|  |  |  |  |
| --- | --- | --- | --- |
| ***SPRAWOZDANIE NR 7*** | | | |
| Nazwa ćwiczenia | Ćwiczenie VII | | Obraz zawierający tekst  Opis wygenerowany automatycznie |
| Przedmiot | Programowanie obiektowe – laboratorium | |
| Student  grupa | Marcin Ogórkiewicz, grupa 7 | |
| Data ćwiczeń | 18.12.2023 | 22.01.2023 | Data oddania sprawozdania |

**HelloApplication**

package com.example.programowanieobiektowe7;  
  
import javafx.application.Application;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Scene;  
import javafx.stage.Stage;  
  
import java.io.IOException;  
  
public class HelloApplication extends Application {  
 @Override  
 public void start(Stage stage) throws IOException {  
 FXMLLoader fxmlLoader = new FXMLLoader(HelloApplication.class.getResource("hello-view.fxml"));  
 Scene scene = new Scene(fxmlLoader.load(), 600, 400);  
 stage.setTitle("Pacjenci");  
 stage.setResizable(false);  
 stage.setScene(scene);  
 stage.show();  
 }  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
}

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

**HelloController**

package com.example.programowanieobiektowe7;  
  
import javafx.fxml.FXML;  
import javafx.scene.control.ChoiceBox;  
import javafx.scene.control.TextField;  
  
import java.util.ArrayList;  
  
import static com.example.programowanieobiektowe7.Utils.*clearInputFields*;  
import static com.example.programowanieobiektowe7.Utils.*isValid*;  
  
public class HelloController {  
  
 @FXML  
 public TextField name, surname, age;  
 @FXML  
 public ChoiceBox<String> sortBox;  
 public final ArrayList<Patient> patientsList = new ArrayList<>();  
  
 public void initialize() {  
 sortBox.getItems().addAll("Imię", "Nazwisko", "Wiek", "Kolejności dodania");  
 sortBox.setValue("Kolejności dodania");  
 }  
  
 @FXML  
 public void handleAdd() {  
 String firstName = name.getText();  
 firstName = firstName.strip();  
 String lastName = surname.getText();  
 lastName = lastName.strip();  
 String ageText = age.getText();  
 ageText = ageText.strip();  
  
 if (firstName.isEmpty() || lastName.isEmpty() || ageText.isEmpty()) {  
 Utils.*errorAlert*("Wszystkie pola muszą być wypełnione");  
 return;  
 }  
  
 if (!*isValid*(firstName)) {  
 Utils.*errorAlert*("Imię musi zaczynać się z wielkiej litery i nie może zawierać cyfr oraz znaków specjalnych");  
 return;  
 }  
  
 if (!*isValid*(lastName)) {  
 Utils.*errorAlert*("Nazwisko musi zaczynać się z wielkiej litery i nie może zawierać cyfr oraz znaków specjalnych");  
 return;  
 }  
  
 int ageValue;  
 try {  
 ageValue = Integer.*parseInt*(ageText);  
 if (ageValue < 0 || ageValue > 150) {  
 Utils.*errorAlert*("Wiek musi być liczbą z przedziału 0-150");  
 return;  
 }  
 } catch (NumberFormatException e) {  
 Utils.*errorAlert*("Wprowadzony wiek nie jest poprawną liczbą całkowitą");  
 return;  
 }  
  
 Patient patient = new Patient(patientsList.size() + 1, firstName, lastName, ageValue);  
 patientsList.add(patient);  
 Utils.*successAlert*("Pacjent dodany pomyślnie!");  
  
 *clearInputFields*(name, surname, age);  
 }  
  
 @FXML  
 public void handlePrint() {  
 if (patientsList.isEmpty()) {  
 Utils.*errorAlert*("Lista pacjentów jest pusta");  
 return;  
 }  
  
 try {  
 Patient.*export*(patientsList, sortBox.getValue());  
 Utils.*successAlert*("Lista pacjentów wyeksportowana pomyślnie!");  
 } catch (Exception e) {  
 Utils.*errorAlert*("Nie udało się wyeksportować listy pacjentów");  
 }  
 }  
}

**Utils**

package com.example.programowanieobiektowe7;  
  
import javafx.scene.control.Alert;  
import javafx.scene.control.TextField;  
  
import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class Utils {  
  
 public static void successAlert(String message) {  
 *showAlert*("Sukces!", message, Alert.AlertType.*INFORMATION*);  
 }  
  
 public static void errorAlert(String message) {  
 *showAlert*("Błąd", message, Alert.AlertType.*ERROR*);  
 }  
  
 private static void showAlert(String title, String message, Alert.AlertType alertType) {  
 Alert alert = new Alert(alertType);  
 alert.setTitle(title);  
 alert.setHeaderText(null);  
 alert.setContentText(message);  
 alert.showAndWait();  
 }  
  
 private static void printDateTime() {  
 LocalDateTime currentDateTime = LocalDateTime.*now*();  
 DateTimeFormatter formatter = DateTimeFormatter.*ofPattern*("dd-MM-yyyy HH:mm:ss");  
 System.*out*.println("Raport pacjentów: " + currentDateTime.format(formatter));  
 }  
  
 public static void prepareTerminal() {  
 System.*out*.println("\n\n--------------------------------------------------");  
 *printDateTime*();  
 System.*out*.println("--------------------------------------------------");  
 }  
  
 public static void clearInputFields(TextField name, TextField surname, TextField age) {  
 name.clear();  
 surname.clear();  
 age.clear();  
 }  
  
 public static boolean isValid(String name) {  
 Pattern pattern = Pattern.*compile*("[A-ZĄĆĘŁŃÓŚŻŹ][a-ząćęłńóśżź]\*");  
 Matcher matcher = pattern.matcher(name);  
 return matcher.matches();  
 }  
}

**Patient**

package com.example.programowanieobiektowe7;  
  
import com.fasterxml.jackson.core.JsonProcessingException;  
import com.fasterxml.jackson.databind.ObjectMapper;  
  
import java.util.ArrayList;  
import java.util.Comparator;  
  
public class Patient {  
 private final int id;  
 private final String imie;  
 private final String nazwisko;  
 private final int wiek;  
  
 public Patient(int id, String firstName, String lastName, int age) {  
 this.id = id;  
 this.imie = firstName;  
 this.nazwisko = lastName;  
 this.wiek = age;  
 }  
 public int getId() {  
 return id;  
 }  
 public String getFistName() {  
 return imie;  
 }  
 public String getLastName() {  
 return nazwisko;  
 }  
 public int getAge() {  
 return wiek;  
 }  
  
 public static void export(ArrayList<Patient> patients, String mode) {  
 switch (mode) {  
 case "Imię" -> patients.sort(Comparator.*comparing*(Patient::getFistName));  
 case "Nazwisko" -> patients.sort(Comparator.*comparing*(Patient::getLastName));  
 case "Wiek" -> patients.sort(Comparator.*comparingInt*(Patient::getAge));  
 case "Kolejności dodania" -> patients.sort(Comparator.*comparingInt*(Patient::getId));  
 default -> {  
 Utils.*errorAlert*("Nieznany tryb sortowania");  
 return;  
 }  
 }  
 ObjectMapper objectMapper = new ObjectMapper();  
 try {  
 String json = objectMapper.writerWithDefaultPrettyPrinter().writeValueAsString(patients);  
 Utils.*prepareTerminal*();  
 System.*out*.println(json);  
 } catch (JsonProcessingException e) {  
 Utils.*errorAlert*("Nie udało się wyeksportować listy pacjentów");  
 }  
 }  
}

**hello-view**

<?xml version="1.0" encoding="UTF-8"?>  
  
<?import javafx.scene.control.\*?>  
<?import javafx.scene.layout.\*?>  
<?import javafx.scene.text.\*?>  
  
<AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="400.0" prefWidth="600.0" xmlns="http://javafx.com/javafx/17.0.2-ea" xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.programowanieobiektowe7.HelloController">  
 <children>  
 <Label layoutX="187.0" layoutY="34.0" text="Dane Pacjenta">  
 <font>  
 <Font size="36.0" />  
 </font>  
 </Label>  
 <Label layoutX="187.0" layoutY="108.0" text="Imię:">  
 <font>  
 <Font size="13.0" />  
 </font>  
 </Label>  
 <Label layoutX="157.0" layoutY="150.0" text="Nazwisko:">  
 <font>  
 <Font size="13.0" />  
 </font></Label>  
 <Label layoutX="183.0" layoutY="190.0" text="Wiek:">  
 <font>  
 <Font size="13.0" />  
 </font>  
 </Label>  
 <TextField fx:id="name" layoutX="226.0" layoutY="105.0" />  
 <TextField fx:id="surname" layoutX="226.0" layoutY="147.0" />  
 <TextField fx:id="age" layoutX="226.0" layoutY="187.0" />  
 <Button fx:id="add\_button" layoutX="226.0" layoutY="230.0" mnemonicParsing="false" onAction="#handleAdd" text="Dodaj do kolekcji" />  
 <Label layoutX="300.0" layoutY="276.0" text="Sortuj na podstawie:" />  
 <ChoiceBox fx:id="sortBox" layoutX="301.0" layoutY="294.0" prefWidth="150.0" />  
 <Button fx:id="print\_button" layoutX="226.0" layoutY="294.0" mnemonicParsing="false" onAction="#handlePrint" text="Wydrukuj" />  
 </children>  
</AnchorPane>

**Wnioski**

Ćwiczenie nauczyło mnie obsługi formularzy z użyciem formatu JSON w języku Java.