	Sơ đồ chung	4
2.	Giới thiệu và nêu chức năng của package	5
3.	Giới thiệu và các tương tác trong giao diện.....	5
3.1.	Kết nối database	5
3.2.	Trang bắt đầu chơi.....	6
3.3.	Trang chơi game	8
3.3.1.	Chức năng chọn đáp án.....	8
3.3.2.	Chức năng trợ giúp.....	10
3.3.3.	Mức độ câu hỏi.....	16
3.4.	Trang kết quả.....	18
3.5.	Trang bảng xếp hạng	20
III.	Bảng cơ sở dữ liệu	22
1.	Bảng câu hỏi	22
2.	Bảng người chơi.....	22
IV.	Kết luận	23

I. Tổng quan đề tài

1. Giới thiệu đề tài

"Ai là triệu phú" được chuyển thể từ phiên bản game phát trên VTV3 tới thứ 3 hàng tuần và trở nên quen thuộc với tất cả mọi người. Cơ hội để ngồi trên chiếc ghế nóng của VTV3 tại trường quay S9 là rất khó cho hầu hết mọi người, tựa game này ra đời đã đáp ứng được yêu cầu đó.


Đây là một thể loại game chơi trên chính sự hiểu biết của bạn về tất cả mọi vấn đề từ cổ chí kim, nhiều lĩnh vực trong cuộc sống với hình thức trả lời câu hỏi trắc nghiệm khách quan 4 lựa chọn với 3 quyền trợ giúp: 50/50, gọi điện thoại cho người thân, hỏi ý kiến khán giả trong trường quay và quyền ngưng cuộc chơi ở mức bất kỳ để bảo toàn số điểm thưởng đang có.

Với giao diện đồ họa sống động, âm thanh đa phương tiện với lời dẫn chương trình thật của MC Đinh Tiến Dũng, đây thật sự là một phần mềm vừa chơi vừa học thú vị nhất là cho các bạn học sinh – sinh viên dùng cho ôn luyện kiến thức.

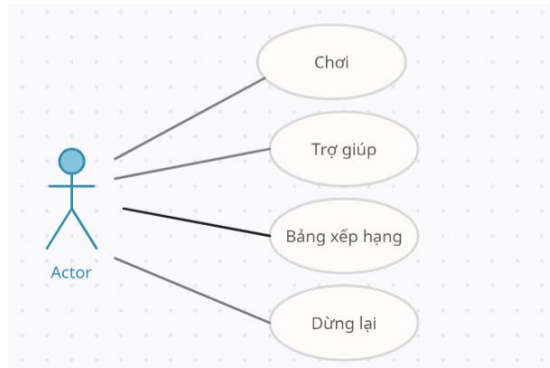
2. Cách chơi

Trò chơi có tất cả 15 câu hỏi (các câu hỏi được chọn lựa ngẫu nhiên trong bộ câu hỏi của game Ai là triệu phú). Chúng được đưa ra với thứ tự từ dễ đến khó. Trong quá trình chơi trò chơi ai là triệu phú, bạn sẽ có 3 sự trợ giúp bao gồm:

- Trợ giúp 50/50: trò chơi sẽ loại bỏ đi 2 đáp án không chính xác cho bạn.
- Trợ giúp gọi điện thoại cho người thân: sẽ đưa ra một đáp án theo hàm ngẫu nhiên
- Trợ giúp hỏi ý kiến khán giả: hiện lên biểu đồ tỉ lệ đáp án.
- Trợ giúp đổi câu hỏi: cho phép người chơi đổi câu hỏi

Bạn có 30s suy nghĩ và trả lời, nhưng hãy cố gắng trả lời thật nhanh và chính xác. Khi người chơi trả lời sai câu hỏi nào đó thì chương trình sẽ công bố đáp án đúng, giải thưởng bạn mang về và dừng lại. Khi bạn trả lời đúng, chương trình tự động chuyển câu tiếp theo và thông báo câu trả lời đúng. Mỗi câu hỏi sẽ có 4 đáp án, bạn cần chọn đúng đáp án chính xác cho chúng. Sử dụng chuột  để thực hiện các thao tác.

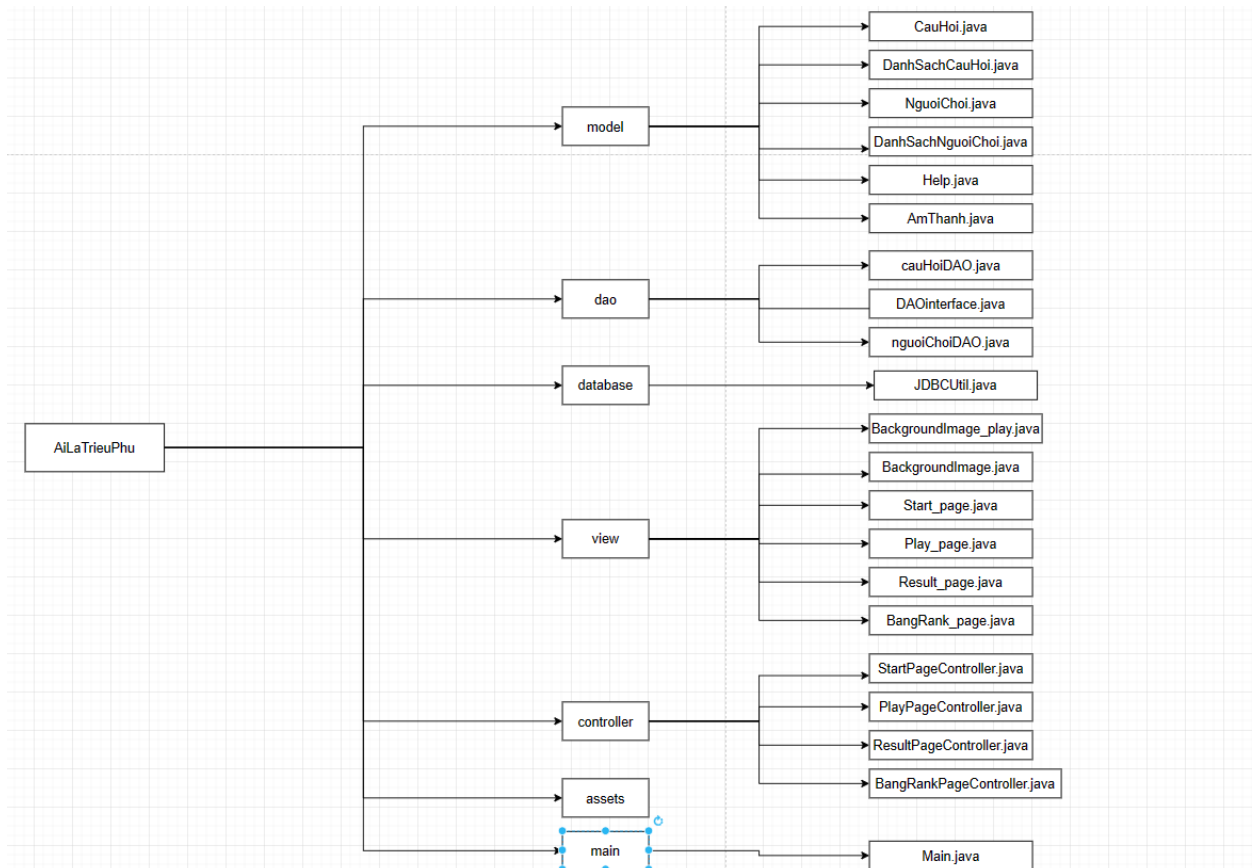
3. Biểu đồ Use case



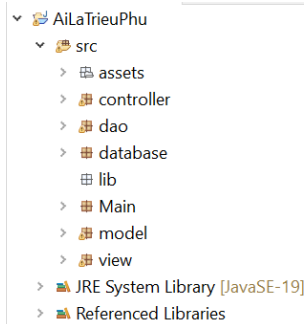
- Giải thích Use Case:
 - Chơi: Bắt đầu trò chơi
 - Trợ giúp: có 4 trợ giúp dành cho người chơi
 - Bảng xếp hạng: bảng xếp hạng
 - Dừng lại: dừng lại và thoát khỏi trò chơi

II. Xây dựng chương trình

1. Sơ đồ chung



2. Giới thiệu và nêu chức năng của package

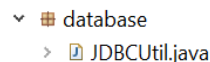


Với đề tài “Game Ai là triệu phú”, nhóm chúng em sử dụng mô hình MVC (Model-View-Controller)

- Model: Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller.
- Database: Chứa phương thức liên kết với cơ sở dữ liệu.
- Dao (Data Access Object): Gồm các hàm để tương tác với cơ sở dữ liệu của từng đối tượng.
- Controller: xử lý các yêu cầu người dùng đưa đến thông qua view.
- View: Chứa các class giao diện của ứng dụng.
- Assets: chứa các file ảnh, file âm thanh.

3. Giới thiệu và các tương tác trong giao diện

3.1. Kết nối database



- Hàm kết nối cơ sở dữ liệu: sử dụng hệ quản trị cơ sở dữ liệu MySQL

```
public class JDBCUtil {  
    public static Connection getConnection1() {  
        Connection connection = null;  
        try {  
            // Đăng kí MySQL  
            Driver driver = new Driver();  
            DriverManager.registerDriver(driver);  
            String url = "jdbc:mysql://localhost:3306/ailatrieuphu";  
            String username = "root";  
            String password = "";  
  
            // Tạo kết nối  
            connection = DriverManager.getConnection(url, username, password);  
        } catch (SQLException e) {  
            e.printStackTrace();  
        }  
        return connection;  
    }  
}
```

- Hàm đóng kết nối cơ sở dữ liệu:

```
public static void closeConnection(Connection connection) {
    try {
        if (connection != null) {
            connection.close();
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

3.2. Trang bắt đầu chơi

✚ Giao diện:



a) Xây dựng giao diện

```
view
> BackgroundImage_play.java
> BackgroundImage.java
> BangRank_page.java
> Play_page.java
> Result_page.java
> Start_page.java
```

- File Start_page.java: thiết kế ra 1JFrame trong đó có background được xây dựng bởi file BackgroundImage.java, 1JButton “Sẵn sàng” và 1JPanel để nhập tên.



b) Xây dựng tương tác

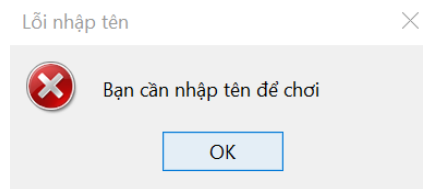
```

  controller
  > BangRankPageController.java
  > PlayPageController.java
  > ResultPageController.java
  > StartPageController.java

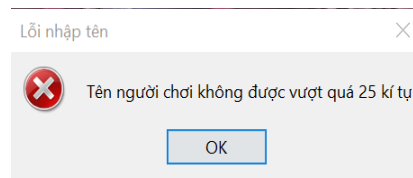
public void actionPerformed(ActionEvent e) {
    if (e.getSource().equals(start_page.getButton_start())) {
        String userName = start_page.getUserName().getText().trim();
        if (userName.equals("")) {
            JOptionPane.showMessageDialog(start_page,
                "Bạn cần nhập tên để chơi",
                "Lỗi nhập tên",
                JOptionPane.ERROR_MESSAGE);
        }
        else if (userName.length() > 25)
        {
            JOptionPane.showMessageDialog(start_page,
                "Tên người chơi không được vượt quá 25 kí tự",
                "Lỗi nhập tên",
                JOptionPane.ERROR_MESSAGE);
        }
        else
        {
            DanhSachCauHoi danhSachCauHoi = new DanhSachCauHoi();
            new Play_page(userName);
            AmThanh.clip_bat_dau.stop();
            start_page.setVisible(false);
        }
    }
}

```

- Khi người chơi không nhập tên, thì chương trình hiện lên thông báo lỗi nhập tên:



- Khi người chơi nhập tên quá 25 ký tự, thì chương trình hiện lên thông báo lỗi nhập tên:



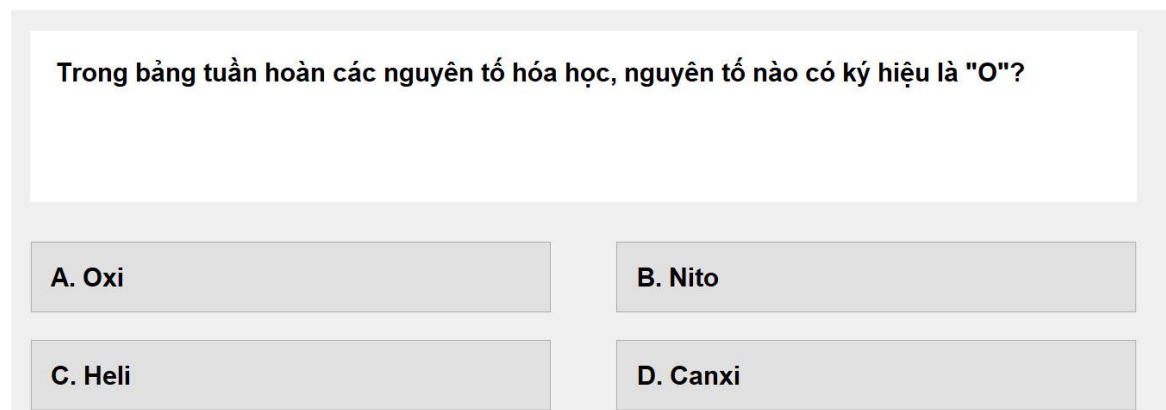
3.3. Trang chơi game

🎮 Giao diện:



3.3.1. Chức năng chọn đáp án

🎮 Giao diện:



a) Xây dựng giao diện

- view
 - > BackgroundImage_play.java
 - > BackgroundImage.java
 - > BangRank_page.java
 - > **Play_page.java**
 - > Result_page.java
 - > Start_page.java


```
// Nội dung + đáp án
JPanel jPanel_center = new JPanel(new GridLayout(2, 1, 30, 30));

JPanel jPanel_center_top = new JPanel(new GridLayout(1, 3, 10, 10));

imageURL = getClass().getClassLoader().getResource("assets/anhnen.jpg");
BackgroundImage_play background_center_top = new BackgroundImage_play(imageURL);

JPanel jPanel_empty = new JPanel();

jPanel_center_top.add(background_center_top);

JPanel jPanel_center_bottom = new JPanel(new GridLayout(2, 1, 30, 30));
cauHoi = new JTextArea("Câu hỏi");
cauHoi.setFont(font);
cauHoi.setLineWrap(true);
cauHoi.setWrapStyleWord(true);
cauHoi.setEditable(false);
JScrollPane jScrollPane = new JScrollPane(cauHoi);
cauHoi.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));

JPanel jPanel_dapAn = new JPanel(new GridLayout(2, 2, 50, 20));

jButtons_dapAns = new JButton[4];
for (int i = 0; i < 4; i++) {
    jButtons_dapAns[i] = new JButton();
    jButtons_dapAns[i].setFont(font);
    jButtons_dapAns[i].setHorizontalAlignment(JButton.LEFT);
    jButtons_dapAns[i].addActionListener(playPageController);
    jButtons_dapAns[i].setVisible(false);
    jPanel_dapAn.add(jButtons_dapAns[i]);
}

jPanel_center_bottom.add(cauHoi);
jPanel_center_bottom.add(jPanel_dapAn);

jPanel_center.add(jPanel_center_top);
jPanel_center.add(jPanel_center_bottom);
jPanel_center.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));
```

b) Xây dựng tương tác

- ▼ 📁 controller
 - > 📄 BangRankPageController.java
 - > 📄 PlayPageController.java
 - > 📄 ResultPageController.java
 - > 📄 StartPageController.java

```
for (int i = 0; i < 4; i++)
{
    final int chi_so = i;
    if (e.getSource().equals(play_page.getjButtons_dapAns(i))) // khi người chơi bấm vào 1 trong đáp án
    {
        AmThanh.clip_dem_nguoc.stop();
        play_page.getTimer_thoiGianChay().stop();

        Timer timer = new Timer(500, new ActionListener() {
            int count = 0;
            @Override
            public void actionPerformed(ActionEvent e) {
                if (count == 0)
                {
                    play_page.setDisableDapAn();
                    play_page.setDapAnChon(play_page.getjButtons_dapAns(chi_so));
                }
                count++;
                if (count == 6)
                {
                    if (play_page.getNguoiChoi().getCauHoiHienTai().getDapAnDung().equals(CauHoi.nhanDapAn(chi_so)))
                    {
                        play_page.setBackGroundDapAnDung(play_page.getjButtons_dapAns(chi_so));
                        AmThanh.clip_vo_tay.setMicrosecondPosition(0);
                        AmThanh.clip_vo_tay.start();
                    }
                    else
                    {
                        final int csoDung = play_page.getNguoiChoi().getCauHoiHienTai().getChiSoDapAnDung();
                        play_page.setBackGroundDapAnDung(play_page.getjButtons_dapAns(csoDung));
                        play_page.setBackGroundDapAnSai(play_page.getjButtons_dapAns(chi_so));
                        AmThanh.clip_tra_loi_sai.start();
                        play_page.updateBangRank();
                        play_page.updateBangRank();
                    }
                }
            }
        });
    }
}
```

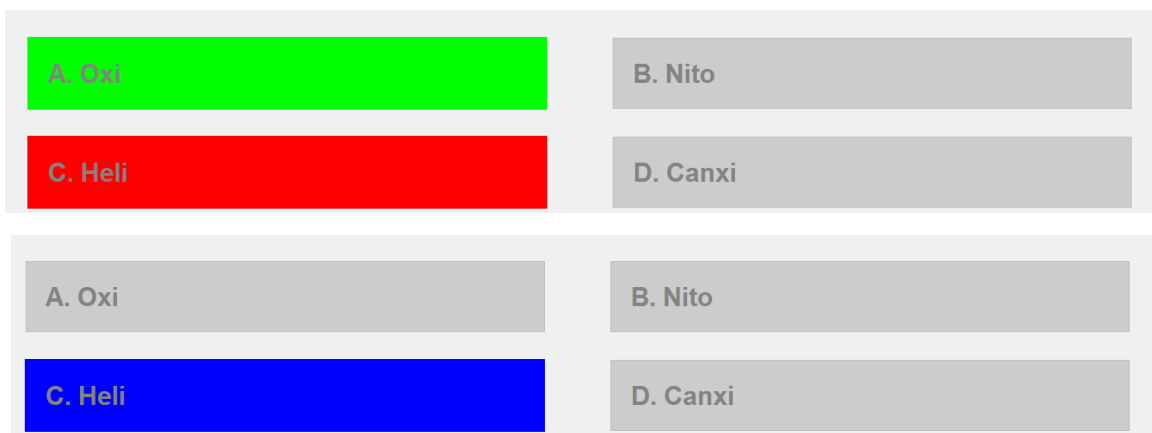
```

else if (count == 16)
{
    if (play_page.getNguoiChoi().getCauHoiHienTai().getDapAnDung().equals(CauHoi.nhanDapAn(chi_so)))
    {
        play_page.setMauChuChoCauDung(play_page.getNguoiChoi().getLevel());
        play_page.getNguoiChoi().increaseLevel();
        AmThanh.clip_vo_tay.stop();
        play_page.reset();
        play_page.xuli(play_page.getNguoiChoi().getLevel());
        ((Timer) e.getSource()).stop();
    }
    else
    {
        AmThanh.clip_tra_loi_sai.stop();
        play_page.setVisible(false);
        new Result_page(play_page.getNguoiChoi());
    }
    ((Timer) e.getSource()).stop();
}
}
});
timer.start();
}
}
}

```

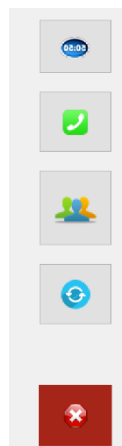
- Khi người chơi chọn 1 trong 4 đáp án, thì 3 đáp án còn lại sẽ bị vô hiệu hóa (không chọn được) và đáp án được chọn sẽ chuyển thành màu xanh dương. Và sau đó sẽ hiện lên đáp án đúng.

- *Giao diện khi chọn đáp án*



3.3.2. Chức năng trợ giúp

 Giao diện:



a) Xây dựng giao diện

- view
 - BackgroundImage_play.java
 - BackgroundImage.java
 - BangRank_page.java
 - Play_page.java**
 - Result_page.java
 - Start_page.java

```
// Các quyền trợ giúp + thời gian
JPanel jPanel_left = new JPanel(new GridBagLayout());
GridBagConstraints c = new GridBagConstraints();

c.fill = GridBagConstraints.BOTH;
c.insets = new Insets(10, 10, 10, 10);

c.gridx = 0; c.gridy = 0; c.weighty = 1;
jPanel_left.add(jLabel_time, c);

c.gridx = 0; c.gridy = 1; c.gridwidth = 0; c.gridheight = 1;
jPanel_left.add(new JLabel(), c);

c.gridx = 0; c.gridy = 2; c.gridwidth = 1; c.gridheight = 1;
jPanel_left.add(jButton_50_50, c);

c.gridx = 0; c.gridy = 3;
jPanel_left.add(jButton_phone, c);

c.gridx = 0; c.gridy = 4;
jPanel_left.add(jButton_khan_gia, c);

c.gridx = 0; c.gridy = 5;
jPanel_left.add(jButton_doi_cau_hoi, c);

c.gridx = 0; c.gridy = 6; c.gridwidth = 0; c.gridheight = 1;
jPanel_left.add(new JLabel(), c);

c.gridx = 0; c.gridy = 8;
jPanel_left.add(jButton_stop, c);

jPanel_left.setBorder(BorderFactory.createEmptyBorder(20, 20, 20, 20));
```

b) Xây dựng tương tác

- controller
 - BangRankPageController.java
 - PlayPageController.java**
 - ResultPageController.java
 - StartPageController.java



Trợ giúp 50/50: khi người chơi click vào đi 2 đáp án không đúng.



thì chương trình sẽ loại

```


if (e.getSource().equals(play_page.getjButton_50_50())) // xử lý trợ giúp 50/50
{
    Thread thread = new Thread(new Runnable() {
        @Override
        public void run() {
            try {
                AmThanh.clip_dem_nguoc.stop();
                play_page.getTimer_thoiGianChay().stop();
                Thread.sleep(3000);
                Random random = new Random();
                int ans1 = random.nextInt(4);
                int ans2 = random.nextInt(4);
                int ansDung = play_page.getNguoiChoi().getCauHoiHienTai().getChiSoDapAnDung();
                while(ans1 == ans2 || ans2 == ansDung) ans2 = random.nextInt(4);
                while(ans1 == ans2 || ans1 == ansDung) ans1 = random.nextInt(4);
                play_page.getNguoiChoi().getCauHoiHienTai().setEnabled(ans1, false);
                play_page.getNguoiChoi().getCauHoiHienTai().setEnabled(ans2, false);
                for (int i = 0; i < 4; i++)
                    if (!play_page.getNguoiChoi().getCauHoiHienTai().getEnable(i))
                    {
                        play_page.getjButtons_dapAns(i).setEnabled(false);
                    }
                play_page.getjButton_50_50().setEnabled(false);

                play_page.getTimer_thoiGianChay().restart();
                AmThanh.clip_dem_nguoc.start();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    });
    thread.run();
}

```

- Chọn hai đáp án này bằng cách lấy ngẫu nhiên hai trong ba đáp án sai bằng hàm random (). Chương trình sẽ thực hiện ẩn đi 2 đáp án sai được chọn.
- *Giao diện khi sử dụng trợ giúp 50/50*



- ✚ **Trợ giúp gọi điện thoại cho người thân:** Khi người chơi click vào  thì chương trình sẽ hiện lên tiếng chuông điện thoại, hết tiếng chuông hiện lên 1 hộp thoại có đáp án mà người thân đưa ra.

```

else if (e.getSource().equals(play_page.getjButton_phone())) // xử lý sự trợ giúp gọi điện thoại cho người thân
{
    Thread thread = new Thread(new Runnable() {
        @Override
        public void run() {
            try {
                AmThanh.clip_goi_dien.start();
                play_page.getTimer_thoiGianChay().stop();
                AmThanh.clip_dem_nguoc.stop();
                Thread.sleep(5000);
                AmThanh.clip_goi_dien.stop();
                play_page.getjButton_phone().setEnabled(false);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    });
    thread.run();

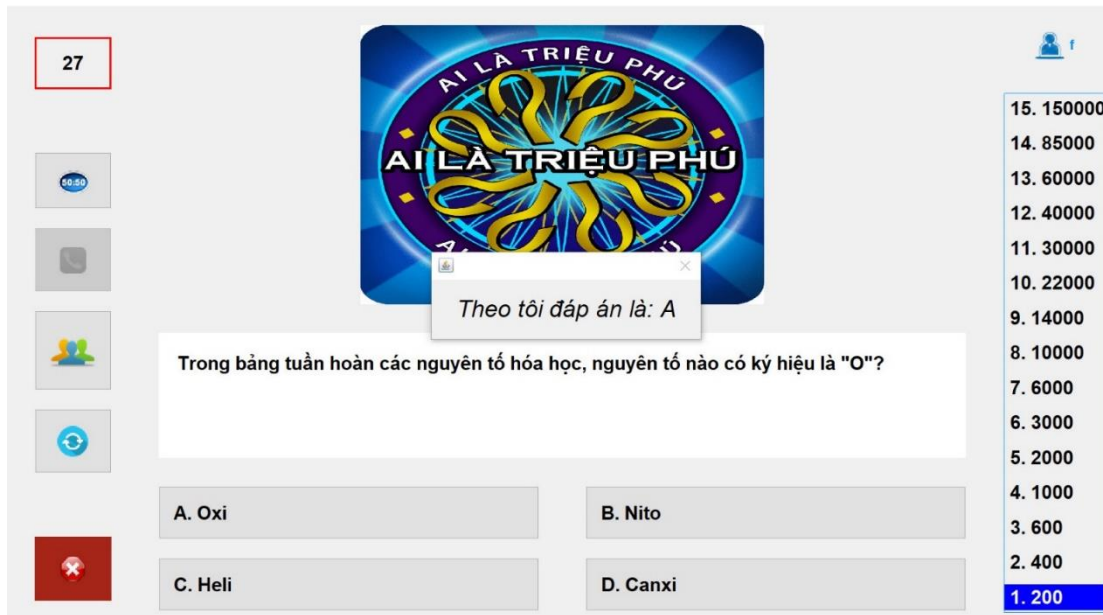
    String dapAnNguoiThan;
    if (play_page.getNguoiChoi().getCauHoiHienTai().getLevel() < 10) {
        dapAnNguoiThan = play_page.getNguoiChoi().getCauHoiHienTai().getDapAnDung();
    }
    else
    {
        int stt;
        do {
            Random random = new Random();
            stt = random.nextInt(4);
        } while (!play_page.getNguoiChoi().getCauHoiHienTai().getEnable(stt));
        dapAnNguoiThan = play_page.getNguoiChoi().getCauHoiHienTai().nhanDapAn(stt);
    }
    // dialog hiển thị đáp án
    JDialog dialog = new JDialog();
    dialog.setName("Gọi điện thoại cho homie");
    dialog.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
    dialog.setLayout(new BorderLayout());
    JLabel jLabel = new JLabel("Theo tôi đáp án là: " + dapAnNguoiThan, SwingConstants.CENTER);
    jLabel.setHorizontalAlignment(JLabel.CENTER);
    jLabel.setFont(new Font("Arial", Font.ITALIC, 25));
    dialog.add(jLabel, BorderLayout.CENTER);
    dialog.setSize(300, 100);
    dialog.setLocationRelativeTo(null);
    dialog.setVisible(true);


    Timer timer = new Timer(4000, new ActionListener() {
        int count = 1;
        @Override
        public void actionPerformed(ActionEvent e) {
            count--;
            if (count == 0)
            {
                dialog.dispose();
                play_page.getTimer_thoiGianChay().restart();
                ((Timer) e.getSource()).stop();
            }
        }
    });
    timer.start();
}

```

- Nếu như level < 10 thì trả ra đáp án đúng, còn không thì sẽ dùng hàm random () để chọn 1 trong 4 đáp án.

- Giao diện khi dùng quyền gọi điện thoại cho người thân



- + **Trợ giúp hỏi ý kiến khán giả:** Khi người chơi click vào  chương trình sẽ hiện lên 1 biểu đồ tỷ lệ đáp án. Người chơi sẽ dựa vào biểu đồ đáp án đấy và chọn 1 đáp án trong 4 đáp án trên.

```
// xử lý hỏi ý kiến khán giả
else if (e.getSource().equals(play_page.getJButton_khan_gia()))
{
    AmThanh.clip_dem_nguoc.stop();
    play_page.getTimer_thoiGianChay().stop();
    try {
        Thread.sleep(3000);
        play_page.getJButton_khan_gia().setEnabled(false);
    } catch (InterruptedException e1) {
        e1.printStackTrace();
    }
    int tile[] = new int[4];
    CauHoi cauHoi = play_page.getNguoiChoi().getCauHoiHienTai();
    int ansDung = cauHoi.getChiSoDapAnDung();
    int sum = 0;
    Random random = new Random();
    if (cauHoi.getLevel() <= 5) tile[ansDung] = 50 + random.nextInt(50);
    else if (cauHoi.getLevel() <= 10) tile[ansDung] = 35 + random.nextInt(70);
    else tile[ansDung] = 25 + random.nextInt(75);
    sum = tile[ansDung];
    for (int j = 0; j < 4; j++)
        if (!cauHoi.getEnable(j))
        {
            tile[j] = 0;
        }
        else if (j != ansDung) {
            tile[j] = random.nextInt(100 - sum);
            sum += tile[j];
        }
    if (sum != 100) {
        for (int j = 0; j < 4; j++)
            if (cauHoi.getEnable(j))
            {
                tile[j] += (100 - sum);
                break;
            }
    }
}
```

```

// Tạo biểu đồ
DefaultCategoryDataset dataset = new DefaultCategoryDataset();
for (int i = 0; i < 4; i++)
    dataset.addValue(tiLe[i], "Phần trăm", CauHoi.nhanDapAn(i));

JFreeChart jFreeChart = ChartFactory.createBarChart(
    "Ý kiến của khán giả trong trường quay", // Tên biểu đồ
    "Đáp án", // Tên trục x
    "Phần trăm", // Tên trục y
    dataset, // dữ liệu
    PlotOrientation.VERTICAL, // hướng của biểu đồ
    true, // hiển thị chú thích
    true, //
    false);

// Thiết lập kiểu hiển thị và vị trí của nhãn số trên các cột
CategoryPlot plot = jFreeChart.getCategoryPlot();
BarRenderer renderer = (BarRenderer) plot.getRenderer();
renderer.setDefaultItemLabelGenerator(new StandardCategoryItemLabelGenerator());
renderer.setDefaultItemLabelsVisible(true);
renderer.setDefaultPositiveItemLabelPosition(new ItemLabelPosition(
    ItemLabelAnchor.OUTSIDE12, TextAnchor.BOTTOM_CENTER, TextAnchor.CENTER, 0
));
renderer.setDefaultItemLabelGenerator(new StandardCategoryItemLabelGenerator("{2}%", new DecimalFormat("0")));

ValueAxis rangeAxis = plot.getRangeAxis();
rangeAxis.setRange(0, 100);

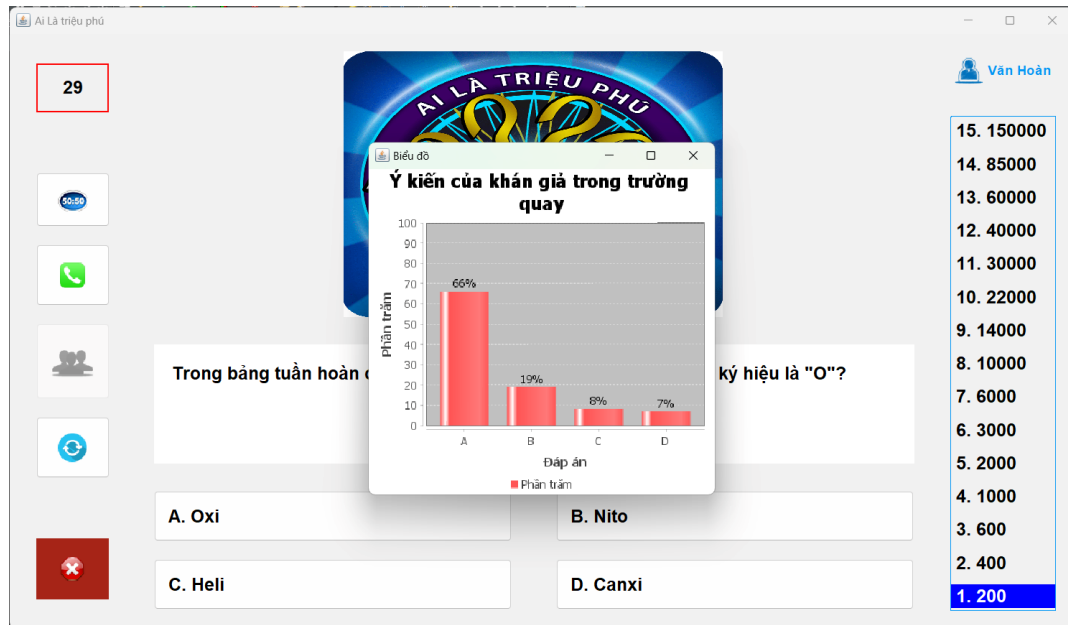
ChartFrame frame = new ChartFrame("Biểu đồ", jFreeChart);
frame.setSize(400, 400);
frame.setLocationRelativeTo(null);
frame.setVisible(true);


Timer timer = new Timer(4000, new ActionListener() {
    int count = 1;
    @Override
    public void actionPerformed(ActionEvent e) {
        count--;
        if (count == 0)
        {
            frame.setVisible(false);
            AmThanh.clip_dem_nguoc.start();
            play_page.getTimer_thoiGianChay().restart();
            ((Timer) e.getSource()).stop();
        }
    }
});
timer.start();

```

- Cho 1 biểu đồ gồm 4 phần tử ứng với tỉ lệ 4 đáp án. Nếu level ≤ 5 thì tỉ lệ của đáp án đúng là: $50 + \text{random.nextInt}(50)$, nếu level ≤ 10 thì tỉ lệ của đáp án đúng là: $30 + \text{random.nextInt}(70)$ và trường hợp còn lại level ≤ 15 thì tỉ lệ đáp án đúng là: $25 + \text{random.nextInt}(75)$.

- *Giao diện khi dùng quyền trợ giúp hỏi ý kiến khán giả:*

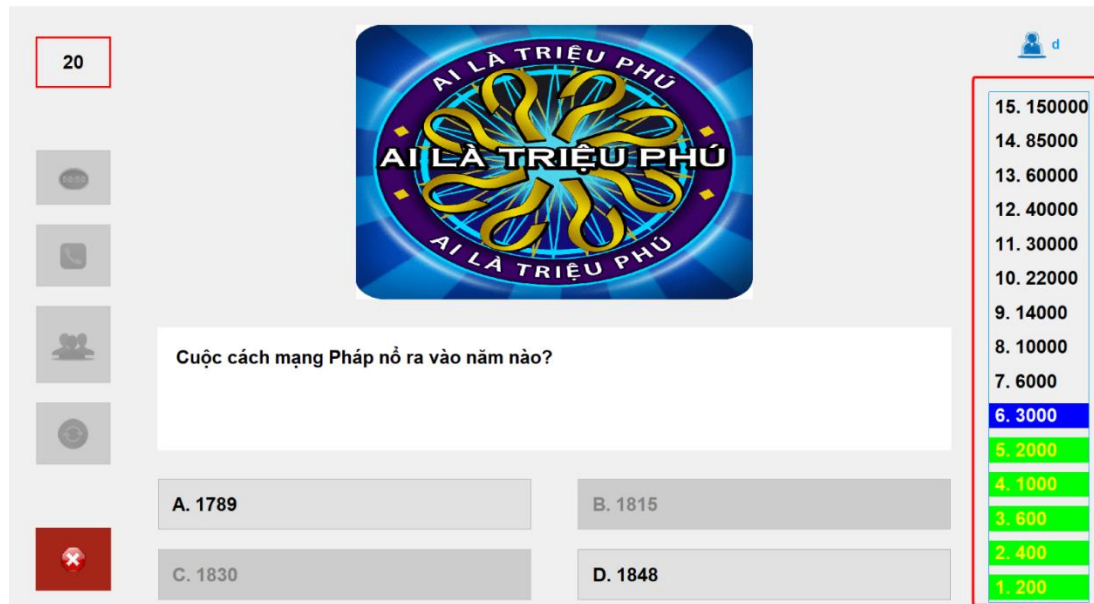


Trợ giúp đổi câu hỏi: Khi người chơi click vào  chương trình sẽ đổi câu hỏi khác cùng level.

```
else if (e.getSource().equals(play_page.getjButton_doi_cau_hoi())) // xử lý đổi câu hỏi
{
    play_page.getTimer_thoiGianChay().stop();
    AmThanh.clip_dem_nguoc.stop();
    int level = play_page.getNguoiChoi().getLevel();
    play_page.getNguoiChoi().setCauHoi(level, DanhSachCauHoi.sinhThemCauHoi(play_page.getNguoiChoi().getCauHoiHienTai(), level));
    play_page.reset();
    play_page.xuli(level);
    play_page.getjButton_doi_cau_hoi().setEnabled(false);
}
else if (e.getSource().equals(play_page.getjButton_stop()))
{
    JPanel panel = new JPanel();
    JLabel label = new JLabel("Bạn có chắc chắn muốn dừng lại?");
    label.setFont(new Font("Arial", Font.ITALIC, 15));
    panel.add(label);
    int result = JOptionPane.showOptionDialog(null, panel, "Xác nhận dừng lại",
        JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE,
        null, new String[] { "Đồng ý", "Hủy bỏ" }, null);
    if (result == JOptionPane.OK_OPTION) {
        AmThanh.clip_dem_nguoc.stop();
        play_page.getTimer_thoiGianChay().stop();
        play_page.setVisible(false);
        play_page.updateBangRank();
        new Result_page(play_page.getNguoiChoi());
    } else {
        JOptionPane.getRootFrame().dispose();
    }
}
}
```

3.3.3. Mức độ câu hỏi

✚ Giao diện:



a) Xây dựng giao diện

- view
 - BackgroundImage_play.java
 - BackgroundImage.java
 - BangRank_page.java
 - Play_page.java**
 - Result_page.java
 - Start_page.java

```
JPanel jPanel_right = new JPanel(new BorderLayout(10, 30));
JPanel jPanel_user = new JPanel(new FlowLayout());
JPanel jPanel_levels = new JPanel(new GridLayout(15, 1, 10, 10));

jLabel_levels = new JLabel[16];
for (int i = 15; i > 0; i--)
{
    jLabel_levels[i] = new JLabel(" " + i + ". " + NguoiChoi.levels[i]);
    jLabel_levels[i].setFont(font);
    jLabel_levels[i].setMinimumSize(new Dimension(200, 500));
    jPanel_levels.add(jLabel_levels[i]);
}
jPanel_levels.setBorder(BorderFactory.createLineBorder(new Color(3, 151, 244), 1, true));
```

b) Xây dựng tương tác với giao diện

- Khi người chơi trả lời đúng từng câu hỏi thì level trong bảng cũng tăng lên. Những câu hỏi nào đã trả lời đúng ứng với số level thì level đó được chuyển sang màu *GREEN*.

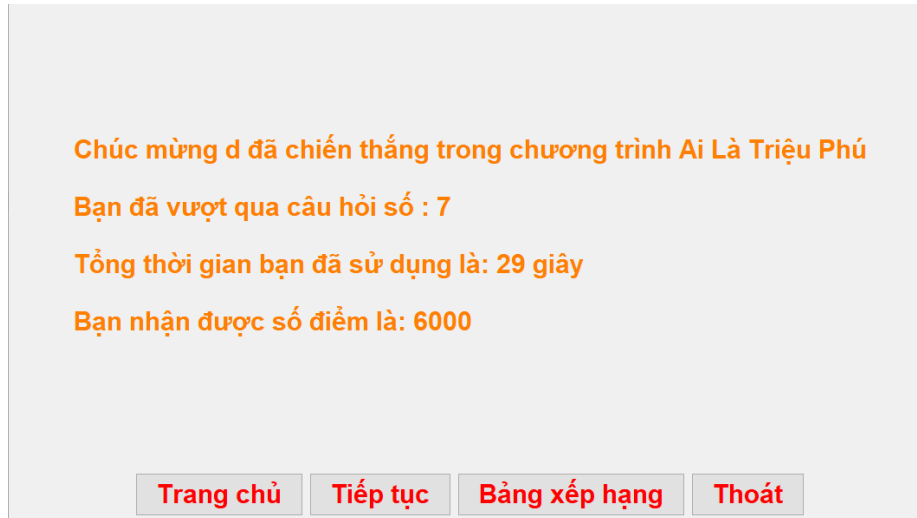
```

public void themBackGroundChoLevel(int i) {
    jLabel_levels[i].setForeground(Color.WHITE);
    jLabel_levels[i].setBackground(Color.BLUE);
    jLabel_levels[i].setOpaque(true);
}

```

3.4. Trang kết quả

Giao diện



a) Xây dựng giao diện

```

view
├── BackgroundImage_play.java
├── BackgroundImage.java
├── BangRank_page.java
├── Play_page.java
├── Result_page.java
└── Start_page.java

```

```

int level = nguoiChoi.getLevel();

AmThanh.clip_giai_thuong.start();

ResultPageController resultPageController = new ResultPageController(this);

this.setLayout(new BorderLayout());
Font font = new Font("Arial", Font.BOLD, 20);

Border border = BorderFactory.createEmptyBorder(100, 50, 100, 50);

Color color_result = new Color(255, 127, 0);

JLabel jLabel_chucMung = new JLabel("Chúc mừng " + nguoiChoi.getUserName() + " đã chiến thắng trong chương trình Ai Là Triệu Phú");
jLabel_chucMung.setFont(font);
jLabel_chucMung.setForeground(color_result);

JLabel jLabel_soCau = new JLabel("Bạn đã vượt qua câu hỏi số : " + String.valueOf(level - 1));
jLabel_soCau.setFont(font);
jLabel_soCau.setForeground(color_result);

JLabel jLabel_tienThuong = new JLabel("Bạn nhận được số điểm là : " + String.valueOf(NguoiChoi.Levels[level-1]));
jLabel_tienThuong.setFont(font);
jLabel_tienThuong.setForeground(color_result);

JLabel jLabel_thoiGian = new JLabel("Tổng thời gian bạn đã sử dụng là : " + nguoiChoi.getTimes() + " giây");
jLabel_thoiGian.setFont(font);
jLabel_thoiGian.setForeground(color_result);

JButton_trangChu = new JButton("Trang chủ");
JButton_trangChu.setFont(font);
JButton_trangChu.setForeground(Color.RED);
JButton_trangChu.addActionListener(resultPageController);

```

```

jButton_tiepTuc = new JButton("Tiếp tục");
jButton_tiepTuc.setFont(font);
jButton_tiepTuc.setForeground(Color.RED);
jButton_tiepTuc.addActionListener(resultPageController);

jButton_thoat = new JButton("Thoát");
jButton_thoat.setFont(font);
jButton_thoat.setForeground(Color.RED);
jButton_thoat.addActionListener(resultPageController);

jButton_banhRank = new JButton("Bảng xếp hạng");
jButton_banhRank.setFont(font);
jButton_banhRank.setForeground(Color.RED);
jButton_banhRank.addActionListener(resultPageController);

JPanel jPanel_group_buttons = new JPanel(new FlowLayout());
jPanel_group_buttons.add(jButton_trangChu);
jPanel_group_buttons.add(jButton_tiepTuc);
jPanel_group_buttons.add(jButton_banhRank);
jPanel_group_buttons.add(jButton_thoat);

JPanel jPanel_khungKetQua = new JPanel(new GridLayout(4, 1, 20, 20));
jPanel_khungKetQua.setBorder(border);

jPanel_khungKetQua.add(jLabel_chucMung);
jPanel_khungKetQua.add(jLabel_soCau);
jPanel_khungKetQua.add(jLabel_thoiGian);
jPanel_khungKetQua.add(jLabel_tienThuong);

this.add(jPanel_khungKetQua, BorderLayout.CENTER);
this.add(jPanel_group_buttons, BorderLayout.SOUTH);
this.pack();

```

b) Xây dựng tương tác giao diện

- ▼ controller
 - > BangRankPageController.java
 - > PlayPageController.java
 - > **ResultPageController.java**
 - > StartPageController.java


```

public class ResultPageController implements ActionListener{
    private Result_page result_page;
    public ResultPageController(Result_page result_page) {
        this.result_page = result_page;
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource().equals(result_page.getJButton_thoat()))
        {
            AmThanh.clip_giai_thuong.stop();
            result_page.close();
            System.exit(0);
        }
        else if (e.getSource().equals(result_page.getJButton_tiepTuc())) {
            AmThanh.clip_giai_thuong.stop();
            result_page.close();
            new Play_page(result_page.getNguoiChoi().getUserName());
        }
        else if (e.getSource().equals(result_page.getJButton_banhRank())) {
            AmThanh.clip_giai_thuong.stop();
            result_page.close();
            new BangRank_page(result_page.getNguoiChoi());
        }
        else if (e.getSource().equals(result_page.getJButton_trangChu())) {
            AmThanh.clip_giai_thuong.stop();
            result_page.close();
            new Start_page();
        }
    }
}

```

- Trang kết quả đưa ra 4 nội dung: tên người chơi, số câu vượt qua, tổng thời gian chơi và số điểm đạt được. Phía dưới có 4 nút:
 - Trang chủ: đưa về trang bắt đầu chơi
 - Tiếp tục: đưa về trang chơi với tên người chơi giữ nguyên
 - Bảng xếp hạng: đưa đến trang bảng xếp hạng
 - Thoát: thoát khỏi chương trình

3.5. Trang bảng xếp hạng

 Giao diện

Bảng xếp hạng				
Thứ hạng	Tên người chơi	Số câu lớn nhất	Tổng thời gian	Thời gian thi
1	Đạt Thường Tín	11	47	2023-04-25 17:...
2	Thắng cá chép	11	83	2023-04-27 10:...
3	Việt Hoàng nỗ	8	57	2023-04-26 23:...
4	d	7	29	2023-05-02 15:...
5	hiep	7	40	2023-04-26 13:...
6	đạt	7	72	2023-04-27 11:...
7	Hoang ha	6	59	2023-05-01 09:...
8	tiến đạt	5	53	2023-04-27 11:...
9	Thành Đạt	4	5	2023-04-25 17:...
10	Việt Hoàng ngu	3	9	2023-04-25 17:...
11	aaa	3	18	2023-04-26 23:...
12	hoang	3	34	2023-04-27 12:...
13	a	2	0	2023-04-26 23:...
14	vh19	2	9	2023-04-26 13:...

Trang chủ
Tiếp tục
Thoát

a) Xây dựng giao diện

view

- > BackgroundImage_play.java
- > BackgroundImage.java
- > **BangRank_page.java**
- > Play_page.java
- > Result_page.java
- > Start_page.java

```

// Nút quay về trang chủ
JPanel jPanel_top = new JPanel(new FlowLayout());
JButton jButton_trangChu = new JButton("Trang chủ");
jButton_trangChu.setFont(new Font(Font.SANS_SERIF, Font.BOLD, 15));
jButton_trangChu.setForeground(Color.RED);
jButton_trangChu.addActionListener(bangRankPageController);

// Nút chơi tiếp
JButton jButton_choiTiep = new JButton("Tiếp tục");
jButton_choiTiep.setFont(new Font(Font.SANS_SERIF, Font.BOLD, 15));
jButton_choiTiep.setForeground(Color.RED);
jButton_choiTiep.addActionListener(bangRankPageController);

// Nút thoát game
JButton jButton_thoat = new JButton("Thoát");
jButton_thoat.setFont(new Font(Font.SANS_SERIF, Font.BOLD, 15));
jButton_thoat.setForeground(Color.RED);
jButton_thoat.addActionListener(bangRankPageController);

jPanel_top.add(jButton_trangChu);
jPanel_top.add(jButton_choiTiep);
jPanel_top.add(jButton_thoat);

JPanel jPanel_center = new JPanel(new BorderLayout());

JLabel jLabel_tenBang = new JLabel("Bảng xếp hạng");
jLabel_tenBang.setFont(new Font(Font.SANS_SERIF, Font.BOLD, 18));
jLabel_tenBang.setHorizontalAlignment(JLabel.CENTER);
jLabel_tenBang.setForeground(Color.GREEN);

Object[] columnNames = {"Thứ hạng", "Tên người chơi", "Số câu lớn nhất",
    "Tổng thời gian", "Thời gian thi"};

```

```

NguoiChoi danhSachNguoiChoi[] = DanhSachNguoiChoi.getDanhSachNguoiChoi();
Object[][] data = new Object[danhSachNguoiChoi.getCount()][5];
for (int i = 0; i < danhSachNguoiChoi.getCount(); i++)
{
    data[i] = new Object[] {danhSachNguoiChoi[i].getSTT(),
        danhSachNguoiChoi[i].getUserName(),
        danhSachNguoiChoi[i].getLevel(),
        danhSachNguoiChoi[i].getTimes(),
        danhSachNguoiChoi[i].getDateTime()};
}

DefaultTableModel model = new DefaultTableModel(data, columnNames)
{
    @Override
    public boolean isCellEditable(int row, int column) {
        // Không cho phép chỉnh sửa tất cả các ô
        return false;
    }
};

JTable table = new JTable(model);

Font font_table_content = new Font(Font.SANS_SERIF, Font.PLAIN, 15);
table.setRowHeight(25);
table.setFont(font_table_content);
table.setPreferredScrollableViewportSize(new Dimension(500, 25 * 50));

JTableHeader header = table.getTableHeader();
header.setFont(new Font("Arial", Font.BOLD, 15));

JScrollPane scrollPane = new JScrollPane(table);
scrollPane.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
jPanel_center.add(jLabel_tenBang, BorderLayout.NORTH);
jPanel_center.add(scrollPane, BorderLayout.CENTER);

```

b) Xây dựng tương tác với giao diện

```

controller
  > BangRankPageController.java
  > PlayPageController.java
  > ResultPageController.java
  > StartPageController.java

public class BangRankPageController implements ActionListener{
    private BangRank_page bangRank;
    public BangRankPageController(BangRank_page bangRank) {
        this.bangRank = bangRank;
    }
    @Override
    public void actionPerformed(ActionEvent e) {
        // TODO Auto-generated method stub
        String src = e.getActionCommand();
        if (src.equals("Trang chủ")) {
            bangRank.setVisible(false);
            new Start_page();
        }
        else if (src.equals("Tiếp tục")) {
            bangRank.setVisible(false);
            new Play_page(bangRank.getNguoiChoi().getUserName());
        }
        else if (src.equals("Thoát")){
            System.exit(0);
        }
    }
}

```

- Bảng xếp hạng gồm 1 bảng lưu thông tin kết quả của người chơi. Bảng bao gồm 5 cột ứng với thứ hạng, tên người chơi, số câu đạt được lớn nhất, tổng thời gian, thời gian thi. Phía dưới trang bảng xếp hạng có 3 nút:
- Trang chủ: quay về trang bắt đầu chơi
 - Tiếp tục: quay về trang chơi với tên người chơi giữ nguyên
 - Thoát: thoát khỏi chương trình

III. Bảng cơ sở dữ liệu

1. Bảng câu hỏi

Tên trường	Kiểu	Độ rộng	Khóa	Null
NoiDung	Nvarchar	150		No
DapAn_A	Nvarchar	150		No
DapAn_B	Nvarchar	150		No
DapAn_C	Nvarchar	150		No
DapAn_D	Nvarchar	150		No
DapAnDung	Nvarchar	150		No
Level	Int	11		No
ID	int	11	PK	No

2. Bảng người chơi

Tên trường	Kiểu	Độ rộng	Khóa	Null
TENNGUOICHOI	Nvarchar	50	PK	No
SOCAU	int	11		No
TONGTHOIGIAN	int	11		No
THOIGIANTHUC	datetime			No

IV. Kết luận

Với kiến thức nền tảng đã được học ở trường và bằng sự nỗ lực của mình, nhóm em đã hoàn thành đề tài “Lập trình trò chơi Ai là triệu phú”. Do trình độ có hạn, nên đồ án không thể hoàn thành một cách hoàn hảo các chức năng. Tuy nhiên, nhóm em đã cố gắng hết sức để những chức năng đó tốt nhất trong khả năng của nhóm em và những chức năng chưa hoàn thiện sẽ sớm được bổ sung và hoàn chỉnh trong thời gian sớm nhất.

Trong quá trình thiết kế và thực hiện đồ án không tránh khỏi sai sót, mong thầy và các bạn thông cảm và có những góp ý quý báu nhằm hoàn thiện hơn cho sản phẩm.