БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра информационных систем управления

Сачек Илья Валерьевич

Отчет

по 5 лабораторной работе

студента 2 курса 2 группы

Минск, 2020

Содержание:

Условие лабораторной работы…………………………………………………..2

Код решения задачи………………………………………………………………3

Пояснения к решению и результаты…………………………………………….6

Список использованных источников……………………………………………7

**Условие**

**Интерфейс предложить на Java FX**

Задание 4.1

Регулярные выражения. Из выпадающего списка можно выбрать тип данных (натуральное число, целое число, число с плавающей запятой (не забывать про запись 1e-5), дата, время, e-mail). Рядом со списком есть текстовое поле с проверяемыми данными. «Включать» зеленый кружок, если текстовое поле соответствует выбранному типу данных и красный кружок в противном случае.

Задание 4.2

Регулярные выражения. В многострочном тестовом поле находится текст. В список поместить все даты из этого поля.

**Код решения**

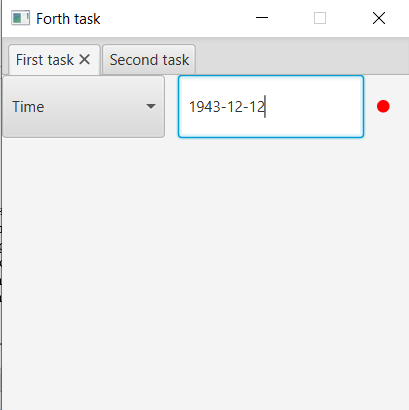
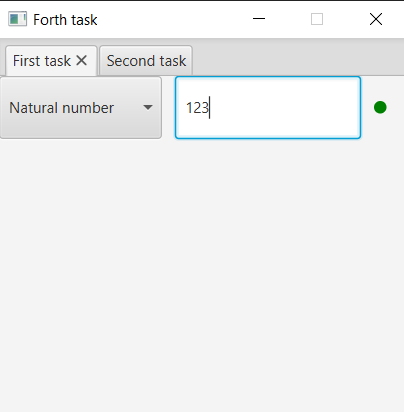
package sample;  
  
import javafx.application.Application;  
import javafx.collections.FXCollections;  
import javafx.geometry.Orientation;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.input.MouseEvent;  
import javafx.scene.layout.FlowPane;  
import javafx.scene.layout.Vbox;  
import javafx.scene.paint.Color;  
import javafx.scene.shape.Circle;  
import javafx.stage.Stage;  
  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;  
  
public class Main extends Application {  
 // разные паттерны для регулярных выражений  
 private final static Pattern N = Pattern.compile(«([0-9]\*)([1-9])([0-9]\*)»);  
 private final static Pattern Z = Pattern.compile(«[-+]?([0-9]\*)([0-9])([0-9]\*)»);  
 private final static Pattern R = Pattern.compile(«[-+]?(([0-9]\*[.]?[0-9]+)|([0-9]+[.]?[0-9]\*))([eE][-+]?[0-9]+)?»);  
 private final static Pattern TIME = Pattern.compile(«(([0-1][0-9])|([2][0-3]))☹[0-5][0-9])»);  
 private final static Pattern DATE = Pattern.compile(«\\d{4}-[01]\\d-[0-3]\\d»);  
 private final static Pattern EMAIL = Pattern.compile(«[a-z]([.\_]?[a-z]+)\*[@][a-z][a-z]\*[.]((org)|(net)|(ru)|(com)|(by))»,  
 Pattern.CASE\_INSENSITIVE);  
 // массив строк для выпадающего списка  
 private final static String[] ELEMENTS = {«Natural number», «Date», «E-Mail», «Time», «Real number», «Integer»};  
 // исходный текст для второго задания  
 private final static String START\_STRING = «dfkjbkjxdfhbkxdjkfbjkxdfn2020-12-20sjv<JSBVh,bSV» +  
 «sjbvdhjBDSvhjbSDvhjbSD1212-01-06<JvhjDvhjhdsvhjSJdvbhjBSDvhj,s6543-10-12sdkjvbS<HDhj,bSj,hvshdvhjSDJvs»;  
  
 @Override  
 public void start(Stage primaryStage) {  
 // название окна  
 primaryStage.setTitle(«Forth task»);  
  
 // высота и ширина  
 final double width = 325, height = 300;  
  
 // первое задание  
 // вкладки для разделения двух заданий  
 TabPane tabPane = new TabPane();  
 Tab firstTask = new Tab(«First task»), secondTask = new Tab(«Second task»);  
 tabPane.getTabs().add(firstTask);  
 tabPane.getTabs().add(secondTask);  
 Vbox vBox = new Vbox(tabPane);  
  
 // выпадающий список  
 ComboBox<String> comboBox = new ComboBox(FXCollections.observableArrayList(ELEMENTS));  
 comboBox.setPrefHeight(50);  
 comboBox.setValue(«Natural number»);  
  
 // панель для первого задания  
 FlowPane firstPane = new FlowPane(10, 10, comboBox);  
  
 // обработчик сообщений выпадающего списка  
 comboBox.setOnAction(event -> {  
 if (comboBox.getValue() == «Date») {  
 Alert alert = new Alert(Alert.AlertType.INFORMATION);  
 alert.setTitle(«Correct type of data!»);  
 alert.setHeaderText(«Please, write date in such format, as YYYY-MM-DD»);  
 alert.showAndWait();  
 }  
 });  
  
 TextField textField = new TextField();  
 textField.setPrefHeight(50);  
  
 // кружок для правильности данных  
 Circle circle = new Circle(40, 100, 5);  
 circle.setFill(Color.RED);  
  
 // обработчик сообщений смены цвета кружка  
 textField.textProperty().addListener((observable, oldValue, newValue) -> {  
 switch (comboBox.getValue()) {  
 case «E-Mail»: {  
 if (EMAIL.matcher(newValue).matches()) {  
 circle.setFill(Color.GREEN);  
 } else {  
 circle.setFill(Color.RED);  
 }  
 break;  
 }  
 case «Time»: {  
 if (TIME.matcher(newValue).matches()) {  
 circle.setFill(Color.GREEN);  
 } else {  
 circle.setFill(Color.RED);  
 }  
 break;  
 }  
 case «Natural number»: {  
 if (N.matcher(newValue).matches()) {  
 circle.setFill(Color.GREEN);  
 } else {  
 circle.setFill(Color.RED);  
 }  
 break;  
 }  
 case «Integer»: {  
 if (Z.matcher(newValue).matches()) {  
 circle.setFill(Color.GREEN);  
 } else {  
 circle.setFill(Color.RED);  
 }  
 break;  
 }  
 case «Real number»: {  
 if (R.matcher(newValue).matches()) {  
 circle.setFill(Color.GREEN);  
 } else {  
 circle.setFill(Color.RED);  
 }  
 break;  
 }  
 case «Date»: {  
 if (DATE.matcher(newValue).matches()) {  
 circle.setFill(Color.GREEN);  
 } else {  
 circle.setFill(Color.RED);  
 }  
 break;  
 }  
 }  
 });  
 firstPane.getChildren().add(textField);  
 firstPane.getChildren().add(circle);  
 firstTask.setContent(firstPane);  
  
 // второе задание  
 // панель для второго задания  
 FlowPane secondPane = new FlowPane();  
 secondPane.setOrientation(Orientation.VERTICAL);  
  
 // две TextArea для входного текста и для вывода дат  
 TextArea textArea = new TextArea(), finalArea = new TextArea();  
  
 // задаем разные характеристики для окон ввода и вывода  
 textArea.setMaxWidth(width);  
 textArea.setWrapText(true);  
 textArea.setPrefRowCount(6);  
 textArea.setText(START\_STRING);  
 finalArea.setPrefRowCount(6);  
 finalArea.setWrapText(true);  
 finalArea.setMaxWidth(width);  
  
 // кнопка поиска дат в тексте  
 Button findDates = new Button(«find dates!»);  
  
 // обработчик сообщений кнопки  
 findDates.addEventHandler(MouseEvent.MOUSE\_CLICKED, event -> {  
 String text = textArea.getText(); // получаем текст из поля  
 Matcher matcher = DATE.matcher(text); // ищем дату  
 finalArea.clear(); // чистим поле для выходного текста  
 while (matcher.find()) { // если нашли дату то выводим ее  
 finalArea.appendText(matcher.group() + «\n»);  
 }  
 });  
 secondPane.getChildren().add(textArea);  
 secondPane.getChildren().add(findDates);  
 secondPane.getChildren().add(finalArea);  
 secondTask.setContent(secondPane);  
  
 primaryStage.setScene(new Scene(vBox, width, height));  
 primaryStage.setResizable(false); // окно не изменет размер  
 primaryStage.show();  
 }  
  
 public static void main(String[] args) {  
 launch(args);  
 }  
}

**Пояснения к решению и результаты**

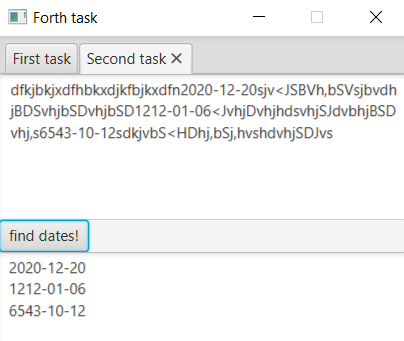
Пояснения:

В ходе решения задачи была использована библиотека для создания приложений JavaFX. Во вкладке первого задания есть выпадающий список, поле ввода текста и кружок обозначающий соответствие введенного текста формату выбранному в списке. Реализовано с помощью классов TabPane, Vbox, Tab, ComboBox, FlowPane, TextField и Circle. Во вкладке второго есть два поля ввода текста и кнопка поиска даты в тексте. Реализовано с помощью классов задания, а так же классов TextArea и Button.

Результаты:

Первое задание:

Второе задание:



**Список использованных источников**

1. <https://drive.google.com/file/d/1oWD4ajBuA_j18SLycmRg5lWPV98hEEYZ/view>
2. <https://stackoverflow.com/questions/16868446/javafx-exception-in-thread-main-java-lang-noclassdeffounderror-javafx-applica>
3. <https://metanit.com/java/javafx/3.2.php>