

```
private void rdoUpdateActionPerformed(java.awt.event.ActionEvent evt) {  
    lblCode.setVisible(true);  
    lblGame.setVisible(true);  
    lblGenre.setVisible(true);  
    txtCode.setVisible(true);  
    txtGenre.setVisible(true);  
    txtGenre.setEditable(true);  
    txtGenre.setBackground(Color.WHITE);  
    txtGame.setVisible(true);  
    txtGame.setEditable(true);  
    txtGame.setBackground(Color.WHITE);  
    btnSubmit.setText("Update");  
    btnSubmit.setVisible(true);  
    btnExit.setVisible(true);  
    lblRating.setVisible(true);  
    txtRating.setVisible(true);  
    txtRating.setEditable(true);  
    txtRating.setBackground(Color.WHITE);  
    txtPrice.setVisible(true);  
    txtPrice.setEditable(true);  
    lblPrice.setVisible(true);  
    txtPrice.setBackground(Color.WHITE);  
}
```

```
private void rdoDeleteActionPerformed(java.awt.event.ActionEvent evt) {  
    lblCode.setVisible(true);  
    lblGame.setVisible(false);  
    lblGenre.setVisible(false);  
    txtCode.setVisible(true);  
    txtGenre.setVisible(false);  
    txtGame.setVisible(false);  
    btnSubmit.setText("Delete");  
    btnSubmit.setVisible(true);  
    btnExit.setVisible(true);  
    lblRating.setVisible(false);  
    lblGenre.setVisible(false);  
    txtRating.setVisible(false);  
    txtGenre.setVisible(false);  
    txtPrice.setVisible(false);  
    lblPrice.setVisible(false);  
}
```

```
private void btnSubmitActionPerformed(java.awt.event.ActionEvent evt) {  
    if (rdoDisplay.isSelected()) {  
        displayIt();  
    }  
    else if (rdoInsert.isSelected()) {  
        insertIt();  
    }  
    else if (rdoUpdate.isSelected()) {  
        updateIt();  
    }  
    else if (rdoDelete.isSelected()) {  
        deleteIt();  
    }  
}
```

```
private void btnExitActionPerformed(java.awt.event.ActionEvent evt) {  
    System.exit(0);  
}
```

```
public void displayIt() {  
    try {  
        Games newGame = new Games();  
        GameDB newGameDB = new GameDB();  
        newGame = newGameDB.getProduct(txtCode.getText());  
        txtCode.setText(newGame.getCode());  
        txtGame.setText(newGame.getGame());  
        txtGenre.setText(newGame.getGenre());  
        txtRating.setText(newGame.getRating());  
        txtPrice.setText(newGame.getFormattedPrice());  
    }  
    catch (Exception e) {  
        {  
            txtCode.setText("");  
            txtGame.setText("");  
            txtPrice.setText("");  
            txtGenre.setText("");  
            txtRating.setText("");  
        }  
    }  
}
```

```

private Connection getConnection()
{
    try
    {
        conn = DriverManager.getConnection(url, user, password);
        return conn;
    }
    catch(SQLException e)
    {
        System.err.println(e);
        return null;
    }
}

public ArrayList<Games> getProducts()
{
    String sql = "SELECT productCode, game, genre, rating, Price "
                + "From Games "
                + "ORDER BY productCode";
    ArrayList<Games> products = new ArrayList();

    try (Connection connection = getConnection();
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery(sql))
    {
        while (rs.next())
        {
            String code = rs.getString("ProductCode");
            String game = rs.getString("Game");
            String genre = rs.getString("Genre");
            String rating = rs.getString("Rating");
            Double price = rs.getDouble("Price");
            Games p = new Games(code, game, genre, rating, price);
            products.add(p);
        }
        return products;
    }
    catch (SQLException e)
    {
        System.err.println(e);
        return null;
    }
}

```

```

public Games getProduct (String code)
{
    String sql = "SELECT productCode, game, genre, rating, Price "
        + "From Games "
        + "WHERE ProductCode = '" + code + "'";

    try {Connection connection = getConnection();
        Statement stmt = conn.createStatement();

        ResultSet rs = stmt.executeQuery(sql);
        if (rs.next())
        {
            String game = rs.getString("Game");
            String genre = rs.getString("Genre");
            String rating = rs.getString("Rating");
            double price = rs.getDouble("Price");
            Games p = new Games(code, game, genre, rating, price);
            rs.close();
            return p;
        }
        else
        {
            rs.close();
            return null;
        }
    }
    catch (SQLException e)
    {
        System.err.println(e);
        return null;
    }
}

```

```

public void insertIt(){
    Games newGame = new Games();
    GameDB newGameDB = new GameDB();

    newGame = newGameDB.addProduct(txtCode.getText(), txtGame.getText(), txtGenre.getText(), txtRating.getText(), txtPrice.getText());
    txtRating.setEditable(true);
    txtGenre.setEditable(true);
    txtPrice.setEditable(true);
    txtPrice.setEditable(true);
}

public void updateIt(){
    Games newGame = new Games();
    GameDB newGameDB = new GameDB();

    newGame = newGameDB.updateProduct(txtCode.getText(), txtGame.getText(), txtGenre.getText(), txtRating.getText(), txtPrice.getText());
    txtRating.setEditable(true);
    txtGenre.setEditable(true);
    txtPrice.setEditable(true);
    txtPrice.setEditable(true);
}

public void deleteIt(){
    Games newGame = new Games();
    GameDB newGameDB = new GameDB();
    newGame = newGameDB.deleteProduct(txtCode.getText());
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new GameDatabase().setVisible(true);
        }
    });
}

```

```

public String getFormattedPrice()
{
    NumberFormat currency = NumberFormat.getCurrencyInstance();
    return currency.format(price);
}

```

```

@Override
public String toString()
{
    return "Code:      " + code + "\n" +
           "Description: " + game + "\n" +
           "Price:      " + this.getFormattedPrice() + "\n";
}

```

```

public static int getCount()
{
    return count;
}

```

```

public static Games deleteProduct(String code)
{
    String url = "jdbc:mysql://localhost:3306/VideoGames";
    String user = "root";
    String password = "";

    Connection conn = null;
    Statement stmt = null;

    Games character = new Games();
        String query = "DELETE FROM Games "
                        + "WHERE productCode = '"
                        + code + "'";
        System.out.println (query);

        try {
            conn = DriverManager.getConnection(url, user, password);
            stmt = conn.createStatement();
            stmt.executeUpdate(query);
        }
        catch (Exception e)
        {
            JOptionPane.showMessageDialog(null, e.toString());
        }
    return character;
}

public Games modifyProduct (String code, String description, String price)
{
    String sql = "UPDATE Products"
                + "SET ProductCode = '"
                + code + "'";
    System.out.println (sql);

    try (Connection connection = getConnection();
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery(sql);)
    {
        rs.close();
        return null;
    }
    catch (SQLException e)
    {
        System.err.println(e);
        return null;
    }
}

```

```
public class Games {
    private String code;
    private String game;
    private String genre;
    private String rating;
    private double price;
    protected static int count = 0;

    public Games()
    {
        code = "";
        game = "";
        genre = "";
        rating = "";
        price = 0;
    }

    public Games (String codeIn, String gameIn, String genreIn, String ratingIn, double priceIn)
    {
        setCode(codeIn);
        this.game = gameIn;
        genre = genreIn;
        rating = ratingIn;
        price = priceIn;
    }

    public void setCode(String code)
    {
        this.code = code;
    }

    public String getCode(){
        return code;
    }

    public void setGame(String game)
    {
        this.game = game;
    }

    public String getGame()
    {
        return game;
    }
}
```

```
private void rdoDisplayActionPerformed(java.awt.event.ActionEvent evt) {  
    lblCode.setVisible(true);  
    lblGame.setVisible(true);  
    lblGenre.setVisible(true);  
    txtCode.setVisible(true);  
    txtGenre.setVisible(true);  
    txtGenre.setEditable(false);  
    txtGenre.setBackground(Color.LIGHT_GRAY);  
    txtGame.setVisible(true);  
    txtGame.setEditable(false);  
    txtGame.setBackground(Color.LIGHT_GRAY);  
    btnSubmit.setText("Display");  
    btnSubmit.setVisible(true);  
    btnExit.setVisible(true);  
    lblRating.setVisible(true);  
    txtRating.setVisible(true);  
    txtRating.setEditable(false);  
    txtRating.setBackground(Color.LIGHT_GRAY);  
    txtPrice.setVisible(true);  
    txtPrice.setEditable(false);  
    lblPrice.setVisible(true);  
    txtPrice.setBackground(Color.LIGHT_GRAY);  
}
```

```
private void rdoInsertActionPerformed(java.awt.event.ActionEvent evt) {  
    lblCode.setVisible(true);  
    lblGame.setVisible(true);  
    lblGenre.setVisible(true);  
    txtCode.setVisible(true);  
    txtGenre.setVisible(true);  
    txtGenre.setEditable(true);  
    txtGenre.setBackground(Color.WHITE);  
    txtGame.setVisible(true);  
    txtGame.setEditable(true);  
    txtGame.setBackground(Color.WHITE);  
    btnSubmit.setText("Insert");  
    btnSubmit.setVisible(true);  
    btnExit.setVisible(true);  
    lblRating.setVisible(true);  
    txtRating.setVisible(true);  
    txtRating.setEditable(true);  
    txtRating.setBackground(Color.WHITE);  
    txtPrice.setVisible(true);  
    txtPrice.setEditable(true);  
    lblPrice.setVisible(true);  
    txtPrice.setBackground(Color.WHITE);  
}
```