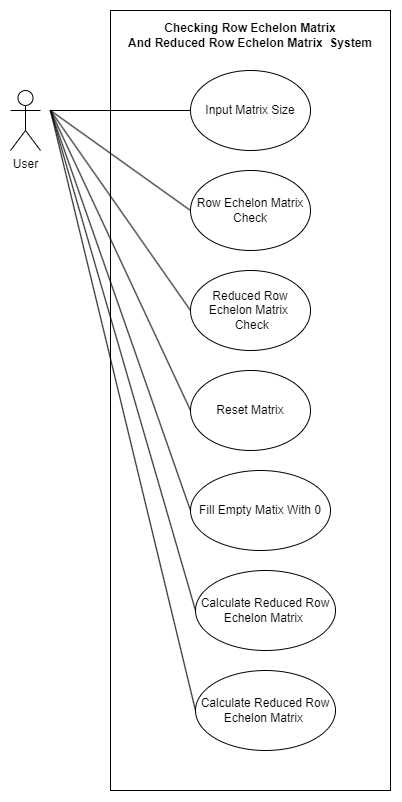
**อธิบายแนวคิด/ ไอเดีย พร้อมทั้งวาด Use case ,Sequence และ Class diagram**

**ตอบ** แนวคิด มาจากการเรียนวิชาพีชคณิตเชิงเส้นพื้นฐาน แล้วเกิดปัญหาในการแก้โจทย์ปัญหา การตรวจคำตอบ เกี่ยวกับ Reduced row echelon matrix และ Row echelon matrix ทำให้อยากสร้างโปรแกรมที่ช่วยในการเรียนรู้เกี่ยวกับเรื่อง Reduced row echelon matrix และ Row echelon matrix

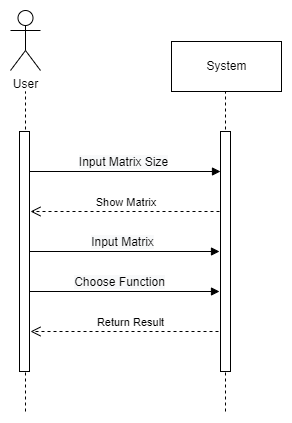
โดยโปรแกรมจะทำการรับขนาดของ matrix มาแล้วจึงให้ใส่ค่าของ matrix ในแต่ละช่วงโดยผู้ใช้สามารถเลือกที่จะกดปุ่ม “reset matrix” เพื่อ reset ค่าของ matrix , “fill empty matrix with 0” เพื่อใส่ 0 ในช่อง matrix ที่ว่าง , “REM check” เพื่อที่จะตรวจสอบว่า matrix นั้นเป็น Row echelon matrix หรือเปล่า , “RREM check” เพื่อที่จะตรวจสอบว่า matrix นั้นเป็น Reduced row echelon matrix หรือเปล่า , “calculate REM” เพื่อที่จะแสดงวิธีในการเปลี่ยน matrix นั้นให้เป็น Row echelon matrix , “calculate RREM” เพื่อที่จะแสดงวิธีในการเปลี่ยน matrix นั้นให้เป็น Reduced row echelon matrix

โดยปุ่ม “reset matrix” จะเปลี่ยนช่องของ JTextField ของ matrix ให้เป็น “”  
 โดยปุ่ม “fill empty matrix with 0” จะเปลี่ยนช่องของ JTextField ของ matrix ที่เป็น “” ให้เป็น “0”  
 โดยปุ่ม “REM check” จะทำการดึงค่า JTextField ของ matrix มาคำนวณ แล้วบอกว่าเป็นหรือไม่เป็น Row echelon matrix  
 โดยปุ่ม “RREM check” จะทำการดึงค่า JTextField ของ matrix มาคำนวณ แล้วบอกว่าเป็นหรือไม่เป็น Reduced row echelon matrix  
 โดยปุ่ม “calculate REM” จะทำการดึงค่า JTextField ของ matrix มาคำนวณ แล้วแสดงขั้นตอนในการทำให้ matrix กลายเป็น Row echelon matrix  
 โดยปุ่ม “calculate RREM” จะทำการดึงค่า JTextField ของ matrix มาคำนวณ แล้วแสดงขั้นตอนในการทำให้ matrix กลายเป็น Reduced row echelon matrix

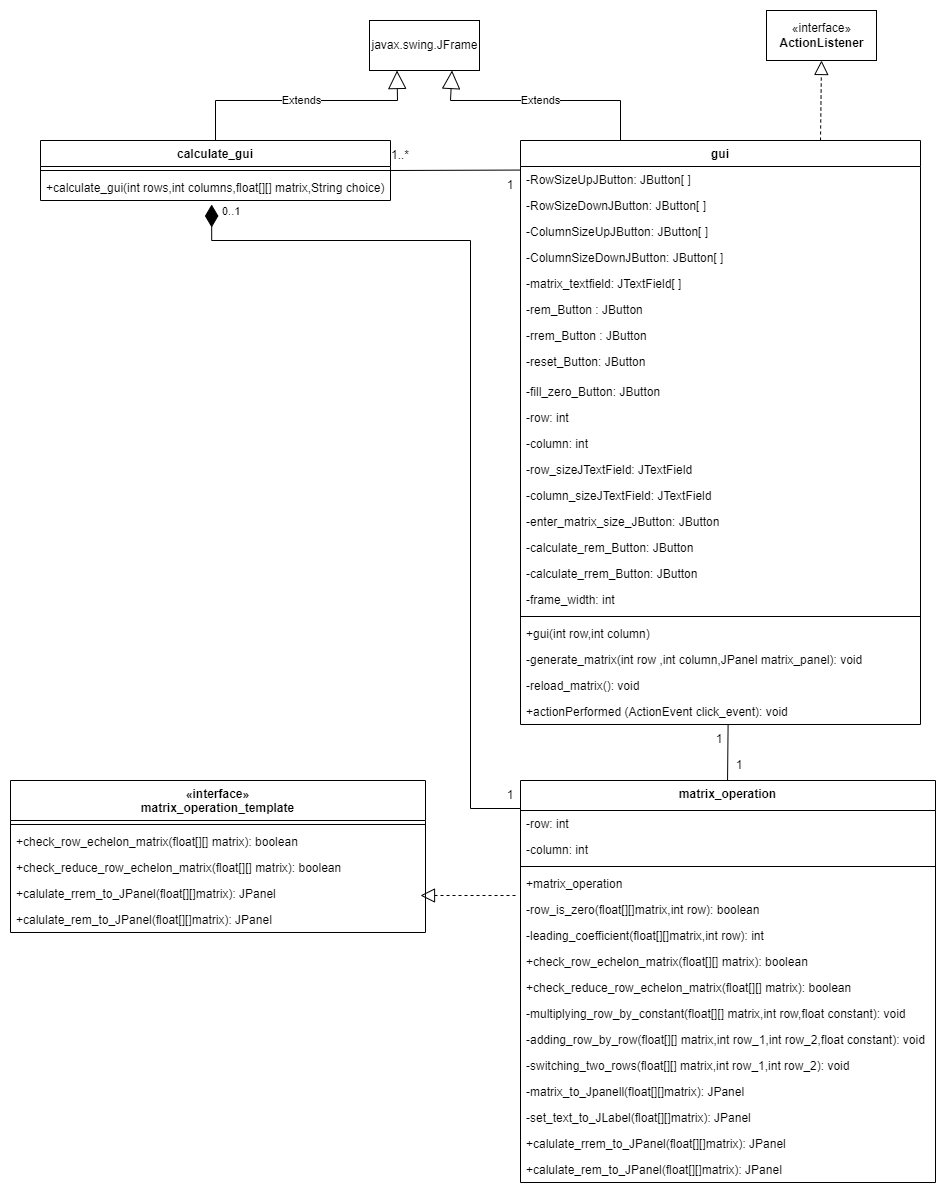
**Use case diagram**



**Sequence diagram**



**Class diagram**



|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303  304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347  348  349  350  351  352  353  354  355  356  357  358  359  360  361  362  363  364  365  366  367  368  369  370  371  372  373  374  375  376  377  378  379  380  381  382  383  384  385  386  387  388  389  390  391  392  393  394  395  396  397  398  399  400  401  402  403  404  405  406  407  408  409  410  411  412  413  414  415  416  417  418  419  420  421  422  423  424  425  426  427  428  429  430  431  432  433  434  435  436  437  438  439  440  441  442  443  444  445  446  447  448  449  450  451  452  453  454  455  456  457  458  459  460  461  462  463  464  465  466  467  468  469  470  471  472  473  474  475  476  477  478  479  480  481  482  483  484  485  486  487  488  489  490  491  492  493  494  495  496  497  498  499  500  501  502  503  504  505  506  507  508  509  510  511  512  513  514  515  516  517  518  519  520  521  522  523  524  525  526  527  528  529  530  531  532  533  534  535  536  537  538  539  540  541  542  543  544  545  546  547  548  549  550  551  552  553  554  555  556  557  558  559  560  561  562  563  564  565  566  567  568  569  570  571  572  573  574  575  576  577  578  579  580  581  582  583  584  585  586  587  588  589  590  591  592  593  594  595  596  597  598  599  600  601  602  603  604  605  606  607  608  609  610  611  612  613  614  615  616  617  618  619  620  621  622  623  624  625  626  627  628  629  630  631  632  633  634  635  636  637  638  639  640  641  642  643  644  645  646  647  648  649  650  651  652  653  654  655  656  657  658  659  660  661  662  663  664  665  666  667  668  669  670  671  672  673  674  675  676  677  678  679  680  681  682  683  684  685  686  687  688  689  690  691  692  693  694  695  696  697  698  699  700  701  702  703  704  705  706  707  708  709  710  711  712  713  714  715  716  717  718  719  720  721  722  723  724  725 | import java.awt.\*;  import java.awt.event.\*;  import javax.swing.\*;  public class main\_final {  public static void main(String[] arngs) {  new gui(1, 1);  }  }  class gui extends javax.swing.JFrame implements ActionListener { //เพิ่มให้ class extends javax.swing.JFrame  private int max = 10;  private JButton RowSizeUpJButton = new JButton();  private JButton RowSizeDownJButton = new JButton();  private JButton ColumnSizeUpJButton = new JButton();  private JButton ColumnSizeDownJButton = new JButton();  private JTextField matrix\_textfield[][] = new JTextField[max][max];  private JButton rem\_Button = new JButton();  private JButton rrem\_Button = new JButton();  private JButton reset\_Button = new JButton();  private JButton fill\_zero\_Button = new JButton();  private int row, column;  JFrame frame = new JFrame();  Container container = frame.getContentPane();  private JPanel main\_panel = new JPanel();  private JPanel matrix\_panel = new JPanel();  private JPanel button\_panel = new JPanel();  private JPanel top\_panel = new JPanel();  private JPanel matrix\_panel\_center = new JPanel();  private JTextField row\_sizeJTextField = new JTextField("", 3);  private JTextField column\_sizeJTextField = new JTextField("", 3);  private JButton enter\_matrix\_size\_JButton = new JButton("Enter Matrix Size");  private JButton calculate\_rem\_Button = new JButton();  private JButton calculate\_rrem\_Button = new JButton();  GridBagConstraints c = new GridBagConstraints();  private int frame\_width;  gui(int row, int column) {  this.row = row;  this.column = column;  //////////////////////////////////////////////////////////////////////////////ทำให้ปุ่มใช้ feature ที่มีใน javax.swing มาใช้  RowSizeUpJButton = new javax.swing.JButton();  RowSizeDownJButton = new javax.swing.JButton();  ColumnSizeUpJButton = new javax.swing.JButton();  ColumnSizeDownJButton = new javax.swing.JButton();  rem\_Button = new javax.swing.JButton("REM check");  rrem\_Button = new javax.swing.JButton("RREM check");  reset\_Button = new javax.swing.JButton("reset matrix");  fill\_zero\_Button = new javax.swing.JButton("fill empty matix with 0");  calculate\_rem\_Button = new javax.swing.JButton("calculate REM");  calculate\_rrem\_Button = new javax.swing.JButton("calculate RREM");  ////////////////////////////////////////////////////////////////////////// set รูป/ลักษณะปุ่มให้ปุ่ม , กำหนดตัวอักษร  RowSizeUpJButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("up.png")));  RowSizeUpJButton.setBorder(new javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));  RowSizeDownJButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("down.png")));  RowSizeDownJButton.setBorder(new javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));  ColumnSizeUpJButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("right.png")));  ColumnSizeUpJButton.setBorder(new javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));  ColumnSizeDownJButton.setIcon(new javax.swing.ImageIcon(getClass().getResource("left.png")));  ColumnSizeDownJButton.setBorder(new javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));  rem\_Button.setFont(new java.awt.Font("Tahoma", 1, 13));  rrem\_Button.setFont(new java.awt.Font("Tahoma", 1, 13));  reset\_Button.setFont(new java.awt.Font("Tahoma", 1, 13));  fill\_zero\_Button.setFont(new java.awt.Font("Tahoma", 1, 13));  calculate\_rem\_Button.setFont(new java.awt.Font("Tahoma", 1, 13));  calculate\_rrem\_Button.setFont(new java.awt.Font("Tahoma", 1, 13));  row\_sizeJTextField.setHorizontalAlignment(SwingConstants.CENTER);  column\_sizeJTextField.setHorizontalAlignment(SwingConstants.CENTER);  ////////////////////////////////////////////////////////////////////////////////////////////////////////  container.setLayout(new BorderLayout());  //////////////////////////////////////////////////////ทำ panel ย่อย  main\_panel.setLayout(new GridBagLayout());  matrix\_panel.setLayout(new GridBagLayout());  button\_panel.setLayout(new GridBagLayout());  top\_panel.setLayout(new GridBagLayout());  matrix\_panel\_center.setLayout(new GridBagLayout());  //////////////////////////////////////////////////////  //////////////////////////////////////////////////////สร้างช่อง matrix  for (int i = 0; i < max; i++) {  for (int l = 0; l < max; l++) {  matrix\_textfield[i][l] = new JTextField("", 3);  matrix\_textfield[i][l].setHorizontalAlignment(SwingConstants.CENTER);  }  }  //////////////////////////////////////////////////////  //////////////////////////////////////////////////////จัดองค์ประกอบต่างๆ  c.insets = new Insets(5, 5, 5, 5);  for (int i = 0; i < row; i++) {  for (int l = 0; l < column; l++) {  c.gridx = l;  c.gridy = i;  matrix\_panel.add(matrix\_textfield[i][l], c);  }  }  c.gridx = 1;  c.gridy = 1;  button\_panel.add(rem\_Button, c);  c.gridx = 2;  c.gridy = 1;  button\_panel.add(rrem\_Button, c);  c.gridx = 1;  c.gridy = 2;  button\_panel.add(calculate\_rem\_Button, c);  c.gridx = 2;  c.gridy = 2;  button\_panel.add(calculate\_rrem\_Button, c);  c.gridx = 1;  c.gridy = 1;  matrix\_panel\_center.add(matrix\_panel, c);  c.gridx = 1;  c.gridy = 2;  main\_panel.add(reset\_Button, c);  c.gridx = 1;  c.gridy = 3;  main\_panel.add(fill\_zero\_Button, c);  c.gridx = 1;  c.gridy = 4;  main\_panel.add(button\_panel, c);  c.gridx = 2;  c.gridy = 1;  top\_panel.add(RowSizeUpJButton, c);  c.gridx = 2;  c.gridy = 2;  top\_panel.add(RowSizeDownJButton, c);  c.gridx = 6;  c.gridy = 1;  top\_panel.add(ColumnSizeUpJButton, c);  c.gridx = 5;  c.gridy = 1;  top\_panel.add(ColumnSizeDownJButton, c);  c.gridx = 1;  c.gridy = 1;  top\_panel.add(row\_sizeJTextField, c);  row\_sizeJTextField.setText(row + "");  c.gridx = 4;  c.gridy = 1;  top\_panel.add(column\_sizeJTextField, c);  column\_sizeJTextField.setText(column + "");  c.gridx = 3;  c.gridy = 1;  top\_panel.add(enter\_matrix\_size\_JButton, c);  //////////////////////////////////////////////////////  container.add(top\_panel, BorderLayout.NORTH);  container.add(matrix\_panel\_center, BorderLayout.CENTER);  container.add(main\_panel, BorderLayout.SOUTH);  fill\_zero\_Button.addActionListener(this);  reset\_Button.addActionListener(this);  rem\_Button.addActionListener(this);  rrem\_Button.addActionListener(this);  RowSizeUpJButton.addActionListener(this);  RowSizeDownJButton.addActionListener(this);  ColumnSizeUpJButton.addActionListener(this);  ColumnSizeDownJButton.addActionListener(this);  enter\_matrix\_size\_JButton.addActionListener(this);  calculate\_rem\_Button.addActionListener(this);  calculate\_rrem\_Button.addActionListener(this);  frame.pack();  //frame.setSize(500,550);  frame.setVisible(true);  //frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  frame.setLocationRelativeTo(null);  frame\_width = frame.getWidth();  }  private void generate\_matrix(int row, int column, JPanel matrix\_panel) {//เอาไว้สร้าง matrix  matrix\_panel.setLayout(new GridBagLayout());  GridBagConstraints c = new GridBagConstraints();  c.insets = new Insets(5, 5, 5, 5);  for (int i = row; i < max; i++) {  for (int l = 0; l < max; l++) {  matrix\_textfield[i][l].setText("");  }  }  for (int i = 0; i < max; i++) {  for (int l = column; l < max; l++) {  matrix\_textfield[i][l].setText("");  }  }  for (int i = 0; i < row; i++) {  for (int l = 0; l < column; l++) {  c.gridx = l;  c.gridy = i;  matrix\_panel.add(matrix\_textfield[i][l], c);  }  }  }  private void reload\_matrix() {//เอาไว้ reload matrix เมื่อเกิดการเพิ่มขนาด matrix  matrix\_panel.setLayout(new GridBagLayout());  matrix\_panel.removeAll();  generate\_matrix(row, column, matrix\_panel);  frame.revalidate();  row\_sizeJTextField.setText(row + "");  column\_sizeJTextField.setText(column + "");  frame.pack();  Point frame\_location = frame.getLocation();  frame.setLocation((int) frame\_location.getX() - (frame.getWidth() - frame\_width) / 2, (int) frame\_location.getY());  frame\_width = frame.getWidth();  }  @Override  public void actionPerformed(ActionEvent click\_event) {  matrix\_operation m\_o = new matrix\_operation(row, column);  try {  if (click\_event.getSource() == reset\_Button) {  for (int i = 0; i < row; i++) {  for (int l = 0; l < column; l++) {  matrix\_textfield[i][l].setText("");  }  }  } else if (click\_event.getSource() == rem\_Button) {  float[][] matrix = new float[row][column];  for (int i = 0; i < row; i++) {  for (int l = 0; l < column; l++) {  matrix[i][l] = Float.parseFloat(matrix\_textfield[i][l].getText());  }  }  JOptionPane.showMessageDialog(null, "Row Echelon Matrix : " + m\_o.check\_row\_echelon\_matrix(matrix));  } else if (click\_event.getSource() == rrem\_Button) {  float[][] matrix = new float[row][column];  for (int i = 0; i < row; i++) {  for (int l = 0; l < column; l++) {  matrix[i][l] = Float.parseFloat(matrix\_textfield[i][l].getText());  }  }  JOptionPane.showMessageDialog(null, "Reduce Row Echelon Matrix : " + m\_o.check\_reduce\_row\_echelon\_matrix(matrix));  } else if (click\_event.getSource() == fill\_zero\_Button) {  for (int i = 0; i < row; i++) {  for (int l = 0; l < column; l++) {  if ((matrix\_textfield[i][l].getText().isBlank())) {  matrix\_textfield[i][l].setText("0");  }  }  }  } else if (click\_event.getSource() == RowSizeUpJButton) {  if (row < max) {  row++;  reload\_matrix();  }  } else if (click\_event.getSource() == RowSizeDownJButton) {  if (row > 1) {  row--;  reload\_matrix();  }  } else if (click\_event.getSource() == ColumnSizeUpJButton) {  if (column < max) {  column++;  reload\_matrix();  }  } else if (click\_event.getSource() == ColumnSizeDownJButton) {  if (column > 1) {  column--;  reload\_matrix();  }  } else if (click\_event.getSource() == enter\_matrix\_size\_JButton) {  try {  if (Integer.parseInt(column\_sizeJTextField.getText()) >= 1 && Integer.parseInt(column\_sizeJTextField.getText()) <= max  && Integer.parseInt(row\_sizeJTextField.getText()) >= 1 && Integer.parseInt(row\_sizeJTextField.getText()) <= max) {  column = Integer.parseInt(column\_sizeJTextField.getText());  row = Integer.parseInt(row\_sizeJTextField.getText());  reload\_matrix();  } else {  JOptionPane.showMessageDialog(null, "Pls Matrix Size in 1 to " + max);  }  } catch (Exception e) {  JOptionPane.showMessageDialog(null, "Pls Input Only Integer In Matrix Size"/\*+e\*/);  }  } else if (click\_event.getSource() == calculate\_rem\_Button) {  float[][] matrix = new float[row][column];  for (int i = 0; i < row; i++) {  for (int l = 0; l < column; l++) {  matrix[i][l] = Float.parseFloat(matrix\_textfield[i][l].getText());  }  }  if (m\_o.check\_row\_echelon\_matrix(matrix) == true) {  JOptionPane.showMessageDialog(null, "It Already Is Row Echelon Matrix");  } else {  new calculate\_gui(row, column, matrix, "rem");  }  } else if (click\_event.getSource() == calculate\_rrem\_Button) {  float[][] matrix = new float[row][column];  for (int i = 0; i < row; i++) {  for (int l = 0; l < column; l++) {  matrix[i][l] = Float.parseFloat(matrix\_textfield[i][l].getText());  }  }  if (m\_o.check\_reduce\_row\_echelon\_matrix(matrix) == true) {  JOptionPane.showMessageDialog(null, "It Already Is Reduce Row Echelon Matrix");  } else {  new calculate\_gui(row, column, matrix, "rrem");  }  }  } catch (Exception e) {  JOptionPane.showMessageDialog(null, "Pls Input Only Float In Matrix"/\*+e\*/);  }  }  }  class calculate\_gui extends javax.swing.JFrame {  private JFrame frame = new JFrame();  private Container container = frame.getContentPane();  calculate\_gui(int rows, int columns, float[][] matrix, String choice) {  matrix\_operation m\_o = new matrix\_operation(rows, columns);  container.setLayout(new BorderLayout());  JPanel calculate\_panel = new JPanel();  JPanel subPanel = new JPanel();  subPanel.setLayout(new GridBagLayout());  GridBagConstraints c = new GridBagConstraints();  JLabel titleLabel = new JLabel();  titleLabel.setFont(new java.awt.Font("Tahoma", 1, 13));  if (choice.equals("rem")) {  titleLabel.setText("Calculate Row Echelon Matrix");  c.gridx = 0;  c.gridy = 0;  subPanel.add(titleLabel, c);  c.gridx = 0;  c.gridy = 1;  subPanel.add(m\_o.calulate\_rem\_to\_JPanel(matrix), c);  calculate\_panel.add(subPanel, BorderLayout.CENTER);  } else if (choice.equals("rrem")) {  titleLabel.setText("Calculate Reduced Row Echelon Matrix");  c.gridx = 0;  c.gridy = 0;  subPanel.add(titleLabel, c);  c.gridx = 0;  c.gridy = 1;  subPanel.add(m\_o.calulate\_rrem\_to\_JPanel(matrix), c);  calculate\_panel.add(subPanel, BorderLayout.CENTER);  }  JScrollPane ScrollPane = new JScrollPane(calculate\_panel, JScrollPane.VERTICAL\_SCROLLBAR\_AS\_NEEDED, JScrollPane.HORIZONTAL\_SCROLLBAR\_NEVER);  container.add(ScrollPane, BorderLayout.CENTER);  //frame.pack();  frame.setSize(500, 550);  frame.setVisible(true);  //frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  frame.setLocationRelativeTo(null);  }  }  interface matrix\_operation\_template {  boolean check\_row\_echelon\_matrix(float[][] matrix);  boolean check\_reduce\_row\_echelon\_matrix(float[][] matrix);  JPanel calulate\_rrem\_to\_JPanel(float[][] matrix);  JPanel calulate\_rem\_to\_JPanel(float[][] matrix);  }  class matrix\_operation implements matrix\_operation\_template {  /\*  rows คือ จำนวนแถว  columns คือ จำนวนหลัก  ปัญหาคือ เวลาเก็บใน array มันเริ่มที่ 0 แต่ใน matrix มันเริ่มที่ 1 เวลาใช้ method เลยนับว่าให้เริ่มที่ 1 ตาม matrix  \*/  private int rows, columns;  matrix\_operation(int rows, int columns) {//กำหนดจำนวนแถวและหลัก  this.rows = rows;  this.columns = columns;  }  private boolean row\_is\_zero(float[][] matrix, int row) {//return ว่าแถวเป็น 0 ทั้งแถวหรือเปล่า  boolean row\_is\_zero = true;  for (int i = 0; i < columns; i++) {  if (matrix[row - 1][i] == 0) {  } else {  row\_is\_zero = false;  }  }  if (row\_is\_zero == true) {  return true;  } else {  return false;  }  }  private int leading\_coefficient(float[][] matrix, int row) {//return ตำแหน่งของตัวนำ 1 ในแถว ถ้าไม่มีจะ return -1  for (int i = 0; i < columns; i++) {  if (matrix[row - 1][i] == 1) {  return i + 1;  } else if (matrix[row - 1][i] == 0) {  } else {  return -1;  }  }  return -1;  }  public boolean check\_row\_echelon\_matrix(float[][] matrix) {//เช็คว่าเป็น row echelon matrix เปล่า  if (rows == 1) {  if ((row\_is\_zero(matrix, 1)) || leading\_coefficient(matrix, 1) != -1) {  return true;  }//ถ้า matrix มี 1 แถวแล้วแถว 1 เป็น 0 ทั้งแถวหรือมีตัวนำ 1  else {  return false;  }  }  for (int i = 2; i <= rows; i++) {  if ((leading\_coefficient(matrix, i - 1) != -1 || row\_is\_zero(matrix, i - 1)) && (leading\_coefficient(matrix, i) != -1 || row\_is\_zero(matrix, i))) {//ถ้า (matrix ตัวก่อนหน้ามีตัวนำ 1 หรือเป็น 0 ทั้งแถว) และ (matrix ตัวปัจจุบันมีตัวนำ 1 หรือเป็น 0 ทั้งแถว)  if (leading\_coefficient(matrix, i - 1) != -1 && leading\_coefficient(matrix, i) != -1) {  if (leading\_coefficient(matrix, i - 1) < leading\_coefficient(matrix, i)) {  } else {  return false;  }  } else if (leading\_coefficient(matrix, i - 1) != -1 && row\_is\_zero(matrix, i)) {  }//matrix ตัวก่อนหน้าต้องมี ตัวนำ 1 และ ตัวปัจจุบันต้องเป็น 0 ทั้งแถว  else if (row\_is\_zero(matrix, i - 1) && row\_is\_zero(matrix, i)) {  }///matrix ตัวก่อนหน้าต้องป็น 0 ทั้งแถวและ ตัวปัจจุบันต้องเป็น 0 ทั้งแถว  else {  return false;  }  } else {  return false;  }  }  return true;  }  public boolean check\_reduce\_row\_echelon\_matrix(float[][] matrix) {//เช็คว่าเป็น reduce row echelon matrix เปล่า  if (check\_row\_echelon\_matrix(matrix)) {//เช็คว่าเป็น rem เปล่าถ้าไม่เป็นก็ไม่มีทางเป็น rrem ได้  for (int i = 1; i <= rows; i++) {  if (leading\_coefficient(matrix, i) != -1) {//เลือกม่าเฉพาะแถวที่มีตัวนำ 1  for (int l = 1; l <= rows; l++) {  if (matrix[l - 1][leading\_coefficient(matrix, i) - 1] == 0) {  }//ในหลักของตัวนำ 1 ตัวที่ไม่ใช่ตัวนำ 1 ต้องเป็น 0 เท่านั้น ทั้งข้างบนข้างล่าง  else if (l == i) {  }//กัน for loop แล้วเจอตัวนำ 1 เพราะตัวนำ 1 ไม่ใช่ 0  else {  return false;  }  }  }  }  return true;  }  return false;  }  private void multiplying\_row\_by\_constant(float[][] matrix, int row, float constant) {  if (constant == 0) {  return;  } else {  for (int i = 0; i < columns; i++) {  if (matrix[row - 1][i] == 0) {  } else {  matrix[row - 1][i] = matrix[row - 1][i] \* constant;  }  }  }  }  private void adding\_row\_by\_row(float[][] matrix, int row\_1, int row\_2, float constant) {//row\_1 คือตัวตั้ง row\_2 คือตัวบวก  if (constant == 0) {  return;  } else {  for (int i = 0; i < columns; i++) {  matrix[row\_1 - 1][i] = matrix[row\_1 - 1][i] + (matrix[row\_2 - 1][i] \* constant);  }  }  }  private void switching\_two\_rows(float[][] matrix, int row\_1, int row\_2) {//สลับแถว  for (int i = 0; i < columns; i++) {  float tmp = matrix[row\_1 - 1][i];  matrix[row\_1 - 1][i] = matrix[row\_2 - 1][i];  matrix[row\_2 - 1][i] = tmp;  }  }  private JPanel matrix\_to\_Jpanel(float[][] matrix) {//เปลี่ยน matrix เป็น JPanel  JPanel main\_panel = new JPanel();  JPanel matrix\_panel = new JPanel();  JLabel[][] matrix\_Jlabel = new JLabel[rows][columns];  matrix\_panel.setLayout(new GridBagLayout());  GridBagConstraints c = new GridBagConstraints();  for (int i = 0; i < rows; i++) {  for (int l = 0; l < columns; l++) {  matrix\_Jlabel[i][l] = new JLabel();  matrix\_Jlabel[i][l].setText(matrix[i][l] + " ");  matrix\_Jlabel[i][l].setFont(new java.awt.Font("Tahoma", 1, 13));  c.gridx = l;  c.gridy = i;  matrix\_panel.add(matrix\_Jlabel[i][l], c);  }  }  main\_panel.add(matrix\_panel);  return main\_panel;  }  private JLabel set\_text\_to\_JLabel(String tmp) {  JLabel text\_Label = new JLabel();  text\_Label.setText(tmp);  text\_Label.setFont(new java.awt.Font("Tahoma", 1, 13));  return text\_Label;  }  public JPanel calulate\_rrem\_to\_JPanel(float[][] matrix) {//คำนวณ rrem และแปลงเป็น JPanel  JPanel panel = new JPanel();  panel.setLayout(new GridBagLayout());  GridBagConstraints c = new GridBagConstraints();  c.gridx = 1;  int leading\_coefficient\_position\_in\_row = 0;//คือ ค่าของแถวของตัวนำ 1 ที่คาดหวัง ใช้หาว่าตัวนำต้องอยู่ในแถวไหน  String tmp = "";  int JPanel\_sqsequence = 1;  for (int i = 0; i < columns; i++) {  for (int l = 0; l < rows; l++) {  if (leading\_coefficient\_position\_in\_row != -1) {//ใช้กันไม่ให้เกิดเหตุการณ์ leading\_coefficient\_position\_in\_row มีค่ามากกว่าจำนวนแถวสูงสุด  if (matrix[leading\_coefficient\_position\_in\_row][i] == 1) {  if (leading\_coefficient\_position\_in\_row != l && matrix[l][i] != 0) {//ถ้าในหลักนั้นมีตัวนำ 1 ให้เปลี่ยนตัวที่เหลือในหลักนั้นเป็น 0  tmp = "R" + (l + 1) + "+(" + (-matrix[l][i]) + ")R" + (leading\_coefficient\_position\_in\_row + 1) + "\n\n";  c.gridy = JPanel\_sqsequence;  adding\_row\_by\_row(matrix, l + 1, leading\_coefficient\_position\_in\_row + 1, -matrix[l][i]);  panel.add(matrix\_to\_Jpanel(matrix), c);  JPanel\_sqsequence++;  c.gridy = JPanel\_sqsequence;  panel.add(set\_text\_to\_JLabel(tmp), c);  JPanel\_sqsequence++;  //////////////////////////////สร้างเว้นวรรคด้วย JLabel  c.gridy = JPanel\_sqsequence;  panel.add(set\_text\_to\_JLabel(" "), c);  JPanel\_sqsequence++;  /////////////////////////////////  }  } else if (matrix[l][i] != 0 && leading\_coefficient\_position\_in\_row <= l) {//ใช้หาตัวนำ 1 โดยต้องอยู่ในแถวที่น้อยกว่าหรือเท่ากับตำนำ1ที่คาดหวัง  if (l != leading\_coefficient\_position\_in\_row) {//อย่างแรกสลับแถวปัจจุบันไปแถวที่ควรมีตัวนำ 1  tmp = "R" + (l + 1) + "↔R" + (leading\_coefficient\_position\_in\_row + 1) + "\n\n";  c.gridy = JPanel\_sqsequence;  switching\_two\_rows(matrix, l + 1, leading\_coefficient\_position\_in\_row + 1);  panel.add(matrix\_to\_Jpanel(matrix), c);  JPanel\_sqsequence++;  c.gridy = JPanel\_sqsequence;  panel.add(set\_text\_to\_JLabel(tmp), c);  JPanel\_sqsequence++;  //////////////////////////////สร้างเว้นวรรคด้วย JLabel  c.gridy = JPanel\_sqsequence;  panel.add(set\_text\_to\_JLabel(" "), c);  JPanel\_sqsequence++;  /////////////////////////////////  }  if (matrix[leading\_coefficient\_position\_in\_row][i] != 1) {//เปลี่ยนช่องที่อยู่ให้กลายเป็น 1 โดยการหารตัวมันเองทั้งแถว  tmp = "R" + (leading\_coefficient\_position\_in\_row + 1) + "/" + matrix[leading\_coefficient\_position\_in\_row][i] + "\n\n";  c.gridy = JPanel\_sqsequence;  multiplying\_row\_by\_constant(matrix, leading\_coefficient\_position\_in\_row + 1, 1 / matrix[leading\_coefficient\_position\_in\_row][i]);  panel.add(matrix\_to\_Jpanel(matrix), c);  JPanel\_sqsequence++;  c.gridy = JPanel\_sqsequence;  panel.add(set\_text\_to\_JLabel(tmp), c);  JPanel\_sqsequence++;  //////////////////////////////สร้างเว้นวรรคด้วย JLabel  c.gridy = JPanel\_sqsequence;  panel.add(set\_text\_to\_JLabel(" "), c);  JPanel\_sqsequence++;  /////////////////////////////////  }  l = -1;//ต้องเป็น -1 เพราะว่า อยากให้ l=0 เพื่อที่มันจะได้วน loop แต่พอจบ if มันจะ +1 เลยตัองเป็น -1  }  }  }  if (leading\_coefficient\_position\_in\_row == -1) {  }//ใช้กันไม่ให้เกิดเหตุการณ์ leading\_coefficient\_position\_in\_row มีค่ามากกว่าจำนวนแถวสูงสุด  else if (matrix[leading\_coefficient\_position\_in\_row][i] == 1) {  leading\_coefficient\_position\_in\_row++;  if (leading\_coefficient\_position\_in\_row == rows) {//ถ้า leading\_coefficient\_position\_in\_row มีค่าเท่ากับจำนวนแถวสูงสุดแล้วปรับให้เป็น -1 ซะ เพื่อจะไม่ต้องทำต่อ  leading\_coefficient\_position\_in\_row = -1;  }  }  }  for (int i = 0; i < columns; i++) {  for (int l = 0; l < rows; l++) {  if (matrix[l][i] == -0) {  matrix[l][i] = 0;  }  }  }  return panel;  }  public JPanel calulate\_rem\_to\_JPanel(float[][] matrix) {//คำนวณ rem และแปลงเป็น JPanel  JPanel panel = new JPanel();  panel.setLayout(new GridBagLayout());  GridBagConstraints c = new GridBagConstraints();  c.gridx = 1;  int leading\_coefficient\_position\_in\_row = 0;//คือ ค่าของแถวของตัวนำ 1 ที่คาดหวัง ใช้หาว่าตัวนำต้องอยู่ในแถวไหน  String tmp = "";  int JPanel\_sqsequence = 1;  for (int i = 0; i < columns; i++) {  for (int l = 0; l < rows; l++) {  if (leading\_coefficient\_position\_in\_row != -1) {//ใช้กันไม่ให้เกิดเหตุการณ์ leading\_coefficient\_position\_in\_row มีค่ามากกว่าจำนวนแถวสูงสุด  if (matrix[leading\_coefficient\_position\_in\_row][i] == 1) {  if (leading\_coefficient\_position\_in\_row != l && matrix[l][i] != 0 && leading\_coefficient\_position\_in\_row < l) {//ถ้าในหลักนั้นมีตัวนำ 1 ให้เปลี่ยนตัวที่เหลือในหลักนั้นเป็น 0  tmp = "R" + (l + 1) + "+(" + (-matrix[l][i]) + ")R" + (leading\_coefficient\_position\_in\_row + 1) + "\n\n";  c.gridy = JPanel\_sqsequence;  adding\_row\_by\_row(matrix, l + 1, leading\_coefficient\_position\_in\_row + 1, -matrix[l][i]);  panel.add(matrix\_to\_Jpanel(matrix), c);  JPanel\_sqsequence++;  c.gridy = JPanel\_sqsequence;  panel.add(set\_text\_to\_JLabel(tmp), c);  JPanel\_sqsequence++;  //////////////////////////////สร้างเว้นวรรคด้วย JLabel  c.gridy = JPanel\_sqsequence;  panel.add(set\_text\_to\_JLabel(" "), c);  JPanel\_sqsequence++;  /////////////////////////////////  }  } else if (matrix[l][i] != 0 && leading\_coefficient\_position\_in\_row <= l) {//ใช้หาตัวนำ 1 โดยต้องอยู่ในแถวที่น้อยกว่าหรือเท่ากับตำนำ1ที่คาดหวัง  if (l != leading\_coefficient\_position\_in\_row) {//อย่างแรกสลับแถวปัจจุบันไปแถวที่ควรมีตัวนำ 1  tmp = "R" + (l + 1) + "↔R" + (leading\_coefficient\_position\_in\_row + 1) + "\n\n";  c.gridy = JPanel\_sqsequence;  switching\_two\_rows(matrix, l + 1, leading\_coefficient\_position\_in\_row + 1);  panel.add(matrix\_to\_Jpanel(matrix), c);  JPanel\_sqsequence++;  c.gridy = JPanel\_sqsequence;  panel.add(set\_text\_to\_JLabel(tmp), c);  JPanel\_sqsequence++;  //////////////////////////////สร้างเว้นวรรคด้วย JLabel  c.gridy = JPanel\_sqsequence;  panel.add(set\_text\_to\_JLabel(" "), c);  JPanel\_sqsequence++;  /////////////////////////////////  }  if (matrix[leading\_coefficient\_position\_in\_row][i] != 1) {//เปลี่ยนช่องที่อยู่ให้กลายเป็น 1 โดยการหารตัวมันเองทั้งแถว  tmp = "R" + (leading\_coefficient\_position\_in\_row + 1) + "/" + matrix[leading\_coefficient\_position\_in\_row][i] + "\n\n";  c.gridy = JPanel\_sqsequence;  multiplying\_row\_by\_constant(matrix, leading\_coefficient\_position\_in\_row + 1, 1 / matrix[leading\_coefficient\_position\_in\_row][i]);  panel.add(matrix\_to\_Jpanel(matrix), c);  JPanel\_sqsequence++;  c.gridy = JPanel\_sqsequence;  panel.add(set\_text\_to\_JLabel(tmp), c);  JPanel\_sqsequence++;  //////////////////////////////สร้างเว้นวรรคด้วย JLabel  c.gridy = JPanel\_sqsequence;  panel.add(set\_text\_to\_JLabel(" "), c);  JPanel\_sqsequence++;  /////////////////////////////////  }  l = -1;//ต้องเป็น -1 เพราะว่า อยากให้ l=0 เพื่อที่มันจะได้วน loop แต่พอจบ if มันจะ +1 เลยตัองเป็น -1  }  }  }  if (leading\_coefficient\_position\_in\_row == -1) {  }//ใช้กันไม่ให้เกิดเหตุการณ์ leading\_coefficient\_position\_in\_row มีค่ามากกว่าจำนวนแถวสูงสุด  else if (matrix[leading\_coefficient\_position\_in\_row][i] == 1) {  leading\_coefficient\_position\_in\_row++;  if (leading\_coefficient\_position\_in\_row == rows) {//ถ้า leading\_coefficient\_position\_in\_row มีค่าเท่ากับจำนวนแถวสูงสุดแล้วปรับให้เป็น -1 ซะ เพื่อจะไม่ต้องทำต่อ  leading\_coefficient\_position\_in\_row = -1;  }  }  }  for (int i = 0; i < columns; i++) {  for (int l = 0; l < rows; l++) {  if (matrix[l][i] == -0) {  matrix[l][i] = 0;  }  }  }  return panel;  }  } |