ЗАТВЕРДЖЕНО

1116130.00935-01-ЛЗ

АВТОНОМНА КЛІЄНТСЬКА ЧАСТИНА ДЛЯ СИСТЕМИ «МЕТОДИЧНІ МАТЕРІАЛИ»

Текст програми

1116130.00935-01 12-01

Листів 36

2016

АНОТАЦІЯ

Документ 1116130.00935-01 12-01 «Автономна клієнтська частина для системи «Методичні матеріали». Текст програми» входить до складу програмної документації до дипломного проекту.

У даному документі описана структура та текст додатку для робочого столу. Додаток написано на мові Java. Об’єм пам’яті, що займає програма комплексу та конфігураційні файли, складає 100 Мб. Конфігурація комп’ютера стандартна, на ньому повинно бути встановлені переглядачі файлів формату \*.pdf, \*.doc, \*.ppt.

ЗМІСТ

[1 Структура програми 4](#_Toc453016661)

[1.1 Структура компоненту core 4](#_Toc453016662)

[1.2 Структура компоненту model 4](#_Toc453016663)

[1.3 Структура компоненту ui 5](#_Toc453016664)

[2 Текст програм 6](#_Toc453016665)

[2.1 Текст файлу DocumentBuilder.java 6](#_Toc453016666)

[2.2 Текст файлу OpenSaveDocument.java 7](#_Toc453016667)

[2.3 Текст файлу Application.java 8](#_Toc453016668)

[2.4 Текст файлу WriteXML.java 8](#_Toc453016669)

[2.5 Текст файлу XMLBase.java 10](#_Toc453016670)

[2.6 Текст файлу XMLData.java 13](#_Toc453016671)

[2.7 Текст файлу DocumentEntity.java 13](#_Toc453016672)

[2.8 Текст файлу MaterialTableModel.java 15](#_Toc453016673)

[2.9 Текст файлу SearchResultsTableModel.java 17](#_Toc453016674)

[2.10 Текст файлу AdvancedSearch.java 20](#_Toc453016675)

[2.11 Текст файлу First.java 23](#_Toc453016676)

[2.12 Текст файлу LoginAdminForm.java 28](#_Toc453016677)

[2.13 Текст файлу MainForm.java 31](#_Toc453016678)

# 1 СТРУКТУРА ПРОГРАМИ

Додаток написано на мові програмування Java в середовищі програмування IntelliJIDEA Community Edition.

Автономна частина складається з наступних компонентів:

* core (компоненти, що включають логіку програми, роботу з базами даних та обробку запитів);
* models (компоненти, що включають у себе моделі для керування компонентами на формі, їх синхронізації з внутрішнім станом програми);
* ui (компоненти, що включають у себе форми та класи для роботи з ними).

## 1.1 Структура компоненту core

Файл DocumentBuilder.java містить методи для побудови моделі документів за результатами парсингу файлу бази даних.

Файл OpenSaveDocument.java містить методи для відкриття та збереження файлів.

Файл Application.java – головний файл програми, використовується для завантаження форм.

Файл WriteXML.java – файл зі створенням усіх полів та їх заповненням у файлі XML.

Файл XMLBase.java – файл, що містить методи для парсингу файлу XML, наприклад, пошук за полями.

Файл XMLData.java – інтерфейс, що використовується у XMLBase.java, містить прототипи функцій.

## 1.2 Структура компоненту model

Файл DocumentEntity.java містить описання документу, всі поля, що містяться в базі та використовуються при пошуку.

Файл MaterialTableModel.java містить описання таблиці головного вікна, у якій розміщені все документи, що є в базі.

Файл SearchResultsTableModel.java містить описання таблиці, що виводить результати пошуку.

## 1.3 Структура компоненту ui

Файл AdvancedSearch.jfd містить форму розширеного пошуку за матеріалами у базі.

Файл AdvancedSearch.java містить описання компонентів форми AdvancedSearch.jfd та дій над ними.

Файл First.jfd містить форму авторизації студента, що пропонує обрати факультет на спеціальність.

Файл First.java містить описання компонентів форми First.jfd та дій над ними.

Файл LoginAdminForm.jfd містить форму авторизації адміністратора, що пропонує ввести логін та пароль

Файл LoginAdminForm.java містить описання компонентів форми LoginAdminForm.jfd та дій над ними.

Файл MainForm.jfd містить головну форму для роботи з системою «Методичні матеріали».

Файл MainForm.java містить описання компонентів форми MainForm.jfd та дій над ними.

# 2 ТЕКСТ ПРОГРАМ

## 2.1 Текст файлу DocumentBuilder.java

package core;

import core.model.DocumentEntity;

import org.jdom.Element;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import static java.io.File.separator;

/\*\*

\* Created by KateS on 6/3/2016.

\*/

public class DocumentBuilder {

@SuppressWarnings("unchecked")

public static DocumentEntity getDocumentEntity(Element document) {

DocumentEntity doc = new DocumentEntity();

Element libraryKey = document.getChild("library\_key");

Element name = document.getChild("name");

Element department = document.getChild("department");

Element publishingYear = document.getChild("publishing\_year");

Element uploadDate = document.getChild("upload\_date");

Element filePath = document.getChild("file\_path");

List<Element> specialties = document.getChildren("specialties");

List<Element> authors = document.getChildren("authors");

doc.setLibraryKey(toInt(libraryKey));

doc.setName(toStr(name));

doc.setDepartment(toStr(department));

doc.setPublishingYear(toInt(publishingYear));

doc.setUploadDate(toDate(uploadDate));

doc.setFilePath(toStr(filePath));

doc.setSpecialties(toStringList(specialties));

doc.setAuthors(toStringList(authors));

return doc;

}

public static String listToString(List<String> stringList){

StringBuilder resultString = new StringBuilder();

String sep = "";

for(String s: stringList) {

resultString.append(sep).append(s);

sep = separator;

}

return resultString.toString();

}

private static List<String> toStringList(List<Element> elementList) {

List<String> stringList = new ArrayList<>();

for (Element element : elementList) {

stringList.add(element.getValue());

}

return stringList;

}

private static int toInt(Element element) {

return Integer.parseInt(element.getValue());

}

private static String toStr(Element element) {

return element.getValue();

}

private static Date toDate(Element element) {

try {

SimpleDateFormat sdf = new SimpleDateFormat("dd-mm-yyyy");

return sdf.parse(element.getValue());

} catch (ParseException e) {

e.printStackTrace();

}

return null;

}

private static List<DocumentEntity> getDocumentEntityList(List<Element> documentList) {

List<DocumentEntity> result = new ArrayList<>();

for (Element document : documentList) {

result.add(getDocumentEntity(document));

}

return result;

}

}

## 2.2 Текст файлу OpenSaveDocument.java

package core;

import java.awt.\*;

import java.io.File;

import java.io.IOException;

/\*\*

\* Created by KateS on 6/5/2016.

\*/

public class OpenSaveDocument {

public static void open(String path) {

File file = new File(path);

Desktop dt = Desktop.getDesktop();

try {

dt.open(file);

} catch (IOException e) {

e.printStackTrace();

}

}

}

## 2.3 Текст файлу Application.java

package core;

import ui.First;

import javax.swing.\*;

/\*\*

\* Created by KateS on 5/3/2016.

\*/

public class Application {

public static void main(String[] args) {

try {

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());

First first = new First();

first.setVisible(true);

} catch (ClassNotFoundException | InstantiationException | UnsupportedLookAndFeelException | IllegalAccessException e) {

e.printStackTrace();

}

}

}

## 2.4 Текст файлу WriteXML.java

package core;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import javax.xml.transform.Transformer;

import javax.xml.transform.TransformerException;

import javax.xml.transform.TransformerFactory;

import javax.xml.transform.dom.DOMSource;

import javax.xml.transform.stream.StreamResult;

import java.io.File;

/\*\*

\* Created by KateS on 5/23/2016.

\*/

public class WriteXML {

public static void main(String[] args) {

try{

DocumentBuilderFactory documentBuilderFactory = DocumentBuilderFactory.newInstance();

DocumentBuilder documentBuilder = documentBuilderFactory.newDocumentBuilder();

Document document = documentBuilder.newDocument();

Element rootElement = document.createElement("library");

document.appendChild(rootElement);

Element file = document.createElement("document");

rootElement.appendChild(file);

//Attr attr = document.createAttribute("id");

//attr.setValue("1");

//file.setAttributeNode(attr);

file.setAttribute("id", "1");

Element libraryKey = document.createElement("library\_key");

libraryKey.appendChild(document.createTextNode("1378"));

file.appendChild(libraryKey);

Element name = document.createElement("name");

name.appendChild(document.createTextNode("Українська мова"));

file.appendChild(name);

Element department = document.createElement("department");

department.appendChild(document.createTextNode("Мовознавства та філології"));

file.appendChild(department);

Element publishingYear = document.createElement("publishing\_year");

publishingYear.appendChild(document.createTextNode("2014"));

file.appendChild(publishingYear);

Element uploadDate = document.createElement("upload\_date");

uploadDate.appendChild(document.createTextNode("01-01-2015"));

file.appendChild(uploadDate);

Element filePath = document.createElement("file\_path");

filePath.appendChild(document.createTextNode("C:\\file.pdf"));

file.appendChild(filePath);

Element specialities = document.createElement("specialities");

file.appendChild(specialities);

Element specialty = document.createElement("specialty");

specialty.appendChild(document.createTextNode("ПЗ"));

specialities.appendChild(specialty);

Element authors = document.createElement("authors");

file.appendChild(authors);

Element author = document.createElement("author");

author.appendChild(document.createTextNode("Бочарова"));

authors.appendChild(author);

TransformerFactory transformerFactory = TransformerFactory.newInstance();

Transformer transformer = transformerFactory.newTransformer();

DOMSource source = new DOMSource(document);

StreamResult result = new StreamResult(new File("K:\\library.xml"));

transformer.transform(source, result);

System.out.println("File saved!");

}catch (ParserConfigurationException pce){

pce.printStackTrace();

}catch (TransformerException tfe){

tfe.printStackTrace();

}

}

}

## 2.5 Текст файлу XMLBase.java

package core;

import org.jdom.Document;

import org.jdom.Element;

import org.jdom.JDOMException;

import org.jdom.input.SAXBuilder;

import java.io.File;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

/\*\*

\* Created by KateS on 6/3/2016.

\*/

public class XMLBase implements XMLData {

private List<Element> documentList = getAllDocuments();

private Element rootElement() {

try {

File inputFile = new File("K:/library.xml");

SAXBuilder saxBuilder = new SAXBuilder();

Document document = saxBuilder.build(inputFile);

return document.getRootElement();

} catch (JDOMException | IOException e) {

e.printStackTrace();

}

return null;

}

@SuppressWarnings("unchecked")

private List<Element> getAllDocuments() {

return (List<Element>) rootElement().getChildren();

}

@Override

@SuppressWarnings("unchecked")

public List<Element> getDocumentsBySpecialty(String specialty) {

List<Element> resultList = new ArrayList<>();

for (Element document : documentList) {

Element specialties = document.getChild("specialties");

List<Element> specialtyList = specialties.getChildren();

for (Element specialtyElement : specialtyList) {

if (specialty.equalsIgnoreCase(specialtyElement.getValue())) {

resultList.add(document);

}

}

}

return resultList;

}

@Override

@SuppressWarnings("unchecked")

public List<Element> getDocumentsByLibraryKey(int libraryKey) {

List<Element> resultList = new ArrayList<>();

for (Element document : documentList) {

Element documentLibraryKey = document.getChild("library\_key");

if (Integer.parseInt(documentLibraryKey.getValue()) == libraryKey) {

resultList.add(document);

}

}

return resultList;

}

@Override

public List<Element> getDocumentsByName(String name) {

List<Element> resultList = new ArrayList<>();

for (Element document : documentList) {

Element documentName = document.getChild("name");

if (documentName.getValue().toLowerCase().contains(name.toLowerCase())) {

resultList.add(document);

}

}

return resultList;

}

@Override

public List<Element> getDocumentsByPublishingYear(int publishingYear) {

List<Element> resultList = new ArrayList<>();

for (Element document : documentList) {

Element documentPublishingYear = document.getChild("publishing\_year");

if (Integer.parseInt(documentPublishingYear.getValue()) == publishingYear) {

resultList.add(document);

}

}

return resultList;

}

@Override

public List<Element> getDocumentsByAuthor(String author) {

List<Element> resultList = new ArrayList<>();

for (Element document : documentList) {

Element authors = document.getChild("authors");

List<Element> authorsList = authors.getChildren();

for (Element authorElement : authorsList) {

if(author.equalsIgnoreCase(authorElement.getValue())){

resultList.add(document);

}

}

}

return resultList;

}

@Override

public List<Element> getDocumentsByDepartment(String department) {

List<Element> resultList = new ArrayList<>();

for (Element document : documentList) {

Element documentDepartment = document.getChild("department");

if (documentDepartment.getValue().equalsIgnoreCase(department)) {

resultList.add(document);

}

}

return resultList;

}

@Override

public List<Element> getDocumentsByKeyWord(String word) {

List<Element> resultList = new ArrayList<>();

for (Element document : documentList){

Element documentLibraryKey = document.getChild("library\_key");

Element documentName = document.getChild("name");

Element documentPublishingYear = document.getChild("publishing\_year");

Element documentDepartment = document.getChild("department");

if (documentLibraryKey.getValue().toLowerCase().contains(word.toLowerCase())||

documentName.getValue().toLowerCase().contains(word.toLowerCase())||

documentPublishingYear.getValue().toLowerCase().contains(word.toLowerCase())||

documentDepartment.getValue().toLowerCase().contains(word.toLowerCase())){

resultList.add(document);

}

Element authors = document.getChild("authors");

List<Element> authorsList = authors.getChildren();

for (Element authorElement : authorsList) {

if(authorElement.getValue().toLowerCase().contains(word.toLowerCase())){

resultList.add(document);

}

}

}

return resultList;

}

@Override

public int getListSize(List<Element> elementList) {

return elementList.size();

}

}

## 2.6 Текст файлу XMLData.java

package core;

import org.jdom.Element;

import java.util.List;

/\*\*

\* Created by KateS on 6/3/2016.

\*/

public interface XMLData {

List<Element> getDocumentsBySpecialty(String specialty);

List<Element> getDocumentsByLibraryKey(int libraryKey);

List<Element> getDocumentsByName(String name);

List<Element> getDocumentsByPublishingYear(int publishingYear);

List<Element> getDocumentsByAuthor(String author);

List<Element> getDocumentsByDepartment(String department);

List<Element> getDocumentsByKeyWord (String word);

int getListSize (List<Element> elementList);

}

## 2.7 Текст файлу DocumentEntity.java

package core.model;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

/\*\*

\* Created by KateS on 6/3/2016.

\*/

public class DocumentEntity {

private int libraryKey;

private String name;

private String department;

private int publishingYear;

private Date uploadDate;

private String filePath;

private List<String> specialties = new ArrayList<>();

private List<String> authors = new ArrayList<>();

public DocumentEntity() {

}

public int getLibraryKey() {

return libraryKey;

}

public void setLibraryKey(int libraryKey) {

this.libraryKey = libraryKey;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getDepartment() {

return department;

}

public void setDepartment(String department) {

this.department = department;

}

public int getPublishingYear() {

return publishingYear;

}

public void setPublishingYear(int publishingYear) {

this.publishingYear = publishingYear;

}

public Date getUploadDate() {

return uploadDate;

}

public void setUploadDate(Date uploadDate) {

this.uploadDate = uploadDate;

}

public String getFilePath() {

return filePath;

}

public void setFilePath(String filePath) {

this.filePath = filePath;

}

public List<String> getSpecialties() {

return specialties;

}

public void setSpecialties(List<String> specialties) {

this.specialties = specialties;

}

public List<String> getAuthors() {

return authors;

}

public void setAuthors(List<String> authors) {

this.authors = authors;

}

}

## 2.8 Текст файлу MaterialTableModel.java

package ui.model;

import core.DocumentBuilder;

import core.XMLBase;

import core.model.DocumentEntity;

import javax.print.Doc;

import javax.swing.event.TableModelListener;

import javax.swing.table.TableModel;

import java.util.\*;

/\*\*

\* Created by KateS on 6/3/2016.

\*/

public class MaterialTableModel implements TableModel {

private Set<TableModelListener> listeners = new HashSet<>();

private List<DocumentEntity> documentEntityList = new ArrayList<>();

public MaterialTableModel(List<DocumentEntity> documentEntityList) {

this.documentEntityList = documentEntityList;

}

private XMLBase xmlBase = new XMLBase();

@Override

public int getRowCount() {

return xmlBase.getListSize(xmlBase.getDocumentsBySpecialty("ПЗ"));

}

@Override

public int getColumnCount() {

return 6;

}

@Override

public String getColumnName(int columnIndex) {

switch (columnIndex) {

case 0:

return "Библиотечный номер";

case 1:

return "Название";

case 2:

return "Кафедра";

case 3:

return "Год издания";

case 4:

return "Дата загрузки";

case 5:

return "Авторы";

}

return null;

}

@Override

public Class<?> getColumnClass(int columnIndex) {

switch (columnIndex) {

case 0:

return int.class;

case 1:

return String.class;

case 2:

return String.class;

case 3:

return int.class;

case 4:

return Date.class;

case 5:

return String.class;

}

return null;

}

@Override

public boolean isCellEditable(int rowIndex, int columnIndex) {

return false;

}

@Override

public Object getValueAt(int rowIndex, int columnIndex) {

DocumentEntity entity = documentEntityList.get(rowIndex);

switch (columnIndex) {

case 0:

return entity.getLibraryKey();

case 1:

return entity.getName();

case 2:

return entity.getDepartment();

case 3:

return entity.getPublishingYear();

case 4:

return entity.getUploadDate();

case 5:

String authors = DocumentBuilder.listToString(entity.getAuthors());

return authors;

}

return null;

}

@Override

public void setValueAt(Object aValue, int rowIndex, int columnIndex) {

}

@Override

public void addTableModelListener(TableModelListener l) {

listeners.add(l);

}

@Override

public void removeTableModelListener(TableModelListener l) {

listeners.remove(l);

}

}

## 2.9 Текст файлу SearchResultsTableModel.java

package ui.model;

import core.XMLBase;

import core.model.DocumentEntity;

import javax.swing.event.TableModelListener;

import javax.swing.table.TableModel;

import java.util.\*;

/\*\*

\* Created by KateS on 6/5/2016.

\*/

public class SearchResultsTableModel implements TableModel {

private Set<TableModelListener> listeners = new HashSet<>();

private List<DocumentEntity> documentEntityList = new ArrayList<>();

private String keyWord;

public SearchResultsTableModel(List<DocumentEntity> documentEntityList, String keyWord) {

this.documentEntityList = documentEntityList;

this.keyWord = keyWord;

}

private XMLBase xmlBase = new XMLBase();

@Override

public int getRowCount() {

return xmlBase.getListSize(xmlBase.getDocumentsByKeyWord(keyWord));

}

@Override

public int getColumnCount() {

return 6;

}

@Override

public String getColumnName(int columnIndex) {

switch (columnIndex) {

case 0:

return "Библиотечный номер";

case 1:

return "Название";

case 2:

return "Кафедра";

case 3:

return "Год издания";

case 4:

return "Дата загрузки";

case 5:

return "Авторы";

}

return null;

}

@Override

public Class<?> getColumnClass(int columnIndex) {

switch (columnIndex) {

case 0:

return int.class;

case 1:

return String.class;

case 2:

return String.class;

case 3:

return int.class;

case 4:

return Date.class;

case 5:

return String.class;

}

return null;

}

@Override

public boolean isCellEditable(int rowIndex, int columnIndex) {

return false;

}

@Override

public Object getValueAt(int rowIndex, int columnIndex) {

DocumentEntity entity = documentEntityList.get(rowIndex);

switch (columnIndex) {

case 0:

return entity.getLibraryKey();

case 1:

return entity.getName();

case 2:

return entity.getDepartment();

case 3:

return entity.getPublishingYear();

case 4:

return entity.getUploadDate();

case 5:

return entity.getAuthors();

}

return null;

}

@Override

public void setValueAt(Object aValue, int rowIndex, int columnIndex) {

}

@Override

public void addTableModelListener(TableModelListener l) {

listeners.add(l);

}

@Override

public void removeTableModelListener(TableModelListener l) {

listeners.remove(l);

}

}

## 2.10 Текст файлу AdvancedSearch.java

/\*

\* Created by JFormDesigner on Sat Jun 04 22:07:18 EEST 2016

\*/

package ui;

import java.awt.\*;

import javax.swing.\*;

import javax.swing.GroupLayout;

import com.jgoodies.forms.factories.\*;

/\*\*

\* @author KateS

\*/

public class AdvancedSearch extends JFrame {

public AdvancedSearch() {

initComponents();

}

private void initComponents() {

// JFormDesigner - Component initialization - DO NOT MODIFY //GEN-BEGIN:initComponents

label1 = new JLabel();

label2 = new JLabel();

label3 = new JLabel();

label4 = new JLabel();

label5 = new JLabel();

textField1 = new JTextField();

textField2 = new JTextField();

textField4 = new JTextField();

textField5 = new JTextField();

textField6 = new JTextField();

button1 = new JButton();

//======== this ========

setTitle("\u0420\u0430\u0441\u0448\u0438\u0440\u0435\u043d\u043d\u044b\u0439 \u043f\u043e\u0438\u0441\u043a");

Container contentPane = getContentPane();

//---- label1 ----

label1.setText("\u041d\u0430\u0437\u0432\u0430\u043d\u0438\u0435:");

label1.setFont(new Font("Times New Roman", Font.PLAIN, 18));

//---- label2 ----

label2.setText("\u0410\u0432\u0442\u043e\u0440:");

label2.setFont(new Font("Times New Roman", Font.PLAIN, 18));

//---- label3 ----

label3.setText("\u0413\u043e\u0434 \u0438\u0437\u0434\u0430\u043d\u0438\u044f:");

label3.setFont(new Font("Times New Roman", Font.PLAIN, 18));

//---- label4 ----

label4.setText("\u041a\u0430\u0444\u0435\u0434\u0440\u0430:");

label4.setFont(new Font("Times New Roman", Font.PLAIN, 18));

//---- label5 ----

label5.setText("\u0411\u0438\u0431\u043b\u0438\u043e\u0442\u0435\u0447\u043d\u044b\u0439 \u043d\u043e\u043c\u0435\u0440:");

label5.setFont(new Font("Times New Roman", Font.PLAIN, 18));

//---- button1 ----

button1.setText("\u041f\u043e\u0438\u0441\u043a");

button1.setFont(new Font("Times New Roman", Font.PLAIN, 18));

GroupLayout contentPaneLayout = new GroupLayout(contentPane);

contentPane.setLayout(contentPaneLayout);

contentPaneLayout.setHorizontalGroup(

contentPaneLayout.createParallelGroup()

.addGroup(contentPaneLayout.createSequentialGroup()

.addGap(27, 27, 27)

.addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.TRAILING, false)

.addGroup(contentPaneLayout.createSequentialGroup()

.addGroup(contentPaneLayout.createParallelGroup()

.addComponent(label3)

.addComponent(label1, GroupLayout.PREFERRED\_SIZE, 104, GroupLayout.PREFERRED\_SIZE))

.addGap(84, 84, 84)

.addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.TRAILING)

.addComponent(textField2, GroupLayout.PREFERRED\_SIZE, 236, GroupLayout.PREFERRED\_SIZE)

.addComponent(textField1, GroupLayout.PREFERRED\_SIZE, 236, GroupLayout.PREFERRED\_SIZE)))

.addGroup(GroupLayout.Alignment.LEADING, contentPaneLayout.createSequentialGroup()

.addComponent(label5)

.addGap(18, 18, 18)

.addGroup(contentPaneLayout.createParallelGroup()

.addComponent(button1, GroupLayout.PREFERRED\_SIZE, 151, GroupLayout.PREFERRED\_SIZE)

.addComponent(textField5))))

.addGap(167, 167, 167)

.addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.TRAILING, false)

.addGroup(contentPaneLayout.createSequentialGroup()

.addComponent(label4)

.addPreferredGap(LayoutStyle.ComponentPlacement.RELATED, GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(textField4, GroupLayout.PREFERRED\_SIZE, 304, GroupLayout.PREFERRED\_SIZE))

.addGroup(contentPaneLayout.createSequentialGroup()

.addComponent(label2)

.addGap(72, 72, 72)

.addComponent(textField6, GroupLayout.PREFERRED\_SIZE, 304, GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(25, Short.MAX\_VALUE))

);

contentPaneLayout.setVerticalGroup(

contentPaneLayout.createParallelGroup()

.addGroup(contentPaneLayout.createSequentialGroup()

.addGap(30, 30, 30)

.addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.BASELINE)

.addComponent(textField6, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(label2)

.addComponent(textField1, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(label1))

.addGap(18, 18, 18)

.addGroup(contentPaneLayout.createParallelGroup()

.addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.BASELINE)

.addComponent(label3)

.addComponent(textField4, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE)

.addComponent(label4))

.addComponent(textField2, GroupLayout.Alignment.TRAILING, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.BASELINE)

.addComponent(label5)

.addComponent(textField5, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(LayoutStyle.ComponentPlacement.RELATED, 42, Short.MAX\_VALUE)

.addComponent(button1)

.addGap(21, 21, 21))

);

pack();

setLocationRelativeTo(null);

// JFormDesigner - End of component initialization //GEN-END:initComponents

}

// JFormDesigner - Variables declaration - DO NOT MODIFY //GEN-BEGIN:variables

private JLabel label1;

private JLabel label2;

private JLabel label3;

private JLabel label4;

private JLabel label5;

private JTextField textField1;

private JTextField textField2;

private JTextField textField4;

private JTextField textField5;

private JTextField textField6;

private JButton button1;

// JFormDesigner - End of variables declaration //GEN-END:variables

}

## 2.11 Текст файлу First.java

package ui;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import javax.swing.GroupLayout;

/\*

\* Created by JFormDesigner on Fri Jun 03 00:37:36 EEST 2016

\*/

/\*\*

\* @author KateS

\*/

public class First extends JFrame {

private final DefaultComboBoxModel RailwayElectrification = new DefaultComboBoxModel(new String[]{"Электрический транспорт", "Электромеханические системы автоматизации и электропривод", "Электротехнические системы электропотребления", "Теплоэнергетика "});

private final DefaultComboBoxModel Mechanic = new DefaultComboBoxModel(new String[]{"Локомотивы и локомотивное хозяйство", "Вагоны и вагонное хозяйство", "Подъемно-транспортные, строительные, дорожные, мелиоративные машины и оборудование"});

private final DefaultComboBoxModel TransportationProcessManagement = new DefaultComboBoxModel(new String[]{"Организация перевозок и управление на железнодорожном транспорте", "Организация перевозок и управление на автомобильном транспорте"});

private final DefaultComboBoxModel BridgesAndTonnels = new DefaultComboBoxModel(new String[]{"Мосты и транспортные тоннели"});

private final DefaultComboBoxModel IndustrialAndCivilBuilding = new DefaultComboBoxModel(new String[]{"Промышленное и гражданское строительство", "Водоснабжение и водоотведение", "Экология и охрана окружающей среды"});

private final DefaultComboBoxModel OrganisationOfConstructionOfRailways = new DefaultComboBoxModel(new String[]{"Железнодорожные сооружения и путевое хозяйство", "Автомобильные дороги и аэродромы"});

private final DefaultComboBoxModel TechnicalCybernetics = new DefaultComboBoxModel(new String[]{"Программное обеспечение автоматизированных систем", "Компьютерные системы и сети", "Экономическая кибернетика", "Защита информации в компьютерных системах и сетях", "Автоматика и автоматизация на транспорте"});

private final DefaultComboBoxModel EconomicsAndManagementOnTransport = new DefaultComboBoxModel(new String[]{"Менеджмент организаций и администрирование", "Учет и аудит", "Финансы и кредит", "Перевод", "Интеллектуальная собственность", "Управление финансово-экономической безопасностью"});

public First() {

initComponents();

specialtyComboBox.setModel(RailwayElectrification);

}

private void FacultyComboBoxItemStateChanged(ItemEvent e) {

}

private void loginStudentBtnActionPerformed(ActionEvent e) {

MainForm mainForm = new MainForm();

mainForm.setVisible(true);

}

private void specialtyComboBoxActionPerformed(ActionEvent e) {

}

private void facultyComboBoxActionPerformed(ActionEvent e) {

switch (facultyComboBox.getSelectedIndex()){

case 0:

specialtyComboBox.setModel(RailwayElectrification);

break;

case 1:

specialtyComboBox.setModel(Mechanic);

break;

case 2:

specialtyComboBox.setModel(TransportationProcessManagement);

break;

case 3:

specialtyComboBox.setModel(BridgesAndTonnels);

break;

case 4:

specialtyComboBox.setModel(IndustrialAndCivilBuilding);

break;

case 5:

specialtyComboBox.setModel(OrganisationOfConstructionOfRailways);

break;

case 6:

specialtyComboBox.setModel(TechnicalCybernetics);

break;

case 7:

specialtyComboBox.setModel(EconomicsAndManagementOnTransport);

break;

}

}

private void initComponents() {

// JFormDesigner - Component initialization - DO NOT MODIFY //GEN-BEGIN:initComponents

label1 = new JLabel();

facultyComboBox = new JComboBox<>();

label3 = new JLabel();

label4 = new JLabel();

specialtyComboBox = new JComboBox();

loginStudentBtn = new JButton();

button2 = new JButton();

//======== this ========

setResizable(false);

setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

setTitle("\u0410\u0432\u0442\u043e\u0440\u0438\u0437\u0430\u0446\u0438\u044f \u0441\u0442\u0443\u0434\u0435\u043d\u0442\u0430");

Container contentPane = getContentPane();

//---- label1 ----

label1.setText("\u0414\u043e\u0431\u0440\u043e \u043f\u043e\u0436\u0430\u043b\u043e\u0432\u0430\u0442\u044c \u0432 \u0441\u0438\u0441\u0442\u0435\u043c\u0443 \"\u041c\u0435\u0442\u043e\u0434\u0438\u0447\u0435\u0441\u043a\u0438\u0435 \u043c\u0430\u0442\u0435\u0440\u0438\u0430\u043b\u044b\"!");

label1.setFont(new Font("Times New Roman", Font.PLAIN, 22));

label1.setHorizontalAlignment(SwingConstants.CENTER);

//---- facultyComboBox ----

facultyComboBox.setModel(new DefaultComboBoxModel<>(new String[] {

"\u042d\u043b\u0435\u043a\u0442\u0440\u0438\u0444\u0438\u043a\u0430\u0446\u0438\u044f \u0436\u0435\u043b\u0435\u0437\u043d\u044b\u0445 \u0434\u043e\u0440\u043e\u0433",

"\u041c\u0435\u0445\u0430\u043d\u0438\u0447\u0435\u0441\u043a\u0438\u0439",

"\u0423\u043f\u0440\u0430\u0432\u043b\u0435\u043d\u0438\u0435 \u043f\u0440\u043e\u0446\u0435\u0441\u0441\u0430\u043c\u0438 \u043f\u0435\u0440\u0435\u0432\u043e\u0437\u043e\u043a",

"\u041c\u043e\u0441\u0442\u044b \u0438 \u0442\u043e\u043d\u043d\u0435\u043b\u0438",

"\u041f\u0440\u043e\u043c\u044b\u0448\u043b\u0435\u043d\u043d\u043e\u0435 \u0438 \u0433\u0440\u0430\u0436\u0434\u0430\u043d\u0441\u043a\u043e\u0435 \u0441\u0442\u0440\u043e\u0438\u0442\u0435\u043b\u044c\u0441\u0442\u0432\u043e",

"\u041e\u0440\u0433\u0430\u043d\u0438\u0437\u0430\u0446\u0438\u044f \u0441\u0442\u0440\u043e\u0438\u0442\u0435\u043b\u044c\u0441\u0442\u0432\u0430 \u0436\u0435\u043b\u0435\u0437\u043d\u044b\u0445 \u0434\u043e\u0440\u043e\u0433",

"\u0422\u0435\u0445\u043d\u0438\u0447\u0435\u0441\u043a\u0430\u044f \u043a\u0438\u0431\u0435\u0440\u043d\u0435\u0442\u0438\u043a\u0430",

"\u042d\u043a\u043e\u043d\u043e\u043c\u0438\u043a\u043e-\u0433\u0443\u043c\u0430\u043d\u0438\u0442\u0430\u0440\u043d\u044b\u0439"

}));

facultyComboBox.setFont(new Font("Times New Roman", Font.PLAIN, 20));

facultyComboBox.setSelectedIndex(0);

facultyComboBox.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

facultyComboBoxActionPerformed(e);

}

});

//---- label3 ----

label3.setText("\u0412\u0430\u0448 \u0444\u0430\u043a\u0443\u043b\u044c\u0442\u0435\u0442:");

label3.setFont(new Font("Times New Roman", Font.PLAIN, 18));

//---- label4 ----

label4.setText("\u0412\u0430\u0448\u0430 \u0441\u043f\u0435\u0446\u0438\u0430\u043b\u044c\u043d\u043e\u0441\u0442\u044c:");

label4.setFont(new Font("Times New Roman", Font.PLAIN, 18));

//---- specialtyComboBox ----

specialtyComboBox.setFont(new Font("Times New Roman", Font.PLAIN, 20));

specialtyComboBox.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

specialtyComboBoxActionPerformed(e);

}

});

//---- loginStudentBtn ----

loginStudentBtn.setText("\u0412\u043e\u0439\u0442\u0438");

loginStudentBtn.setFont(new Font("Times New Roman", Font.PLAIN, 18));

loginStudentBtn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

loginStudentBtnActionPerformed(e);

}

});

//---- button2 ----

button2.setText("\u0412\u043e\u0439\u0442\u0438 \u043a\u0430\u043a \u0430\u0434\u043c\u0438\u043d\u0438\u0441\u0442\u0440\u0430\u0442\u043e\u0440");

button2.setFont(new Font("Times New Roman", Font.PLAIN, 18));

GroupLayout contentPaneLayout = new GroupLayout(contentPane);

contentPane.setLayout(contentPaneLayout);

contentPaneLayout.setHorizontalGroup(

contentPaneLayout.createParallelGroup()

.addGroup(GroupLayout.Alignment.TRAILING, contentPaneLayout.createSequentialGroup()

.addGap(195, 448, Short.MAX\_VALUE)

.addGroup(contentPaneLayout.createParallelGroup()

.addGroup(GroupLayout.Alignment.TRAILING, contentPaneLayout.createSequentialGroup()

.addComponent(loginStudentBtn, GroupLayout.PREFERRED\_SIZE, 135, GroupLayout.PREFERRED\_SIZE)

.addGap(441, 441, 441))

.addGroup(GroupLayout.Alignment.TRAILING, contentPaneLayout.createSequentialGroup()

.addComponent(button2, GroupLayout.PREFERRED\_SIZE, 244, GroupLayout.PREFERRED\_SIZE)

.addGap(383, 383, 383))))

.addGroup(contentPaneLayout.createSequentialGroup()

.addGroup(contentPaneLayout.createParallelGroup()

.addGroup(contentPaneLayout.createSequentialGroup()

.addGroup(contentPaneLayout.createParallelGroup()

.addGroup(contentPaneLayout.createSequentialGroup()

.addGap(17, 17, 17)

.addComponent(label3, GroupLayout.PREFERRED\_SIZE, 156, GroupLayout.PREFERRED\_SIZE)

.addGap(59, 59, 59))

.addGroup(GroupLayout.Alignment.TRAILING, contentPaneLayout.createSequentialGroup()

.addContainerGap()

.addComponent(label4, GroupLayout.PREFERRED\_SIZE, 196, GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)))

.addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.LEADING, false)

.addComponent(facultyComboBox, GroupLayout.DEFAULT\_SIZE, 823, Short.MAX\_VALUE)

.addComponent(specialtyComboBox, GroupLayout.DEFAULT\_SIZE, 823, Short.MAX\_VALUE)))

.addGroup(contentPaneLayout.createSequentialGroup()

.addGap(202, 202, 202)

.addComponent(label1, GroupLayout.PREFERRED\_SIZE, 615, GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(20, Short.MAX\_VALUE))

);

contentPaneLayout.setVerticalGroup(

contentPaneLayout.createParallelGroup()

.addGroup(contentPaneLayout.createSequentialGroup()

.addGap(14, 14, 14)

.addComponent(label1, GroupLayout.PREFERRED\_SIZE, 51, GroupLayout.PREFERRED\_SIZE)

.addGap(54, 54, 54)

.addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.TRAILING)

.addComponent(label3, GroupLayout.PREFERRED\_SIZE, 33, GroupLayout.PREFERRED\_SIZE)

.addComponent(facultyComboBox, GroupLayout.PREFERRED\_SIZE, 30, GroupLayout.PREFERRED\_SIZE))

.addGap(45, 45, 45)

.addGroup(contentPaneLayout.createParallelGroup()

.addComponent(label4, GroupLayout.PREFERRED\_SIZE, 33, GroupLayout.PREFERRED\_SIZE)

.addComponent(specialtyComboBox, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(LayoutStyle.ComponentPlacement.RELATED, 73, Short.MAX\_VALUE)

.addComponent(loginStudentBtn, GroupLayout.PREFERRED\_SIZE, 43, GroupLayout.PREFERRED\_SIZE)

.addGap(40, 40, 40)

.addComponent(button2)

.addGap(16, 16, 16))

);

setSize(1095, 495);

setLocationRelativeTo(null);

// JFormDesigner - End of component initialization //GEN-END:initComponents

}

// JFormDesigner - Variables declaration - DO NOT MODIFY //GEN-BEGIN:variables

private JLabel label1;

private JComboBox<String> facultyComboBox;

private JLabel label3;

private JLabel label4;

private JComboBox specialtyComboBox;

private JButton loginStudentBtn;

private JButton button2;

// JFormDesigner - End of variables declaration //GEN-END:variables

}

## 2.12 Текст файлу LoginAdminForm.java

/\*

\* Created by JFormDesigner on Sat Jun 04 14:30:34 EEST 2016

\*/

package ui;

import java.awt.\*;

import javax.swing.\*;

import javax.swing.GroupLayout;

/\*\*

\* @author KateS

\*/

public class LoginAdminForm extends JFrame {

public LoginAdminForm() {

initComponents();

}

private void initComponents() {

// JFormDesigner - Component initialization - DO NOT MODIFY //GEN-BEGIN:initComponents

label1 = new JLabel();

label2 = new JLabel();

passwordField1 = new JPasswordField();

textField1 = new JTextField();

button3 = new JButton();

button4 = new JButton();

//======== this ========

setDefaultCloseOperation(WindowConstants.DISPOSE\_ON\_CLOSE);

setResizable(false);

setTitle("\u0410\u0432\u0442\u043e\u0440\u0438\u0437\u0430\u0446\u0438\u044f \u0430\u0434\u043c\u0438\u043d\u0438\u0441\u0442\u0440\u0430\u0442\u043e\u0440\u0430");

Container contentPane = getContentPane();

//---- label1 ----

label1.setText("\u041b\u043e\u0433\u0438\u043d:");

label1.setFont(new Font("Times New Roman", Font.PLAIN, 18));

//---- label2 ----

label2.setText("\u041f\u0430\u0440\u043e\u043b\u044c:");

label2.setFont(new Font("Times New Roman", Font.PLAIN, 18));

//---- button3 ----

button3.setText("\u0412\u043e\u0439\u0442\u0438");

button3.setFont(new Font("Times New Roman", Font.PLAIN, 18));

//---- button4 ----

button4.setText("\u041e\u0442\u043c\u0435\u043d\u0430");

button4.setFont(new Font("Times New Roman", Font.PLAIN, 18));

GroupLayout contentPaneLayout = new GroupLayout(contentPane);

contentPane.setLayout(contentPaneLayout);

contentPaneLayout.setHorizontalGroup(

contentPaneLayout.createParallelGroup()

.addGroup(contentPaneLayout.createSequentialGroup()

.addGap(109, 109, 109)

.addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.LEADING, false)

.addGroup(contentPaneLayout.createSequentialGroup()

.addComponent(button3, GroupLayout.PREFERRED\_SIZE, 110, GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(LayoutStyle.ComponentPlacement.RELATED, GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(button4, GroupLayout.PREFERRED\_SIZE, 110, GroupLayout.PREFERRED\_SIZE))

.addGroup(contentPaneLayout.createSequentialGroup()

.addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.LEADING, false)

.addComponent(label1, GroupLayout.DEFAULT\_SIZE, GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(label2, GroupLayout.PREFERRED\_SIZE, 90, GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.LEADING, false)

.addComponent(textField1)

.addComponent(passwordField1, GroupLayout.PREFERRED\_SIZE, 298, GroupLayout.PREFERRED\_SIZE))))

.addGap(0, 121, Short.MAX\_VALUE))

);

contentPaneLayout.setVerticalGroup(

contentPaneLayout.createParallelGroup()

.addGroup(contentPaneLayout.createSequentialGroup()

.addGap(83, 83, 83)

.addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.BASELINE)

.addComponent(label1, GroupLayout.PREFERRED\_SIZE, 36, GroupLayout.PREFERRED\_SIZE)

.addComponent(textField1, GroupLayout.PREFERRED\_SIZE, GroupLayout.DEFAULT\_SIZE, GroupLayout.PREFERRED\_SIZE))

.addGap(33, 33, 33)

.addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.BASELINE)

.addComponent(label2, GroupLayout.PREFERRED\_SIZE, 36, GroupLayout.PREFERRED\_SIZE)

.addComponent(passwordField1, GroupLayout.PREFERRED\_SIZE, 36, GroupLayout.PREFERRED\_SIZE))

.addPreferredGap(LayoutStyle.ComponentPlacement.RELATED, 65, Short.MAX\_VALUE)

.addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.BASELINE)

.addComponent(button4)

.addComponent(button3))

.addGap(51, 51, 51))

);

pack();

setLocationRelativeTo(getOwner());

// JFormDesigner - End of component initialization //GEN-END:initComponents

}

// JFormDesigner - Variables declaration - DO NOT MODIFY //GEN-BEGIN:variables

private JLabel label1;

private JLabel label2;

private JPasswordField passwordField1;

private JTextField textField1;

private JButton button3;

private JButton button4;

// JFormDesigner - End of variables declaration //GEN-END:variables

}

## 2.13 Текст файлу MainForm.java

package ui;

import java.awt.\*;

import java.awt.event.\*;

import java.util.\*;

import java.util.List;

import javax.swing.\*;

import javax.swing.table.TableModel;

import com.jgoodies.forms.factories.\*;

import core.DocumentBuilder;

import core.OpenSaveDocument;

import core.XMLBase;

import core.model.DocumentEntity;

import org.jdom.Element;

import ui.model.MaterialTableModel;

import ui.model.SearchResultsTableModel;

/\*

\* Created by JFormDesigner on Mon May 23 16:12:25 EEST 2016

\*/

/\*\*

\* @author KateS

\*/

public class MainForm extends JFrame {

private XMLBase xmlBase = new XMLBase();

private List<Element> documentsBySpecialty = xmlBase.getDocumentsBySpecialty("пз");

private List<DocumentEntity> documentEntityList = new ArrayList<>();

private TableModel materialTableModel = new MaterialTableModel(documentEntityList);

public MainForm() {

initComponents();

for (Element document : documentsBySpecialty) {

documentEntityList.add(DocumentBuilder.getDocumentEntity(document));

}

// materialTableModel.addTableModelListener(table1);

allMaterialsTbl.setModel(materialTableModel);

allMaterialsTbl.setRowHeight(40);

}

private void openLoginAdminBtnActionPerformed(ActionEvent e) {

LoginAdminForm loginAdminForm = new LoginAdminForm();

loginAdminForm.setVisible(true);

}

private void allMaterialsTableMouseClicked(MouseEvent e) {

int rowIndex = allMaterialsTbl.rowAtPoint(e.getPoint());

if(allMaterialsTbl.getModel().equals(materialTableModel)){

OpenSaveDocument.open(DocumentBuilder.getDocumentEntity(documentsBySpecialty.get(rowIndex)).getFilePath());

}else {

List<Element> documentsByKeyWord = xmlBase.getDocumentsByKeyWord(textField1.getText());

for (Element document : documentsByKeyWord) {

documentEntityList.add(DocumentBuilder.getDocumentEntity(document));

}

OpenSaveDocument.open(DocumentBuilder.getDocumentEntity(documentsByKeyWord.get(rowIndex)).getFilePath());

}

}

private void extendSearchBtnActionPerformed(ActionEvent e) {

AdvancedSearch advancedSearch = new AdvancedSearch();

advancedSearch.setVisible(true);

}

private void searchBtnActionPerformed(ActionEvent e) {

List<DocumentEntity> documentEntityList = new ArrayList<>();

List<Element> documentsByKeyWord = xmlBase.getDocumentsByKeyWord(textField1.getText());

for (Element document : documentsByKeyWord) {

documentEntityList.add(DocumentBuilder.getDocumentEntity(document));

}

SearchResultsTableModel searchResultsTableModel = new SearchResultsTableModel(documentEntityList, textField1.getText());

allMaterialsTbl.setModel(searchResultsTableModel);

}

private void initComponents() {

// JFormDesigner - Component initialization - DO NOT MODIFY //GEN-BEGIN:initComponents

DefaultComponentFactory compFactory = DefaultComponentFactory.getInstance();

textField1 = new JTextField();

searchBtn = new JButton();

extendSearchBtn = new JButton();

openLoginAdminBtn = new JButton();

separator1 = compFactory.createSeparator("\u0421\u043f\u0438\u0441\u043e\u043a \u043f\u043e\u0441\u043b\u0435\u0434\u043d\u0438\u0445 \u0437\u0430\u0433\u0440\u0443\u0436\u0435\u043d\u043d\u044b\u0445 \u043c\u0430\u0442\u0435\u0440\u0438\u0430\u043b\u043e\u0432");

scrollPane1 = new JScrollPane();

allMaterialsTbl = new JTable();

separator2 = compFactory.createSeparator("\u0421\u043f\u0438\u0441\u043e\u043a \u0432\u0430\u0436\u043d\u044b\u0445 \u043c\u0430\u0442\u0435\u0440\u0438\u0430\u043b\u043e\u0432");

scrollPane2 = new JScrollPane();

table2 = new JTable();

//======== this ========

setResizable(false);

setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

setTitle("\u0421\u0438\u0441\u0442\u0435\u043c\u0430 \"\u041c\u0435\u0442\u043e\u0434\u0438\u0447\u0435\u0441\u043a\u0438\u0435 \u043c\u0430\u0442\u0435\u0440\u0438\u0430\u043b\u044b\"");

Container contentPane = getContentPane();

//---- textField1 ----

textField1.setFont(new Font("Times New Roman", Font.PLAIN, 18));

textField1.setToolTipText("\u0412\u0432\u0435\u0434\u0438\u0442\u0435 \u043a\u043b\u044e\u0447\u0435\u0432\u044b\u0435 \u0441\u043b\u043e\u0432\u0430 \u0434\u043b\u044f \u043f\u043e\u0438\u0441\u043a\u0430 ...");

//---- searchBtn ----

searchBtn.setText("\u041f\u043e\u0438\u0441\u043a");

searchBtn.setFont(new Font("Times New Roman", Font.PLAIN, 18));

searchBtn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

searchBtnActionPerformed(e);

}

});

//---- extendSearchBtn ----

extendSearchBtn.setText("\u0420\u0430\u0441\u0448\u0438\u0440\u0435\u043d\u043d\u044b\u0439 \u043f\u043e\u0438\u0441\u043a");

extendSearchBtn.setFont(new Font("Times New Roman", Font.PLAIN, 18));

extendSearchBtn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

extendSearchBtnActionPerformed(e);

}

});

//---- openLoginAdminBtn ----

openLoginAdminBtn.setText("\u0412\u043e\u0439\u0442\u0438 \u043a\u0430\u043a \u0430\u0434\u043c\u0438\u043d\u0438\u0441\u0442\u0440\u0430\u0442\u043e\u0440");

openLoginAdminBtn.setFont(new Font("Times New Roman", Font.PLAIN, 18));

openLoginAdminBtn.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

openLoginAdminBtnActionPerformed(e);

}

});

//---- separator1 ----

separator1.setFont(new Font("Times New Roman", Font.PLAIN, 22));

//======== scrollPane1 ========

{

//---- allMaterialsTbl ----

allMaterialsTbl.setFont(new Font("Times New Roman", Font.PLAIN, 18));

allMaterialsTbl.addMouseListener(new MouseAdapter() {

@Override

public void mouseClicked(MouseEvent e) {

allMaterialsTableMouseClicked(e);

}

});

scrollPane1.setViewportView(allMaterialsTbl);

}

//---- separator2 ----

separator2.setFont(new Font("Times New Roman", Font.PLAIN, 18));

//======== scrollPane2 ========

{

scrollPane2.setViewportView(table2);

}

GroupLayout contentPaneLayout = new GroupLayout(contentPane);

contentPane.setLayout(contentPaneLayout);

contentPaneLayout.setHorizontalGroup(

contentPaneLayout.createParallelGroup()

.addGroup(GroupLayout.Alignment.TRAILING, contentPaneLayout.createSequentialGroup()

.addGap(33, 33, 33)

.addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.LEADING, false)

.addComponent(separator1, GroupLayout.DEFAULT\_SIZE, 876, Short.MAX\_VALUE)

.addComponent(textField1, GroupLayout.DEFAULT\_SIZE, 876, Short.MAX\_VALUE)

.addComponent(scrollPane1, GroupLayout.DEFAULT\_SIZE, 876, Short.MAX\_VALUE))

.addGroup(contentPaneLayout.createParallelGroup()

.addGroup(contentPaneLayout.createSequentialGroup()

.addPreferredGap(LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(searchBtn, GroupLayout.PREFERRED\_SIZE, 118, GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(LayoutStyle.ComponentPlacement.RELATED, 202, Short.MAX\_VALUE)

.addComponent(extendSearchBtn, GroupLayout.PREFERRED\_SIZE, 208, GroupLayout.PREFERRED\_SIZE))

.addGroup(contentPaneLayout.createSequentialGroup()

.addGap(136, 136, 136)

.addGroup(contentPaneLayout.createParallelGroup()

.addComponent(separator2, GroupLayout.DEFAULT\_SIZE, GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(scrollPane2, GroupLayout.DEFAULT\_SIZE, 404, Short.MAX\_VALUE))))

.addContainerGap())

.addGroup(GroupLayout.Alignment.TRAILING, contentPaneLayout.createSequentialGroup()

.addContainerGap(607, Short.MAX\_VALUE)

.addComponent(openLoginAdminBtn, GroupLayout.PREFERRED\_SIZE, 244, GroupLayout.PREFERRED\_SIZE)

.addGap(604, 604, 604))

);

contentPaneLayout.setVerticalGroup(

contentPaneLayout.createParallelGroup()

.addGroup(contentPaneLayout.createSequentialGroup()

.addGap(17, 17, 17)

.addGroup(contentPaneLayout.createParallelGroup(GroupLayout.Alignment.BASELINE, false)

.addComponent(textField1)

.addComponent(searchBtn, GroupLayout.DEFAULT\_SIZE, GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(extendSearchBtn))

.addGap(20, 20, 20)

.addGroup(contentPaneLayout.createParallelGroup()

.addComponent(separator1, GroupLayout.DEFAULT\_SIZE, 38, Short.MAX\_VALUE)

.addComponent(separator2, GroupLayout.DEFAULT\_SIZE, 38, Short.MAX\_VALUE))

.addPreferredGap(LayoutStyle.ComponentPlacement.RELATED)

.addGroup(contentPaneLayout.createParallelGroup()

.addComponent(scrollPane2, GroupLayout.DEFAULT\_SIZE, 564, Short.MAX\_VALUE)

.addComponent(scrollPane1, GroupLayout.DEFAULT\_SIZE, 564, Short.MAX\_VALUE))

.addGap(18, 18, 18)

.addComponent(openLoginAdminBtn)

.addGap(16, 16, 16))

);

pack();

setLocationRelativeTo(getOwner());

// JFormDesigner - End of component initialization //GEN-END:initComponents

}

// JFormDesigner - Variables declaration - DO NOT MODIFY //GEN-BEGIN:variables

private JTextField textField1;

private JButton searchBtn;

private JButton extendSearchBtn;

private JButton openLoginAdminBtn;

private JComponent separator1;

private JScrollPane scrollPane1;

private JTable allMaterialsTbl;

private JComponent separator2;

private JScrollPane scrollPane2;

private JTable table2;

// JFormDesigner - End of variables declaration //GEN-END:variables

}