import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Node;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.input.KeyCode;  
import javafx.scene.input.MouseEvent;  
import javafx.scene.layout.BorderPane;  
import javafx.scene.layout.GridPane;  
import javafx.scene.layout  
import javafx.scene.text.Font;  
import javafx.scene.text.FontWeight;  
import javafx.scene.text.Text;  
import javafx.stage.Stage;  
  
import java.util.ArrayList;  
import java.util.Arrays;  
  
  
class Form {  
  
 private static TextField *textName*;  
 private static TextField *textNumber*;  
 private static TextField *textDate*;  
 private static TextField *textOSMUname*;  
 private static TextField *textCadastr*;  
 private static TextField *textArea*;  
 private static TextField *textUnit*;  
 private static TextField *textID1*;  
 private static TextField *textPoint1X*;  
 private static TextField *textPoint1Y*;  
 private static TextField *textAdres*;  
 private static TextField *textUtil*;  
 private static TextField *textCategory*;  
 private static TextField *textCoordSys*;  
 private static TextField *textImg*;  
 private static GridPane *mainGrid*;  
 private static GridPane *mainGrid1*;  
 private static TextField *newText*;  
 private static TextField *newText1*;  
 private static ArrayList<String> *imgs* = new ArrayList<String>();  
 private static int *i* = 5;  
 private static int *b* = 11;  
  
  
 static void display() {  
 Stage form1 = new Stage();  
 ScrollPane scrollPane = new ScrollPane();  
 form1.setTitle("Форма 1");  
  
 // form1.getIcons().add(new Image("info\_.jpg"));  
 form1.setHeight(650);  
 form1.setWidth(420);  
  
  
 //названия полей  
 Label labelName = new Label("Название");  
 Label labelNumber = new Label("Номер");  
 Label labelDate = new Label("Дата");  
 Label labelOSMUname = new Label("Название ОСМУ");  
 Label labelCadastr = new Label("Кадастровый номер");  
 Label labelArea = new Label("Area");  
 Label labelUnit = new Label("Unit");  
  
 //эти данные в 2-х местах  
 Label labelID1 = new Label("ID1");  
  
 Label labelPoint1X = new Label("Точка 1 X");  
 Label labelPoint1Y = new Label("Точка 1 Y");  
 //может понадобиться добавленеи полей для точек  
 Label labelAdres = new Label("Адрес");  
 Label labelUtil = new Label("Примененеие");  
 Label labelCategory = new Label("Категория");  
 Label labelCoordSys = new Label("Система координат");  
 //сделать так, чтобы можно было добавить несколько  
 Label labelImg = new Label("Изображения");  
  
 //сами поля  
 *textName* = new TextField();  
 *textNumber* = new TextField();  
 *textDate* = new TextField();  
 *textOSMUname* = new TextField();  
 *textCadastr* = new TextField();  
 *textArea* = new TextField();  
 *textUnit* = new TextField();  
 *textID1* = new TextField();  
 *textPoint1X* = new TextField();  
 *textPoint1Y* = new TextField();  
 *textAdres* = new TextField();  
 *textUtil* = new TextField();  
 *textCategory* = new TextField();  
 *textCoordSys* = new TextField();  
 *textImg* = new TextField();  
  
  
  
 Button cast = new Button("Создать XML");  
 cast.setOnAction(e -> *getStrings*());  
 // cast.setOnAction(e -> System.out.println(imgs.size()));  
  
 Button addImg = new Button("+");  
 addImg.setOnAction(e -> {  
 *newText* = new TextField();  
 *i*++;  
 *mainGrid1*.add(*newText*,1,*i*);  
 *newText*.setOnKeyPressed(ke -> {  
 if (ke.getCode().equals(KeyCode.*ENTER*) && *newText*.getLength()!=0) {  
 System.*out*.println(*newText*.getText());  
 //уведомление, что картинка добавлена  
  
 //добавленеи в массив текста  
 *imgs*.add(*newText*.getText());}  
 });  
 });  
  
 Button addPoint = new Button("+");  
 addPoint.setOnAction(e -> {  
  
 *newText* = new TextField();  
 *newText1* = new TextField();  
 *b*++;  
 *mainGrid*.add(*newText*,1,*b*);  
 *mainGrid*.add(*newText1*,1,*b*+1);  
 *b*++;  
// newText.setOnKeyPressed(ke -> {  
// if (ke.getCode().equals(KeyCode.ENTER) && newText.getLength()!=0) {  
// System.out.println(newText.getText());  
// //уведомление, что картинка добавлена  
//  
// //добавленеи в массив текста  
// imgs.add(newText.getText());}  
// });  
 });  
  
 *mainGrid* = new GridPane();  
 *mainGrid*.setAlignment(Pos.*CENTER*);  
 *mainGrid*.setHgap(10);  
 *mainGrid*.setVgap(10);  
 *mainGrid*.setPadding(new Insets(25,25,10,25));  
  
 *mainGrid1* = new GridPane();  
 *mainGrid1*.setAlignment(Pos.*CENTER*);  
 *mainGrid1*.setHgap(10);  
 *mainGrid1*.setVgap(10);  
 *mainGrid1*.setPadding(new Insets(0,25,10,25));  
  
 Text title = new Text("Форма 1");  
 title.setFont(Font.*font*("Times New Roman", FontWeight.*NORMAL*, 20));  
 *mainGrid*.add(title, 0, 0, 2, 1);  
  
 *mainGrid*.add(labelName, 0, 1);  
 *mainGrid*.add(*textName*, 1, 1);  
  
 *mainGrid*.add(labelNumber, 0, 2);  
 *mainGrid*.add(*textNumber*, 1, 2);  
  
 *mainGrid*.add(labelDate, 0, 3);  
 *mainGrid*.add(*textDate*, 1, 3);  
  
 *mainGrid*.add(labelOSMUname, 0, 4);  
 *mainGrid*.add(*textOSMUname*, 1, 4);  
  
 *mainGrid*.add(labelCadastr, 0, 5);  
 *mainGrid*.add(*textCadastr*, 1, 5);  
  
 *mainGrid*.add(labelArea, 0, 6);  
 *mainGrid*.add(*textArea*, 1, 6);  
  
 *mainGrid*.add(labelUnit, 0, 7);  
 *mainGrid*.add(*textUnit*, 1, 7);  
  
 *mainGrid*.add(labelID1, 0, 8);  
 *mainGrid*.add(*textID1*, 1, 8);  
  
 *mainGrid*.add(labelPoint1X, 0, 9);  
 *mainGrid*.add(*textPoint1X*, 1, 9);  
  
 *mainGrid*.add(labelPoint1Y, 0, 10);  
 *mainGrid*.add(*textPoint1Y*, 1, 10);  
  
 *mainGrid*.add(addPoint, 2, 10);  
  
  
 *mainGrid1*.add(labelAdres, 0, 0);  
 *mainGrid1*.add(*textAdres*, 1, 0);  
  
 *mainGrid1*.add(labelUtil, 0, 1);  
 *mainGrid1*.add(*textUtil*, 1, 1);  
  
 *mainGrid1*.add(labelCategory, 0, 2);  
 *mainGrid1*.add(*textCategory*, 1, 2);  
  
 *mainGrid1*.add(labelCoordSys, 0, 3);  
 *mainGrid1*.add(*textCoordSys*, 1, 3);  
  
 *mainGrid1*.add(labelImg, 0, 4);  
 *mainGrid1*.add(*textImg*, 1, 4);  
  
 *mainGrid1*.add(addImg, 2, 4);  
  
  
  
 //mainGrid.add(cast, 3, 16);  
  
 VBox v = new VBox(0);  
 v.getChildren().addAll(*mainGrid*,*mainGrid1*);  
  
 BorderPane b = new BorderPane();  
 b.setCenter(v);  
 b.setBottom(cast);  
 cast.setAlignment(Pos.*BASELINE\_RIGHT*);  
 //cast.set(new Insets(20));  
 // b.getChildren().addAll(mainGrid,cast);  
  
 scrollPane.setContent(b);  
 Scene sceneMain = new Scene(scrollPane);  
  
 form1.setScene(sceneMain);  
 form1.show();  
 }  
  
 private static void getStrings() {  
  
 String strings []= {*textName*.getText(),*textNumber*.getText(), *textDate*.getText(), *textOSMUname*.getText(), *textCadastr*.getText(), *textArea*.getText(),  
 *textUnit*.getText(), *textID1*.getText(), *textPoint1X*.getText(), *textPoint1Y*.getText(), *textAdres*.getText(),  
 *textUtil*.getText(), *textCategory*.getText(),*textCoordSys*.getText(),*textImg*.getText()};  
  
 Contrller.*rewriteForm1*(strings,*imgs*);  
 }  
}

import org.w3c.dom.\*;  
  
import javax.xml.parsers.DocumentBuilder;  
import javax.xml.parsers.DocumentBuilderFactory;  
import javax.xml.transform.OutputKeys;  
import javax.xml.transform.Transformer;  
import javax.xml.transform.TransformerFactory;  
import javax.xml.transform.dom.DOMSource;  
import javax.xml.transform.stream.StreamResult;  
import java.io.File;  
import java.util.ArrayList;  
  
  
class Contrller {  
  
 private static String *name*;  
 private static String *number*;  
 private static String *date*;  
 private static String *osmuName*;  
 private static String *cadastr*;  
 private static String *area*;  
 private static String *unit*;  
 private static String *id1*;  
 private static String *point1x*;  
 private static String *point1y*;  
 private static String *adres*;  
 private static String *util*;  
 private static String *category*;  
 private static String *img*;  
  
// static void bildForm1(String strings []) {  
// name = strings[0];  
// number = stringNumber;  
// date = stringDate;  
// osmuName = stringOSMUname;  
// cadastr = stringCadastr;  
// area = stringArea;  
// unit = stringUnit;  
// id1 = stringID1;  
// point1x = stringPoint1X;  
// point1y = stringPoint1Y;  
// adres = stringAdres;  
// util = stringUtil;  
// category = stringCategory;  
// img = stringImg;  
// }  
  
 static void rewriteForm1(String strings [],ArrayList <String> imgs) {  
 String filePath = "form1.xml";  
 File xmlFile = new File(filePath);  
 DocumentBuilderFactory factory = DocumentBuilderFactory.*newInstance*();  
 DocumentBuilder builder;  
 try {  
 builder = factory.newDocumentBuilder();  
 Document doc = builder.parse(xmlFile);  
 doc.getDocumentElement().normalize();  
  
 // обновляем значения  
 *updateElementValue*(doc, "Document", "Name", strings[0]);  
 *updateElementValue*(doc, "Document", "Number", strings[1]);  
 *updateElementValue*(doc, "Document", "Date", strings[2]);  
 *updateElementValue*(doc, "Document", "IssueOrgan", strings[3]);  
  
 *updateElementValue*(doc, "NewParcels", "CadastralBlock", strings[4]);  
 *updateElementValueIntros*(doc, "Area" ,"Area", strings[5]);  
 *updateElementValue*(doc, "Area", "Unit", strings[6]);  
  
 *updateElementValueIntro*(doc, "Entity\_Spatial", "Ent\_Sys", strings[7]);  
  
 *updateElementValueIntro*(doc, "NewOrdinate" ,"X", strings[8]);  
 *updateElementValueIntro*(doc, "NewOrdinate" ,"Y", strings[9]);  
  
 *updateElementValue*(doc, "NewParcels" ,"Note", strings[10]);  
  
 *updateElementValueIntro*(doc, "Utilization", "ByDoc", strings[11]);  
  
  
 *updateElementValueIntro*(doc, "Category", "Category", strings[12]);  
 *updateElementValueIntro*(doc, "Coord\_System", "Cs\_Id", strings[7]);  
 *updateElementValueIntro*(doc, "Coord\_System", "Name", strings[13]);  
  
  
 //определение длины массива имгс и в зависимость от этого добавление картинок  
 *updateElementValueIntro*(doc, "AppliedFile", "Name", strings[14]);  
 if (imgs.size() > 0) {  
 for (int i = 0; i <= imgs.size()-1; i++)  
 { //добавить нужное количество новых  
 *addElement*(doc,"ParcelSchema\_In\_Block","AppliedFile", "Name",imgs.get(i));  
  
  
 }  
 }  
  
  
 // запишем отредактированный элемент в файл  
 // или выведем в консоль  
 doc.getDocumentElement().normalize();  
 TransformerFactory transformerFactory = TransformerFactory.*newInstance*();  
 Transformer transformer = transformerFactory.newTransformer();  
 DOMSource source = new DOMSource(doc);  
 StreamResult result = new StreamResult(new File("form1Done.xml"));  
 transformer.setOutputProperty(OutputKeys.*INDENT*, "yes");  
 transformer.transform(source, result);  
 System.*out*.println("XML успешно изменен!");  
  
 } catch (Exception exc) {  
 exc.printStackTrace();  
 }  
 }  
  
  
 // изменяем значение существующего элемента  
 private static void updateElementValue(Document doc, String tagName, String elementName, String newText) {  
 Node nodeName = doc.getElementsByTagName(elementName).item(0);  
 nodeName.setTextContent(newText);  
 }  
  
 // изменяем значение существующего элемента атрибут  
 private static void updateElementValueIntro(Document doc, String elementName, String attrName, String newText) {  
 Node nodeName = doc.getElementsByTagName(elementName).item(0);  
 NamedNodeMap attributes = nodeName.getAttributes();  
 Node atr = attributes.getNamedItem(attrName);  
 atr.setTextContent(newText);  
 //nodeName.setTextContent(newText);  
 }  
  
 // изменяем значение существующего вложенные тэги  
 private static void updateElementValueIntros(Document doc, String tagName, String elementName, String newText) {  
 Node nodeName = doc.getElementsByTagName(tagName).item(0);  
 NodeList nodeList = nodeName.getChildNodes();  
 for (int i = 0; i < nodeList.getLength(); i++) {  
 Node nextNode = nodeList.item(i);  
  
 if (nextNode.getNodeName().equals(elementName)) {  
 nextNode.setTextContent(newText);  
 }  
 }  
 }  
 private static void addElement (Document doc,String parentTag, String childTag,String attrName, String content) {  
 Node nodeName = doc.getElementsByTagName(parentTag).item(0);  
 Element newElement = doc.createElement(childTag);  
 nodeName.appendChild(newElement);  
 newElement.setAttribute(attrName, content);  
 }  
}