**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря Сікорського”**

**Л А Б О Р А Т О Р Н А Р О Б О Т А № 1**

***ЗАГАЛЬНІ ПРИНЦИПИ ОРГАНІЗАЦІЇ ЕКСПЕРИМЕНТІВ З***

***ДОВІЛЬНИМИ ЗНАЧЕННЯМИ ФАКТОРІВ***

**З дисципліни:**

**«Теорія планування експерименту»**

Виконали:

Студенти ФІОТ

Бригади #3

Групи ІО – 53

Венгерак Арсеній

Возний Олександр

**Київ 2017**

**Лабораторна робота №1**

ЗАГАЛЬНІ ПРИНЦИПИ ОРГАНІЗАЦІЇ ЕКСПЕРИМЕНТІВ З ДОВІЛЬНИМИ ЗНАЧЕННЯМИ ФАКТОРІВ

**Мета завдання:**

Вивчити основні поняття, визначення, принципи теорії планування експерименту, на основі яких вивчити побудову формалізованих алгоритмів проведення експерименту і отримання формалізованої моделі об’єкта. Закріпити отримані знання практичним їх використанням при написанні програми, що реалізує завдання на лабораторну роботу.

Варіант завдання: 103

Критерій: max(Y)

1. **Лістинг:**

**Model.java**

**import** java.util.Random;  
  
**public class** Model {  
  
 **double leftLimit**;  
 **double rightLimit**;  
  
 **double**[] **factorArray**;  
 **double**[][] **matrixPlan**;  
 **double**[] **standard**;  
 **double**[] **interval**;  
 **int optimum**;  
  
 Model(**double** a, **double** b, **double** a0, **double** a1, **double** a2, **double** a3) {  
 **leftLimit** = a;  
 **rightLimit** = b;  
 **factorArray** = **new double**[4];  
 **factorArray**[0] = a0;  
 **factorArray**[1] = a1;  
 **factorArray**[2] = a2;  
 **factorArray**[3] = a3;  
 **this**.generate();  
  
  
 }  
  
 **public double**[] getRow(**int** i) {  
 **double**[] result = {**matrixPlan**[i][0], **matrixPlan**[i][1], **matrixPlan**[i][2]};  
 **return** result;  
 }  
  
 **public double**[][] getMatrixPlan() {  
 **return matrixPlan**;  
 }  
  
 **public double**[] getStandard() {  
 **return standard**;  
 }  
  
 **public double**[] getInterval() {  
 **return interval**;  
 }  
  
 **public int** getOptimum() {  
 **return optimum**;  
 }  
  
 **void** generate() {  
 Random rand = **new** Random();  
 **matrixPlan** = **new double**[8][7];  
 **for** (**int** i = 0; i < 8; i++) {  
 **matrixPlan**[i][3] = **factorArray**[0];  
 **for** (**int** j = 0; j < 3; j++) {  
 **matrixPlan**[i][j] = **leftLimit** + rand.nextDouble() \* (**rightLimit** - **leftLimit**);  
 **matrixPlan**[i][3] += **factorArray**[j + 1] \* **matrixPlan**[i][j];  
 }  
 }  
  
 **double**[] leftBorder = **new double**[3];  
 **double**[] rightBorder = **new double**[3];  
  
 **for** (**int** j = 0; j < 3; j++) {  
 leftBorder[j] = **matrixPlan**[0][j];  
 rightBorder[j] = **matrixPlan**[0][j];  
 }  
  
 **for** (**int** i = 1; i < 8; i++) {  
 **for** (**int** j = 0; j < 3; j++) {  
 **if** (leftBorder[j] > **matrixPlan**[i][j]) leftBorder[j] = **matrixPlan**[i][j];  
 **if** (rightBorder[j] < **matrixPlan**[i][j]) rightBorder[j] = **matrixPlan**[i][j];  
 }  
 }  
  
 **standard** = **new double**[4];  
 **interval** = **new double**[3];  
 **for** (**int** j = 0; j < 3; j++) {  
 **standard**[j] = (rightBorder[j] + leftBorder[j]) / 2;  
 **interval**[j] = (rightBorder[j] - leftBorder[j]) / 2;  
 **for** (**int** i = 0; i < 8; i++) {  
 **matrixPlan**[i][j + 4] = (**matrixPlan**[i][j] - **standard**[j]) / **interval**[j];  
 }  
 }  
 **standard**[3] = **factorArray**[0];  
 **for** (**int** j = 0; j < 3; j++) {  
 **standard**[3] += **factorArray**[j + 1] \* **standard**[j];  
 }  
 **optimum** = 0;  
 **for** (**int** i = 1; i < 8; i++) {  
 **if** (**matrixPlan**[i][3] > **matrixPlan**[**optimum**][3]) **optimum** = i;  
 }  
  
  
 }  
}

**Main.java**

**public class** Main {  
 **public static void** main(String[] args) {  
 Viewer dr = **new** Viewer();  
 }  
}

**Viewer.java**

**import** javax.swing.\*;  
**import** javax.swing.table.TableModel;  
**import** java.awt.\*;  
**import** java.awt.event.ActionEvent;  
**import** java.awt.event.ActionListener;  
**import** java.util.Formatter;  
**import** java.util.LinkedList;  
  
**public class** Viewer {  
  
 **public static int** *FRAME\_WIDTH* = 1400;  
 **public static int** *FRAME\_HEIGHT* = 300;  
  
 JPanel **panel** = **new** JPanel();  
 GridLayout **g** = **new** GridLayout(1, 2);  
 JPanel **panel1** = **new** JPanel();  
 GridLayout **g1** = **new** GridLayout(2, 1);  
 JPanel **panel2** = **new** JPanel();  
 JPanel **panel1a** = **new** JPanel();  
 GridLayout **g1a** = **new** GridLayout(3, 2);  
 JPanel **panel1b** = **new** JPanel();  
 GridLayout **g1b** = **new** GridLayout(3, 2);  
 JPanel **panel1a1** = **new** JPanel();  
 GridLayout **g1a1** = **new** GridLayout(2, 2);  
 JPanel **panel1a2** = **new** JPanel();  
 GridLayout **g1a2** = **new** GridLayout(2, 2);  
 JPanel **panel1a3** = **new** JPanel();  
 GridLayout **g1a3** = **new** GridLayout(2, 2);  
  
 JPanel **panel1b1** = **new** JPanel();  
 JPanel **panel1b2** = **new** JPanel();  
 JPanel **panel1b3** = **new** JPanel();  
 JPanel **panel1b4** = **new** JPanel();  
  
 JPanel **panel6** = **new** JPanel();  
 GridLayout **g6** = **new** GridLayout(1, 3);  
 JPanel **panel7** = **new** JPanel();  
  
 JLabel **label01** = **new** JLabel(**"Limits of figures"**);  
 JLabel **label02** = **new** JLabel(**"Factors of regressions"**);  
 JButton **task03** = **new** JButton(**"Generate"**);  
  
 JLabel **label04** = **new** JLabel(**"Left limit"**);  
 JTextField **field05** = **new** JTextField(10);  
 JLabel **label06** = **new** JLabel(**"Right limit"**);  
 JTextField **field07** = **new** JTextField(10);  
 JLabel **label08** = **new** JLabel(**"Factor a0"**);  
 JTextField **field09** = **new** JTextField(10);  
 JLabel **label10** = **new** JLabel(**"Factor a1"**);  
 JTextField **field11** = **new** JTextField(10);  
 JLabel **label12** = **new** JLabel(**"Factor a2"**);  
 JTextField **field13** = **new** JTextField(10);  
 JLabel **label14** = **new** JLabel(**"Factor a3"**);  
 JTextField **field15** = **new** JTextField(10);  
  
 JLabel **label16** = **new** JLabel(**"Standard"**);  
 JLabel **label17** = **new** JLabel(**"Interval"**);  
 JLabel **label18** = **new** JLabel(**"Optimum"**);  
 JLabel **label19** = **new** JLabel(**"Criterion: max value of Yi"**);  
  
 JTable **table1** = **new** JTable(9, 8);  
 JTable **table2** = **new** JTable(2, 4);  
 JTable **table3** = **new** JTable(2, 3);  
 JTable **table4** = **new** JTable(2, 4);  
  
 **public** Viewer() {  
  
  
 JFrame frame = **new** JFrame();  
 frame.setTitle(**"TPE: first lab work"**);  
 frame.setSize(*FRAME\_WIDTH*, *FRAME\_HEIGHT*);  
  
  
 **panel1a1**.setLayout(**g1a1**);  
 **panel1a1**.add(**label04**);  
 **panel1a1**.add(**field05**);  
 **panel1a1**.add(**label06**);  
 **panel1a1**.add(**field07**);  
  
 **panel1a2**.setLayout(**g1a2**);  
 **panel1a2**.add(**label08**);  
 **panel1a2**.add(**field09**);  
 **panel1a2**.add(**label10**);  
 **panel1a2**.add(**field11**);  
  
 **panel1a3**.setLayout(**g1a3**);  
 **panel1a3**.add(**label12**);  
 **panel1a3**.add(**field13**);  
 **panel1a3**.add(**label14**);  
 **panel1a3**.add(**field15**);  
  
 **panel6**.setLayout(**g6**);  
  
 **panel6**.add(**task03**);  
 **panel6**.add(**label19**);  
  
 **panel1a**.setLayout(**g1a**);  
 **panel1a**.add(**label01**);  
 **panel1a**.add(**panel1a1**);  
 **panel1a**.add(**label02**);  
 **panel1a**.add(**panel1a2**);  
 **panel1a**.add(**panel6**);  
 **panel1a**.add(**panel1a3**);  
  
  
 TableModel model1 = **table1**.getModel();  
 **for** (**int** i = 0; i < 9; i++)  
 **table1**.setRowHeight(i, 15);  
 **for** (**int** i = 1; i < 9; i++)  
 model1.setValueAt(i, i, 0);  
 model1.setValueAt(**"N"**, 0, 0);  
 model1.setValueAt(**"X1"**, 0, 1);  
 model1.setValueAt(**"X2"**, 0, 2);  
 model1.setValueAt(**"X3"**, 0, 3);  
 model1.setValueAt(**"Y"**, 0, 4);  
 model1.setValueAt(**"Xn1"**, 0, 5);  
 model1.setValueAt(**"Xn2"**, 0, 6);  
 model1.setValueAt(**"Xn3"**, 0, 7);  
  
 TableModel model2 = **table2**.getModel();  
 model2.setValueAt(**"X01"**, 0, 0);  
 model2.setValueAt(**"X02"**, 0, 1);  
 model2.setValueAt(**"X03"**, 0, 2);  
 model2.setValueAt(**"Y0"**, 0, 3);  
  
 TableModel model3 = **table3**.getModel();  
 model3.setValueAt(**"dX1"**, 0, 0);  
 model3.setValueAt(**"dX2"**, 0, 1);  
 model3.setValueAt(**"dX3"**, 0, 2);  
  
 TableModel model4 = **table4**.getModel();  
 model4.setValueAt(**"X1opt"**, 0, 0);  
 model4.setValueAt(**"X2opt"**, 0, 1);  
 model4.setValueAt(**"X3opt"**, 0, 2);  
 model4.setValueAt(**"Yopt"**, 0, 3);  
  
 **panel1b1**.add(**table2**);  
 **panel1b2**.add(**table3**);  
 **panel1b3**.add(**table4**);  
  
 **panel1b**.setLayout(**g1b**);  
 **panel1b**.add(**label16**);  
 **panel1b**.add(**panel1b1**);  
 **panel1b**.add(**label17**);  
 **panel1b**.add(**panel1b2**);  
 **panel1b**.add(**label18**);  
 **panel1b**.add(**panel1b3**);  
 *// panel1b.add(label19);  
 // panel1b.add(panel1b4);* **panel1**.setLayout(**g1**);  
 **panel1**.add(**panel1a**);  
 **panel1**.add(**panel1b**);  
  
 **panel2**.add(**table1**);  
  
 **panel**.setLayout(**g**);  
 **panel**.add(**panel1**);  
 **panel**.add(**panel2**);  
  
  
 **task03**.addActionListener(**new** ActionListener() {  
 **public void** actionPerformed(ActionEvent ae) {  
 **for** (**int** i = 0; i < 9; i++)  
 **table1**.setRowHeight(i, 15);  
 **double** a = Double.*parseDouble*(**field05**.getText());  
 **double** b = Double.*parseDouble*(**field07**.getText());  
 **double** a0 = Double.*parseDouble*(**field09**.getText());  
 **double** a1 = Double.*parseDouble*(**field11**.getText());  
 **double** a2 = Double.*parseDouble*(**field13**.getText());  
 **double** a3 = Double.*parseDouble*(**field15**.getText());  
 Model model = **new** Model(a, b, a0, a1, a2, a3);  
 **double** matrix[][] = model.getMatrixPlan();  
 **for** (**int** i = 0; i < 8; i++)  
 **for** (**int** j = 0; j < 7; j++)  
 model1.setValueAt(matrix[i][j], i + 1, j + 1);  
  
 **double** standard[] = model.getStandard();  
  
 **for** (**int** j = 0; j < 4; j++) {  
 model2.setValueAt(standard[j], 1, j);  
 }  
  
  
 **double** interval[] = model.getInterval();  
 **for** (**int** j = 0; j < 3; j++) {  
 model3.setValueAt(interval[j], 1, j);  
 }  
  
 **for** (**int** j = 0; j < 4; j++) {  
 model4.setValueAt(matrix[model.getOptimum()][j], 1, j);  
 }  
 **table1**.setRowHeight(model.getOptimum() + 1, 45);  
  
  
 }  
 });  
  
 frame.add(**panel**);  
 frame.setVisible(**true**);  
 }  
}

1. **Висновок**

В ході данної лабораторної роботи було вивчено основні поняття, визначення, принципи теорії планування експерименту, на основі яких вивчити побудову формалізованих алгоритмів проведення експерименту і отримання формалізованої моделі об’єкта. Закріпити отримані знання практичним їх використанням при написанні програми, що реалізує завдання на лабораторну роботу.