

ДОДАТОК А

Згенерований звіт моделі СКМ в середовищі Enterprise Architect

Table of Contents

<i>bpm diagram</i>	2
<i>Визначення концептів diagram</i>	3
<i>Створення концептів diagram</i>	4
<i>Формування експертної групи diagram</i>	5
use case diagram	6
Ієрархія екранних форм diagram	7
Склад системи diagram	8
вимоги diagram	9
класи diagram	10

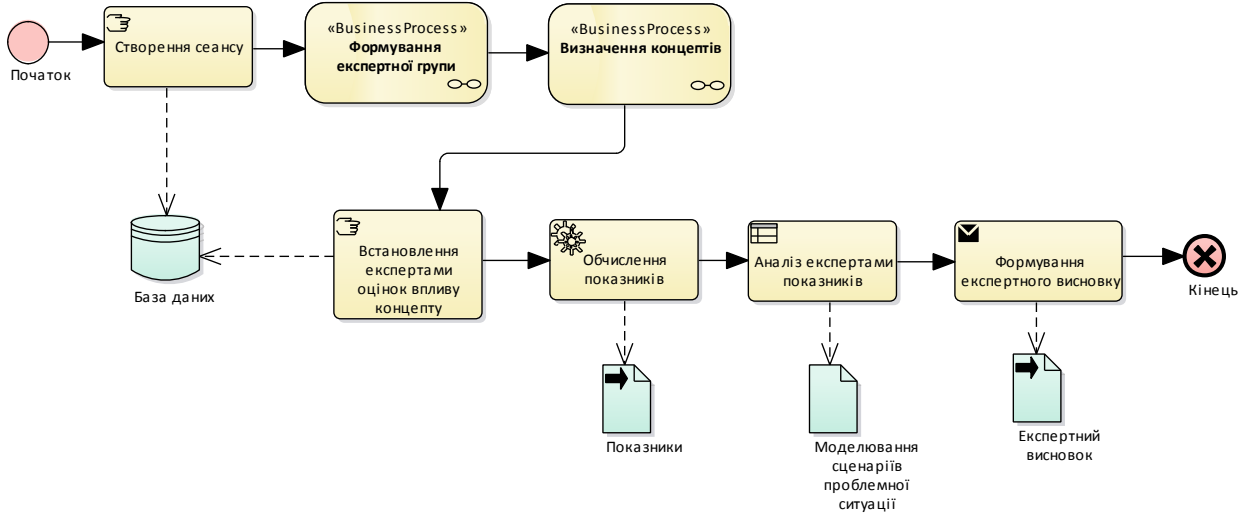
bpm diagram

Business Process diagram in package 'bpm'

bpm

Version 1.0

Profii created on 17.05.2020. Last modified 18.05.2020



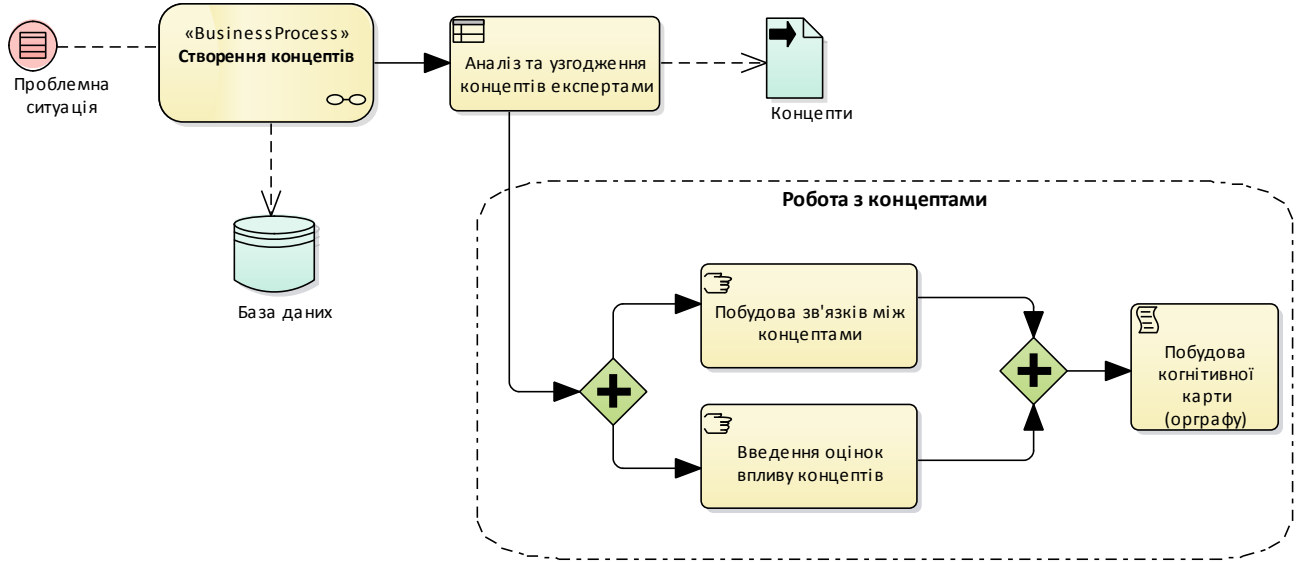
bpm

Визначення концептів diagram

Business Process diagram in package 'bpm'

Визначення концептів
Version 1.0

Profi created on 17.05.2020. Last modified 18.05.2020



Визначення концептів

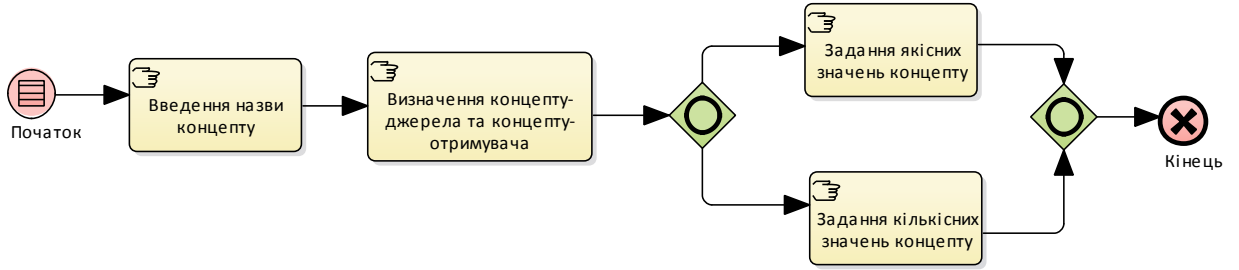
Створення концептів *diagram*

Business Process diagram in package 'bpm'

Створення концептів

Version 1.0

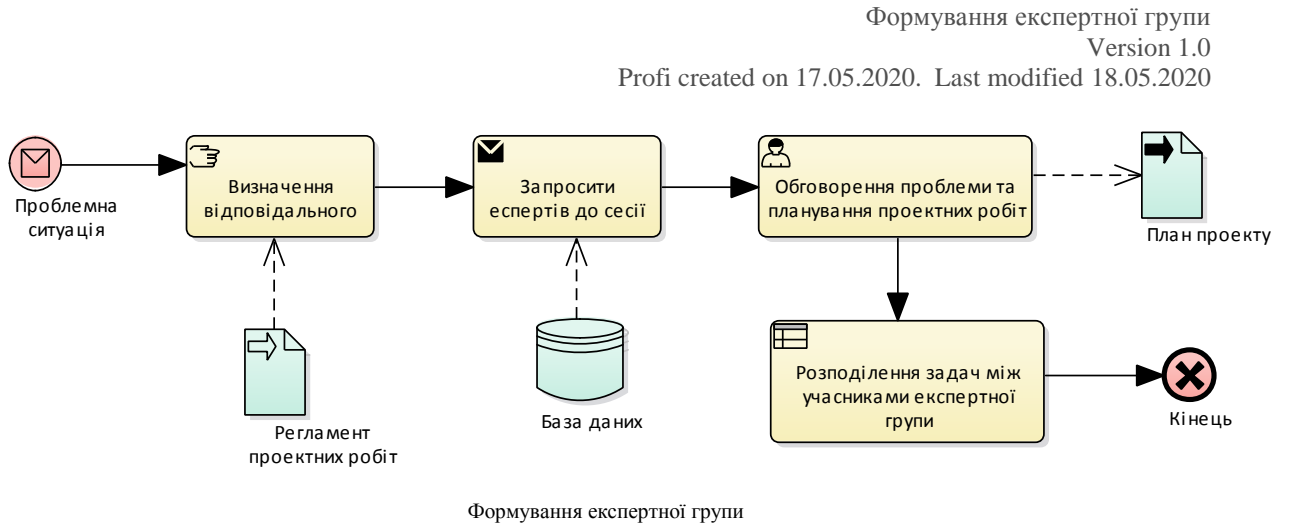
Profi created on 17.05.2020. Last modified 18.05.2020



Створення концептів

Формування експертної групи diagram

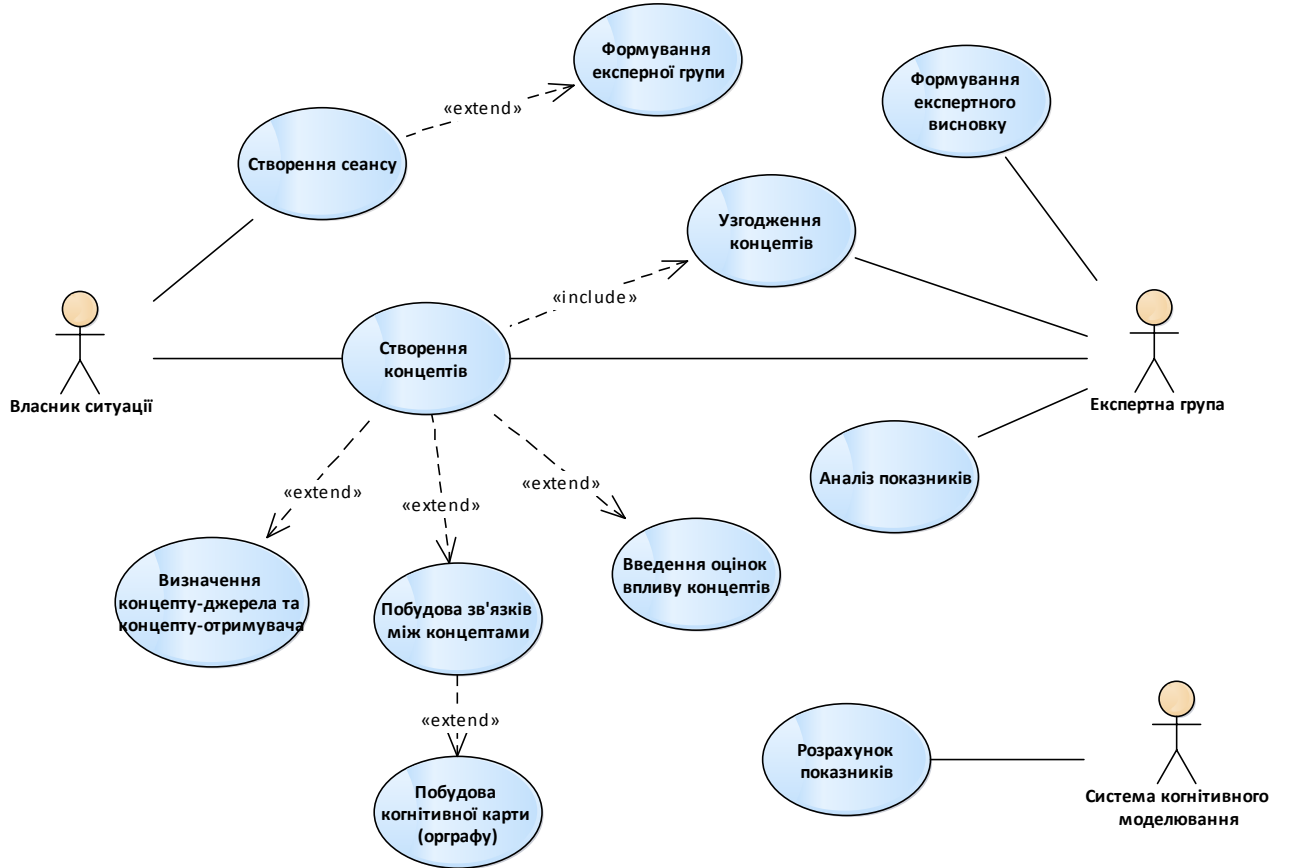
Business Process diagram in package 'bpm'



use case diagram

Use Case diagram in package 'use case'

use case
Version 1.0
Profi created on 17.05.2020. Last modified 18.05.2020



use case

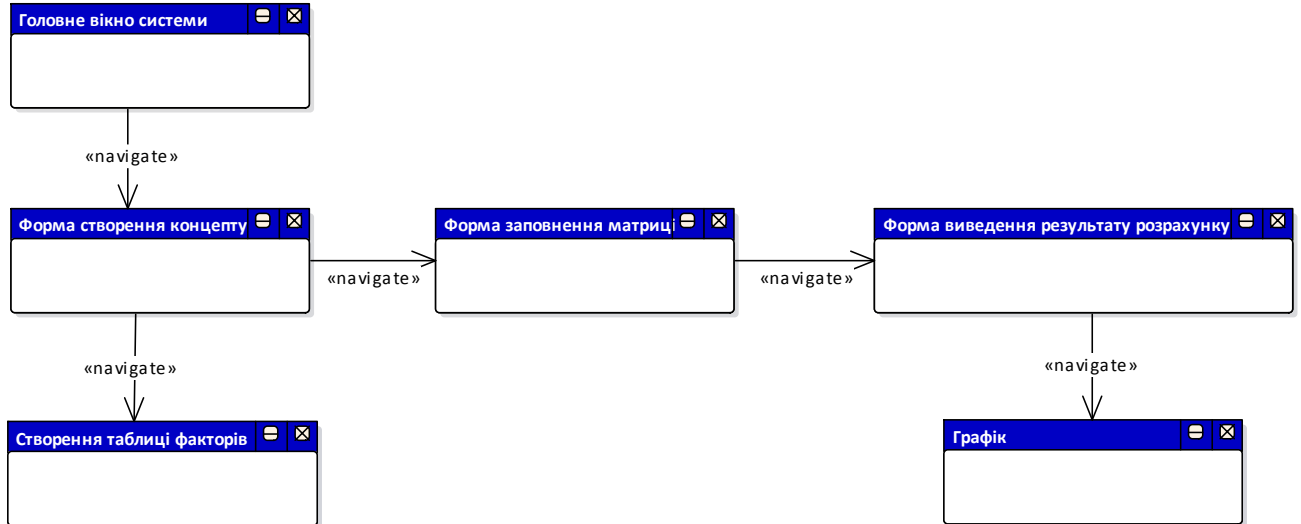
Ієрархія екранних форм diagram

Custom diagram in package 'Ієрархія екранних форм'

Ієрархія екранних форм

Version 1.0

Profi created on 18.05.2020. Last modified 18.05.2020



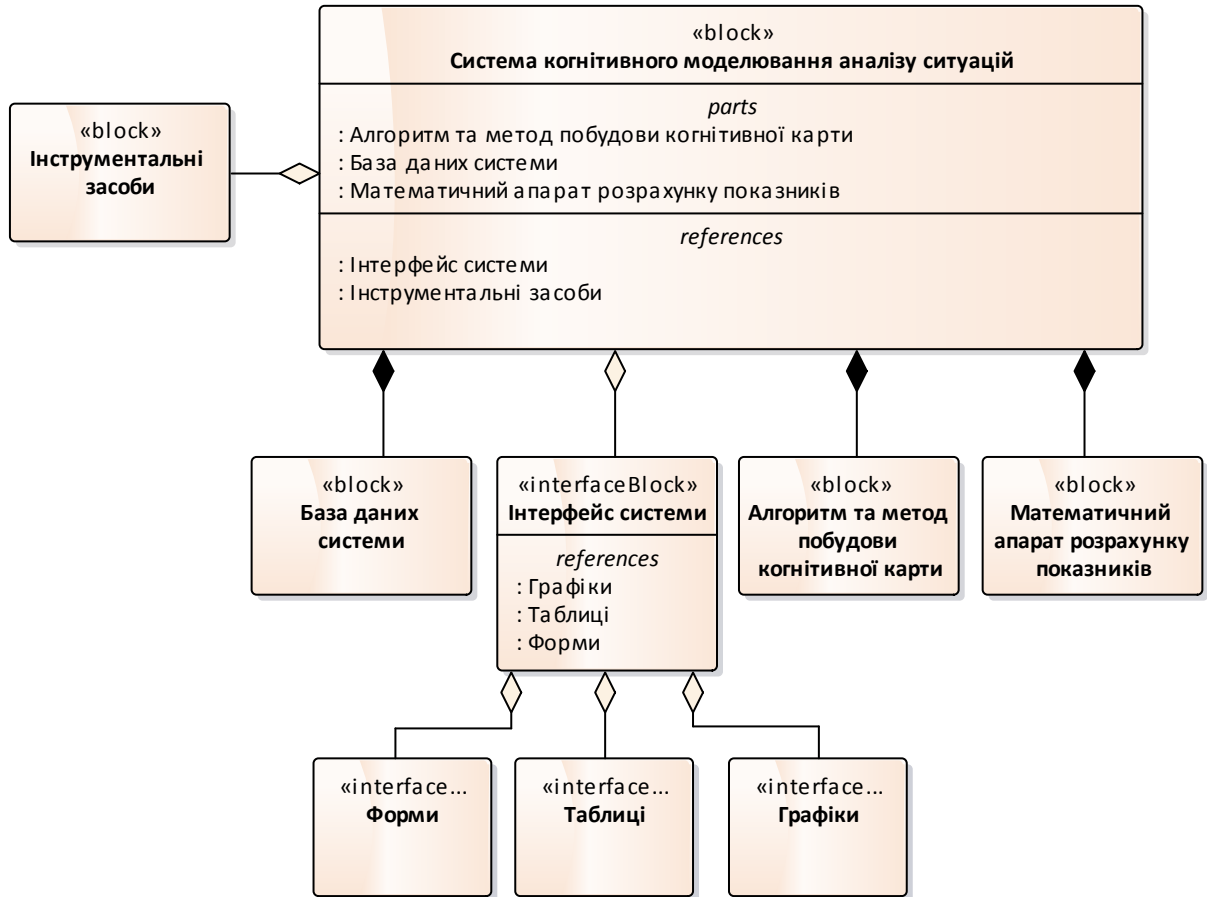
Ієрархія екранних форм

Склад системи diagram

SysML Block Definition diagram in package 'Склад системи'

Склад системи
Version 1.0

Profi created on 18.05.2020. Last modified 18.05.2020



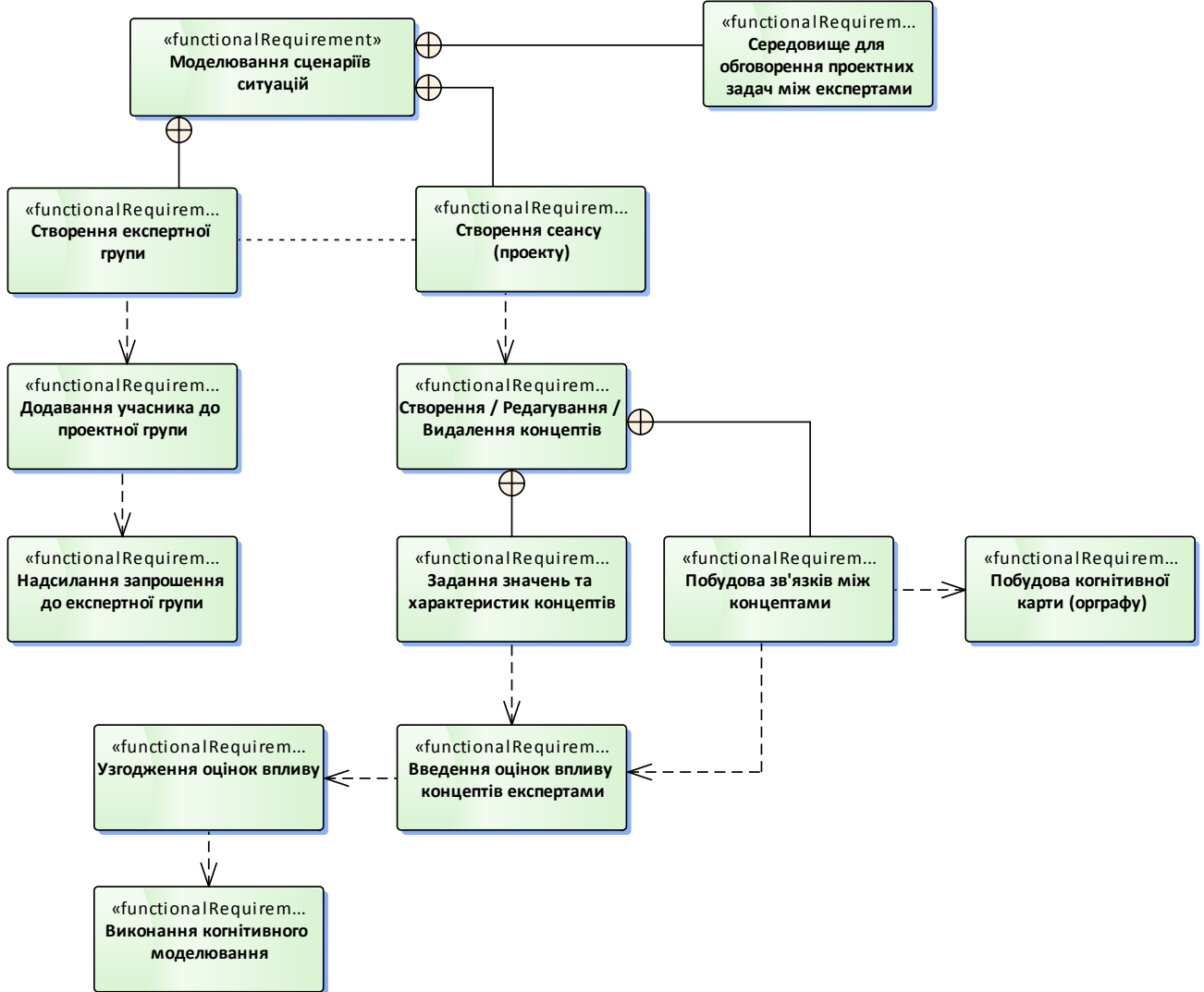
Склад системи

вимоги diagram

SysML Requirements diagram in package 'вимогу'

ВИМОГИ
Version 1.0

Profi created on 17.05.2020. Last modified 19.05.2020



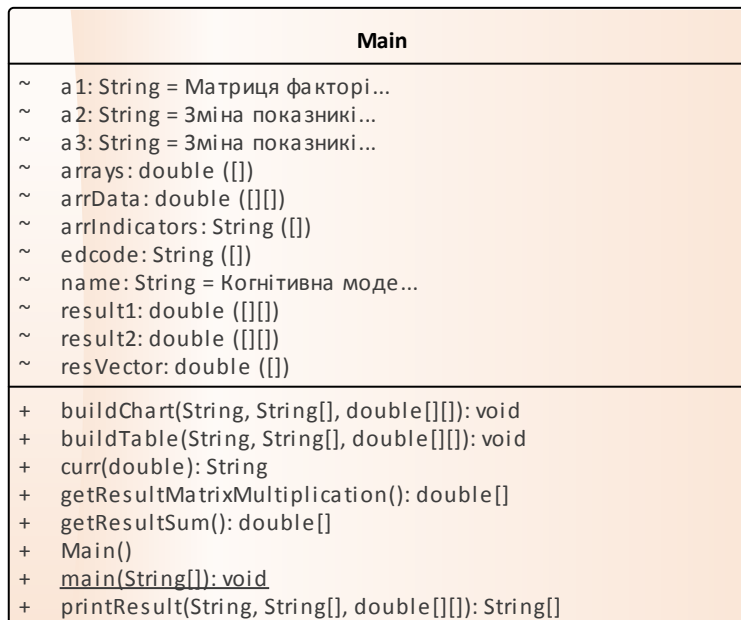
ВИМОГИ

класи diagram

Class diagram in package 'класи'

класи
Version 1.0

Profi created on 18.05.2020. Last modified 18.05.2020



класи

ДОДАТОК Б

Згенерований скрипт БД для СКМ

```

BUSE [master]
GO
/***** Object: Database [Cognitive_Maps]    Script Date: 25.05.2020 6:38:16
*****/
CREATE DATABASE [Cognitive_Maps]
    CONTAINMENT = NONE
    ON PRIMARY
    ( NAME = N'Cognitive_Maps', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\Cognitive_Maps.mdf' , SIZE = 8192KB , MAXSIZE
= UNLIMITED, FILEGROWTH = 65536KB )
    LOG ON
    ( NAME = N'Cognitive_Maps_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\Cognitive_Maps_log.ldf' , SIZE = 8192KB ,
MAXSIZE = 2048GB , FILEGROWTH = 65536KB )
GO
ALTER DATABASE [Cognitive_Maps] SET COMPATIBILITY_LEVEL = 140
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [Cognitive_Maps].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [Cognitive_Maps] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [Cognitive_Maps] SET ANSI_NULLS OFF
GO
ALTER DATABASE [Cognitive_Maps] SET ANSI_PADDING OFF
GO
ALTER DATABASE [Cognitive_Maps] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [Cognitive_Maps] SET ARITHABORT OFF
GO
ALTER DATABASE [Cognitive_Maps] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [Cognitive_Maps] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [Cognitive_Maps] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [Cognitive_Maps] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [Cognitive_Maps] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [Cognitive_Maps] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [Cognitive_Maps] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [Cognitive_Maps] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [Cognitive_Maps] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [Cognitive_Maps] SET DISABLE_BROKER
GO
ALTER DATABASE [Cognitive_Maps] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [Cognitive_Maps] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [Cognitive_Maps] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [Cognitive_Maps] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [Cognitive_Maps] SET PARAMETERIZATION SIMPLE
GO

```

```

ALTER DATABASE [Cognitive_Maps] SET READ_COMMITTED_SNAPSHOT OFF
GO
ALTER DATABASE [Cognitive_Maps] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [Cognitive_Maps] SET RECOVERY SIMPLE
GO
ALTER DATABASE [Cognitive_Maps] SET MULTI_USER
GO
ALTER DATABASE [Cognitive_Maps] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [Cognitive_Maps] SET DB_CHAINING OFF
GO
ALTER DATABASE [Cognitive_Maps] SET FILESTREAM( NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [Cognitive_Maps] SET TARGET_RECOVERY_TIME = 60 SECONDS
GO
ALTER DATABASE [Cognitive_Maps] SET DELAYED_DURABILITY = DISABLED
GO
ALTER DATABASE [Cognitive_Maps] SET QUERY_STORE = OFF
GO
USE [Cognitive_Maps]
GO
/***** Object: Table [dbo].[Assessments]    Script Date: 25.05.2020 6:38:18
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Assessments](
    [id_assessment] [int] NOT NULL,
    [Name_assessment] [char](20) NOT NULL,
    [Description__assessment] [nvarchar](100) NULL,
    CONSTRAINT [PK_Assessments] PRIMARY KEY CLUSTERED
(
    [id_assessment] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Experts]    Script Date: 25.05.2020 6:38:19 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Experts](
    [Id_expert] [int] NOT NULL,
    [PIB_expert] [char](30) NULL,
    [id_Posision] [int] NULL,
    [id_departament] [int] NULL,
    CONSTRAINT [PK_Experts] PRIMARY KEY CLUSTERED
(
    [Id_expert] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Koncepts]    Script Date: 25.05.2020 6:38:20 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Koncepts](

```

```

        [id_Konzept] [int] NOT NULL,
        [Name_konzept] [char](20) NOT NULL,
        [Full_Name_Konc] [nchar](100) NULL,
    CONSTRAINT [PK_Koncepts] PRIMARY KEY CLUSTERED
(
    [id_Konzept] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Session]    Script Date: 25.05.2020 6:38:20 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Session](
    [id_seans] [int] IDENTITY(1,1) NOT NULL,
    [Date_seans] [datetime] NOT NULL,
    [id_leader] [int] NOT NULL,
    [Description_s] [nchar](100) NULL,
    CONSTRAINT [PK_Session] PRIMARY KEY CLUSTERED
(
    [id_seans] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Session_assessment]    Script Date: 25.05.2020
6:38:21 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Session_assessment](
    [id_seans] [int] NOT NULL,
    [id_assessment] [int] NOT NULL,
    [Assessment] [numeric](18, 2) NOT NULL,
    [id_Konzept_input] [int] NULL,
    [id_Konzept_output] [int] NULL
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Team_session]    Script Date: 25.05.2020 6:38:22
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Team_session](
    [Id_seans] [int] NOT NULL,
    [Id_expert] [int] NOT NULL,
    [Description] [nchar](100) NULL,
    CONSTRAINT [PK_Team_session] PRIMARY KEY CLUSTERED
(
    [Id_seans] ASC,
    [Id_expert] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Weigth_Experts]    Script Date: 25.05.2020 6:38:23
*****/
SET ANSI_NULLS ON

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Weigth_Experts](
    [id_seans] [int] NOT NULL,
    [id_Konzept_input] [int] NULL,
    [id_Konzept_output] [int] NULL,
    [Id_expert] [int] NULL,
    [Weigth_expert] [numeric](8, 2) NULL
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Weigth_Koncepts]    Script Date: 25.05.2020 6:38:23
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Weigth_Koncepts](
    [id_seans] [int] NOT NULL,
    [id_Konzept_input] [int] NOT NULL,
    [id_Konzept_output] [int] NOT NULL,
    [Weigth] [numeric](8, 2) NULL,
    CONSTRAINT [PK_Weigth_Koncepts] PRIMARY KEY CLUSTERED
(
    [id_seans] ASC,
    [id_Konzept_input] ASC,
    [id_Konzept_output] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
ALTER TABLE [dbo].[Session] ADD CONSTRAINT [DF_Session_Date_seans] DEFAULT
(getdate()) FOR [Date_seans]
GO
ALTER TABLE [dbo].[Session] WITH CHECK ADD CONSTRAINT [FK_Session_Experts]
FOREIGN KEY([id_leader])
REFERENCES [dbo].[Experts] ([Id_expert])
GO
ALTER TABLE [dbo].[Session] CHECK CONSTRAINT [FK_Session_Experts]
GO
ALTER TABLE [dbo].[Session_assessment] WITH CHECK ADD CONSTRAINT
[FK_Session_assessment_Assessments] FOREIGN KEY([id_assessment])
REFERENCES [dbo].[Assessments] ([id_assessment])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Session_assessment] CHECK CONSTRAINT
[FK_Session_assessment_Assessments]
GO
ALTER TABLE [dbo].[Session_assessment] WITH CHECK ADD CONSTRAINT
[FK_Session_assessment_Weigth_Koncepts] FOREIGN KEY([id_seans],
[id_Konzept_input], [id_Konzept_output])
REFERENCES [dbo].[Weigth_Koncepts] ([id_seans], [id_Konzept_input],
[id_Konzept_output])
GO
ALTER TABLE [dbo].[Session_assessment] CHECK CONSTRAINT
[FK_Session_assessment_Weigth_Koncepts]
GO
ALTER TABLE [dbo].[Team_session] WITH CHECK ADD CONSTRAINT
[FK_Team_session_Experts] FOREIGN KEY([Id_expert])
REFERENCES [dbo].[Experts] ([Id_expert])
ON UPDATE CASCADE

```

```

ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Team_session] CHECK CONSTRAINT [FK_Team_session_Experts]
GO
ALTER TABLE [dbo].[Team_session] WITH CHECK ADD CONSTRAINT
[FK_Team_session_Session] FOREIGN KEY([Id_seans])
REFERENCES [dbo].[Session] ([id_seans])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Team_session] CHECK CONSTRAINT [FK_Team_session_Session]
GO
ALTER TABLE [dbo].[Weigth_Experts] WITH CHECK ADD CONSTRAINT
[FK_Weigth_Experts_Team_session] FOREIGN KEY([id_seans], [Id_expert])
REFERENCES [dbo].[Team_session] ([Id_seans], [Id_expert])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Weigth_Experts] CHECK CONSTRAINT
[FK_Weigth_Experts_Team_session]
GO
ALTER TABLE [dbo].[Weigth_Experts] WITH CHECK ADD CONSTRAINT
[FK_Weigth_Experts_Weigth_Koncepts] FOREIGN KEY([id_seans], [id_Konzept_input],
[id_Konzept_output])
REFERENCES [dbo].[Weigth_Koncepts] ([id_seans], [id_Konzept_input],
[id_Konzept_output])
GO
ALTER TABLE [dbo].[Weigth_Experts] CHECK CONSTRAINT
[FK_Weigth_Experts_Weigth_Koncepts]
GO
ALTER TABLE [dbo].[Weigth_Koncepts] WITH CHECK ADD CONSTRAINT
[FK_Weigth_Koncepts_Koncepts] FOREIGN KEY([id_Konzept_input])
REFERENCES [dbo].[Koncepts] ([id_Konzept])
GO
ALTER TABLE [dbo].[Weigth_Koncepts] CHECK CONSTRAINT [FK_Weigth_Koncepts_Koncepts]
GO
ALTER TABLE [dbo].[Weigth_Koncepts] WITH CHECK ADD CONSTRAINT
[FK_Weigth_Koncepts_Koncepts1] FOREIGN KEY([id_Konzept_output])
REFERENCES [dbo].[Koncepts] ([id_Konzept])
GO
ALTER TABLE [dbo].[Weigth_Koncepts] CHECK CONSTRAINT
[FK_Weigth_Koncepts_Koncepts1]
GO
ALTER TABLE [dbo].[Weigth_Koncepts] WITH CHECK ADD CONSTRAINT
[FK_Weigth_Koncepts_Session] FOREIGN KEY([id_seans])
REFERENCES [dbo].[Session] ([id_seans])
ON UPDATE CASCADE
ON DELETE CASCADE
GO
ALTER TABLE [dbo].[Weigth_Koncepts] CHECK CONSTRAINT [FK_Weigth_Koncepts_Session]
GO
USE [master]
GO
ALTER DATABASE [Cognitive_Maps] SET READ_WRITE
GO

```

ДОДАТОК В

Код програми «Когнітивна система»

Додаток В.1. Код класу “Фабрика методів роботи з інтерфейсом”

```

/**
 * Когнітивна модель (Аналіз факторів при онкозахворюванні)
 * @author Помазун Оксана Миколаївна
 */

public class NewFactory extends JFrame {

    boolean logic1 = false;
    JDialog dialog1;
    JButton buttonNew;

    public static final long serialVersionUID = 1L;
    // Модель даних таблиці
    public DefaultTableModel tableModel;
    public DefaultTableModel tableModelNew;
    public DefaultTableModel tableModelNew1;
    public JTable table1;
    public JTable table2;
    public JTable table3;
    public JTable table4;
    public JLabel jLabel;
    public JTextField jTextField;
    int Row;

    JComboBox comboBox;

    private JLabel jLabel2;

    // Дані для таблиці
    private String[][] array = new String[][]{
        {"Фактор 1", "F1", "Назва", "+"},
        {"Фактор 2", "F2", "Назва", "+"},
        {"Фактор 3", "F3", "Назва", "+"},
        {"Фактор 4", "F4", "Назва", "+"}
    };

    // Заголовок стовпців
    private String[] columnHeader = new String[]{"Номер фактора", "Код", "Назва",
"Вплив"};

    public NewFactory() {

        super("Когнітивна модель \"Аналіз факторів при онкозахворюванні\"");
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        // Створення стандартної моделі

```

```

tableModel = new DefaultTableModel();

// tableModelNew = new DefaultTableModel();
// Визначення стовпців
tableModel.setColumnIdentifiers(columnsHeader);
// Наповнення моделі даними
for (int i = 0; i < array.length; i++) {
    tableModel.addRow(array[i]);
}
// Створення таблиці на підставі моделі даних
table1 = new JTable(tableModel);
// Створення кнопки додавання рядка таблиці
JButton add = new JButton("Додати");
table1.setSelectionInterval(table1.getRowCount() - 1, table1.getColumnCount() - 1);

String[] items = new String[]{"Негативний (-)", "Позитивний (+)"};
comboBox = new JComboBox(items);

DefaultCellEditor editor = new DefaultCellEditor(comboBox);
// Визначення редактора комірок для колонки
table1.getColumnModel().getColumn(3).setCellEditor(editor);

add.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Номер виділеного рядка
        int idx = table1.getSelectedRow();
        // Вставка нового рядка після виділеного
        tableModel.insertRow(idx + 1, new String[]{"Фактор " +
String.valueOf(table1.getRowCount() + 1), "F" + String.valueOf(table1.getRowCount() + 1),
"Назва", "Інгредієнт"});
        getTableData(table1);
    }
});

// Створення кнопки видалення рядка таблиці

final JButton remove = new JButton("Знищити");
remove.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Номер виділеного рядка
        int idx = table1.getSelectedRow();
        // Видалення виділеного рядка
        tableModel.removeRow(idx);
    }
});

dialog1 = new JDialog();
dialog1.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);
dialog1.setSize(550, 550);
dialog1.setVisible(false);

JButton getMatrix = new JButton("Заповнити матрицю");

```

```

getMatrix.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        table2 = new JTable(7, 5);
        table2.setAutoResizeMode(JTable.AUTO_RESIZE_LAST_COLUMN);
        table2.setRowHeight(20);
        table2.setAutoResizeMode(8);

        int row = Row;
        tableModelNew1 = new DefaultTableModel();
        // Заповнення стовпців таблиці
        String[] colHed = new String[row];
        for (int i = 0; i < row; i++) {
            colHed[i] = "F" + (i+1);
        }

        tableModelNew1.setColumnIdentifiers(colHed);
        // Заповнення таблиці даних
        String[][] array2 = new String[row][row];
        for (int i = 0; i < row; i++) {
            for (int j = 0; j < row; j++) {
                array2[i][j] = "0";
            }
        }

        for (int i = 0; i < row; i++) {
            tableModelNew1.addRow(array2[i]);
        }

        // Створення таблиці на основі моделі даних

        table2 = new JTable(tableModelNew1);

        JPanel panel1 = new JPanel();
        JLabel label1 = new JLabel("Заповніть матрицю впливу факторів");
        // panel1.add(label1, BorderLayout.NORTH);
        panel1.add(table2, BorderLayout.NORTH);

        dialog1.add(panel1, BorderLayout.NORTH);
        //

        JPanel panel2 = new JPanel();
        JLabel label2 = new JLabel("Введіть початковий стан системи");

        panel2.add(label2, BorderLayout.NORTH);

        String[][] array4 = new String[1][row];
        String[] array5 = new String[row];
        for (int i = 0; i < row; i++) {
            array4[0][i] = "0";
        }
    }
}

```

```

        array5[i] = "F" + (i+1);
    }
    table3 = new JTable(array4,array5);
    panel2.add(table3, BorderLayout.SOUTH);

    dialog1.add(panel2, BorderLayout.CENTER);

    buttonNew = new JButton("Провести розрахунки");

    dialog1.add(buttonNew, BorderLayout.SOUTH);
//    dialog1.setVisible(true);

    logic1 = (logic1 == true) ? false : true;
    if (logic1 == true) {
        dialog1.setVisible(true);
    } else {
        dialog1.setVisible(false);
    }

    }
});

buttonNew = new JButton("Провести розрахунки");

buttonNew.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.out.println("23432423");
    }
});

// Формування інтерфейсу

jLabel = new JLabel("Введіть назву моделі");
jtextField = new JTextField();

Box contents = new Box(BoxLayout.Y_AXIS);
contents.add(new JLabel(""));
contents.add(jLabel);
contents.add(jtextField);
contents.add(new JLabel(""));
jLabel2 = new JLabel("Визначтесь із факторами впливу");
contents.add(jLabel2);
contents.add(new JScrollPane(table1));
// contents.add(new JScrollPane(table2));
getContentPane().add(contents);

JPanel buttons = new JPanel();

buttons.add(add);
buttons.add(remove);
buttons.add(getMatrix);

```

```
getContentPane().add(buttons, "South");  
// Виведення вікна на екран  
setSize(500, 300);  
setVisible(true);  
}  
  
public Object[][] getTableData(JTable table) {  
    DefaultTableModel dtm = (DefaultTableModel) table.getModel();  
    int nRow = dtm.getRowCount(), nCol = dtm.getColumnCount();  
    Object[][] tableData = new Object[nRow][nCol];  
    for (int i = 0; i < nRow; i++)  
        for (int j = 0; j < nCol; j++)  
            tableData[i][j] = dtm.getValueAt(i, j);  
  
    for (int i = 0; i < nRow; i++) {  
        System.out.println(Arrays.toString(tableData[i]));  
    }  
    System.out.println(table.getRowCount());  
  
    Row = table.getRowCount();  
    return tableData;  
}
```

Додаток В.2. Головний клас програми **"Когнітивна модель"**

```

/**
 * Когнітивна модель (Аналіз факторів при онкозахворюванні)
 * @author Помазун Оксана Миколаївна
 */

public class Main {

    String[] arrIndicators;
    String[] edcode;
    double arrData[][];
    double arrays[];
    double[] resVector;
    double[][] result1;
    double[][] result2;

    String name = "Когнітивна модель \"Аналіз факторів при онкозахворюванні\"";

    String a1 = "Матриця факторів впливу";
    String a2 = "Зміна показників у % покровоко";
    String a3 = "Зміна показників у % з накопичувальним підсумком";

    public Main() {
        arrIndicators = new String[]{};

        arrIndicators = new String[]

        { "Можливість не захворіти",
          "Паління",
          "Низька тривалість сну",
          "Прийом ліків",
          "Вживання алкоголю",
          "Здоровий спосіб життя",
          "Здорове харчування"};

        edcode = new String[]{
            "(А)",
            "(Б)",
            "(В)",
            "(Г)",
            "(Д)",
            "(Е)",
            "(З)"};

        arrData = new double[][]{

```

```

        {0.0, 0.0, 0.0, -0.9, 0.0, 0.1, 0.0},
        {-1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0},
        {-0.5, 0.0, 0.0, -0.4, 0.0, 0.0, 0.0},
        {-0.3, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0},
        {-0.7, 0.0, 0.0, -0.7, 0.0, 0.0, 0.0},
        {0.9, 0.95, 0.8, -0.2, 0.7, 0.0, 0.5},
        {0.7, 0.0, 0.0, 0.0, 0.0, 0.1, 0.0},
    };

    arrays = new double[]{0.0, 0.0, 0.0, 0.0, 0.0, 10.00, 0};

    getResultMatrixMultiplication();
    getResultSum();
}

public double[][] getResultMatrixMultiplication() {

    result1 = new double[19][7];
    result1[0] = arrays;
    int Row = arrData.length;
    int o = arrays.length;

    for (int j = 1; j < result1.length; j++) {

        resVector = new double[]{0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0};

        for (int i = 0; i < Row; i++) {
            for (int k = 0; k < o; k++) {
                resVector[i] += arrData[i][k] * arrays[k];
            }
            result1[j][i] = resVector[i];
        }
        arrays = resVector;
    }

    return result1;
}

public double[][] getResultSum() {
    result2 = new double[19][7];
    int m = arrData.length;

    for (int i = 0; i < result1.length; i++) {
        for (int j = 0; j < m; j++) {
            if (i == 0) {
                result2[i][j] = 100.0 * (1 + result1[i][j] / 100.0) - 100.0;
            } else {
                result2[i][j] = (result2[i - 1][j] + 100.0) * (1 + result1[i][j] / 100.0) - 100.0;
            }
        }
    }
}

```

```

    }
  }
}

return result2;
}

public String curr(double x) {
    String Cur;
    NumberFormat fG = NumberFormat.getNumberInstance();
    Cur = fG.format(x);
    return Cur;
}

public String[][] printResult(String indicators, String[] code, double[][] matrix) {

    String[][] printTable = new String[matrix.length + 1][matrix[0].length + 1];

    printTable[0][0] = indicators;

    for (int j = 0; j < matrix[0].length; j++) {
        printTable[0][j+1]=code[j];
    }

    for (int i = 0; i < printTable.length - 1; i++) {
        for (int j = 0; j < printTable[0].length - 1; j++) {
            if (j == 0) printTable[i + 1][j] = "" + i;
            printTable[i + 1][j + 1] = String.format("%.2f", matrix[i][j]);
        }
    }

    for (int i = 0; i < printTable.length; i++) {
        System.out.println(Arrays.toString(printTable[i]));
    }

    return printTable;
}

public void buildChart(String indicators, String[] code, double[][] matrix){
    // створюємо каркас вікна
    JDialog frame = new JDialog();
    // заголовок вікна
    frame.setTitle(name);
    // Для закриття програми
    frame.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);

    List<XYSeries> list = new ArrayList<XYSeries>();
    for (int i = 0; i < 7; i++) {
        list.add(new XYSeries("Фактор \" + code[i] + "\""));
    }

    int a = -1;

```

```

for (XYSeries line : list){
    a++;
    for (int i = 0; i < matrix.length; i++) {
        for (int j = 0; j < matrix.length; j++) {
            if(i==a)
                line.add(j, matrix[j][i]);
        }
    }
}

// Додаємо необхідні рядки
XYSeriesCollection data = new XYSeriesCollection();
for (XYSeries line : list){
    data.addSeries(line);
}

//створюємо діаграму
final JFreeChart chart = ChartFactory.createXYLineChart(
    indicators,
    "рік",
    "Фактори",
    data,
    PlotOrientation.VERTICAL,
    true,
    true,
    false
);

//створюємо панель для графіка
final ChartPanel chartPanel = new ChartPanel(chart);
//встановлюємо розмір графіка
chartPanel.setPreferredSize(new java.awt.Dimension(900, 370));
//додаємо панель на створений нами фрейм
frame.setContentPane(chartPanel);
frame.pack();
frame.setVisible(true);
}

public void buildTable(String indicators, String[] code, double[][] matrix){
    JDialog frame = new JDialog();
    frame.setTitle(indicators);
    frame.setDefaultCloseOperation(JDialog.DISPOSE_ON_CLOSE);

    String [][] newMatrix = new String[matrix.length][matrix[0].length];

    for (int i = 0; i < matrix.length; i++) {
        for (int j = 0; j < matrix[0].length; j++) {
            newMatrix[i][j] = String.format("%.3f", matrix[i][j]);
        }
    }
}

```

```
JTable tableNew = new JTable(newMatrix, code);
JTableHeader th = tableNew.getTableHeader();
frame.add(th, BorderLayout.NORTH);
frame.add(tableNew, BorderLayout.CENTER);
frame.setSize(900,700);

frame.pack();
frame.setVisible(true);
}

public static void main(String[] args) {
    Main main = new Main();
    // Вихідні дані
    main.setResult(main.a1, main.edcode,main.arrData);

    // Результати в консоль
    main.setResult(main.a2, main.arrIndicators, main.result1);
    main.setResult(main.a3, main.arrIndicators, main.result2);

    // Виведення графіків
    main.buildChart(main.a3, main.arrIndicators,main.result2);
    main.buildChart(main.a2, main.arrIndicators,main.result1);

    // Виведення таблиць
    main.buildTable(main.a2, main.arrIndicators,main.result1);
    main.buildTable(main.a3, main.arrIndicators,main.result2);
}
}
```

ДОДАТОК Г. Відомість дипломної роботи

