```java
public class Piece
{
    private Color color;
    private int row;
    private int col;

    public Piece(Color pColor, int pRow, int pCol)
    {
        color = pColor;
        row = pRow;
        col = pCol;
    }

    public void moveTo(int pRow, int pCol)
    {
        row = pRow;
        col = pCol;
    }

    public Color getColor()
    {
        return color;
    }

    public int getRow()
    {
        return row;
    }

    public int getCol()
    {
        return col;
    }

    public void draw(Graphics g, int offsetLeft, int offsetTop, int squareSide)
    {
        g.setColor(color);
        g.fillOval(offsetLeft + (col * squareSide), offsetTop + (row * squareSide), squareSide, squareSide);
    }
}
```

```java
public class Checkers
{
    private ArrayList<Piece> alPieces = new ArrayList<>();

    public void init()
    {
     int size = 8;
        for (int r=0; r<size; r++)
        {
            for (int c=0; c<size; c++)
            {
                if ((r+c)%2 == 1)
                {
                    if (r < 3)
                        alPieces.add(new Piece(Color.BLUE, r, c));
                    else if (r >= 5)
                        alPieces.add(new Piece(Color.RED, r, c));
                }
            }
        }
    }

    public Piece getPieceAt(int pRow, int pCol)
    {
        Piece p;
        boolean found = false;
        int i = 0;
        while (!found && (i<alPieces.size()))
        {
            p = alPieces.get(i);
            if ((pRow == p.getRow()) && (pCol == p.getCol()))
                found = true;
            else
                i++;
        }
        if (found)
            return alPieces.get(i);
        else
            return null;
    }

    public boolean move(int pStartRow, int pStartCol, int pDestRow, int pDestCol)
    {
        Piece p = getPieceAt(pStartRow, pStartCol);
        if (p == null)
            return false;               // Pas de pièce à cette position de départ --> pas bouger

        if ((pDestRow<0) || (pDestCol<0) || (pDestRow>7) || (pDestCol>7))
        {
            alPieces.remove(p);         // Position en dehors du damier --> effacer la pièce
            return true;
        }

        if (getPieceAt(pDestRow, pDestCol) == null)
        {
            p.moveTo(pDestRow, pDestCol);   // Pas de pièce à la position d'arrivée --> on bouge la pièce
            return true;
        }

        return false;
    }

    public void draw(Graphics g, int width, int height)
    {
        int size = 8;
        int l = Math.min(width, height)-1;
        int cellSize = l / size;

        // Calculer les décalages
        int offsetLeft = (wwidth-(cellSize*size))/2;
        int offsetTop = (height-(cellSize*size))/2;

        // Noir: la couleur de la grille
        int c, r;
        for (int i=0; i<size; i++)
        {
            r = (i*cellSize);
            for (int j=0; j<size; j++)
            {
                c = (j*cellSize);
                if ((i+j)%2 == 1)
                {
                    g.setColor(Color.GRAY);
                    g.fillRect(offsetLeft + c, offsetTop + r, cellSize, cellSize);
                }
                g.setColor(Color.BLACK);
                g.drawRect(offsetLeft + c, offsetTop + r, cellSize, cellSize);
            }
        }

        // Dessiner les pièces... chaque pièce se dessine elle-même!
        for (int i=0; i<alPieces.size(); i++)
            alPieces.get(i).draw(g, offsetLeft, offsetTop, cellSize);
    }
}
```

```java
public class DrawPanel extends javax.swing.JPanel
{
    private Checkers checkers = null;

    public void setCheckers(Checkers checkers)
    {
        this.checkers = checkers;
    }

    public DrawPanel()
    {
        initComponents();
    }

    public void paintComponent(Graphics g)
    {
        int w = getWidth();
        int h = getHeight();

        // Peindre l'arrière fond
        g.setColor(Color.WHITE);
        g.fillRect(0, 0, w, h);

        // Dessiner le damier et les pièces... s'il existe
        if (checkers != null)
            checkers.draw(g, w, h);
    }
// Skipped: ... initComponents { ... }
    // Variables declaration - do not modify//GEN-BEGIN:variables
    // End of variables declaration//GEN-END:variables
}
```

```java
public class MainFrame extends javax.swing.JFrame
{
    // Data-Model....
    private Checkers checkers = null;

    public MainFrame()
    {
        initComponents();
    }
// Skipped: ... initComponents { ... }
    private void moveButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_moveButtonActionPerformed
    {//GEN-HEADEREND:event_moveButtonActionPerformed
        if (checkers == null)
            return;

        int startRow = Integer.valueOf(startRowTextField.getText());
        int startCol = Integer.valueOf(startColTextField.getText());
        int destRow = Integer.valueOf(destRowTextField.getText());
        int destCol = Integer.valueOf(destColTextField.getText());

        if (!checkers.move(startRow, startCol, destRow, destCol))
            errorLabel.setText("Invalid move!");
        else
            errorLabel.setText("");

        repaint();
    }//GEN-LAST:event_moveButtonActionPerformed

    private void newButtonActionPerformed(java.awt.event.ActionEvent evt)//GEN-FIRST:event_newButtonActionPerformed
    {//GEN-HEADEREND:event_newButtonActionPerformed
        checkers = new Checkers();
        checkers.init();

        drawPanel.setCheckers(checkers);
        errorLabel.setText("");

        repaint();
    }//GEN-LAST:event_newButtonActionPerformed
// Skipped: ... Look & Feel
    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JTextField destColTextField;
    private javax.swing.JTextField destRowTextField;
    private DrawPanel drawPanel;
    private javax.swing.JLabel errorLabel;
    private javax.swing.JComboBox jComboBox1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JButton moveButton;
    private javax.swing.JButton newButton;
    private javax.swing.JTextField startColTextField;
    private javax.swing.JTextField startRowTextField;
    // End of variables declaration//GEN-END:variables
}
```