Третьяков І.С.

КІТ-118а

Лабораторна робота №16

# [Розробка графічного інтерфейсу користувача](https://oop-khpi.gitlab.io/#tasks)

***Мета:***

* Розширення функціональності параметризованих класів.

## **Вимоги**

Розробити графічний інтерфейс користувача для програми рішення [попередньої лабораторної роботи](https://oop-khpi.gitlab.io/%D0%B7%D0%B0%D0%B2%D0%B4%D0%B0%D0%BD%D0%BD%D1%8F/task15/) з використанням засобів [JavaFX](https://en.wikipedia.org/wiki/JavaFX).

ПРИКЛАДНА ЗАДАЧА:

Планувальник. Знайти всі конференції, що пройшли за останні три роки в Харкові та області з тривалістю не менше доби.

ОПИС ПРОГРАМИ

***2.1 Опис змінних:***

ArrayList<Event> arrayList = **new** ArrayList<Event>()

// об’єкт параметризованого контейнера

Scanner scan = new Scanner(System.in); // змінна для активування зчитування з консолі

***2.2 Ієрархія та структура класів.***

Main class – головний клас. Містить метод main(точку входу у програму) та методи по роботі з програмою для реалізації індивідуального завдання.

interface Linked - інтерфейс контенеру

class Event - клас прикладної задачі планувальника

class ArrayList - параметризований клас-контейнер, котрий зберігає інформацію планувальника

**ТЕКСТ ПРОГРАМИ**

Main.java

package application;

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.stage.Stage;

import javafx.scene.Parent;

import javafx.scene.Scene;

public class Main extends Application {

@Override

public void start(Stage primaryStage) throws Exception {

Parent root = FXMLLoader.load(getClass().getResource("sample.fxml"));

primaryStage.setTitle("Planner");

primaryStage.setScene(new Scene(root, 1280, 585));

primaryStage.show();

}

public static void main(String[] args) {

launch(args);

}

}

Event.java

**package** application;

**import** javafx.beans.property.SimpleStringProperty;

**public** **class** Event {

**private** String data; // Дата

**private** String startTime; // Время начала

**private** **int** duration; // Продолжительность

**private** String venue; // Место проведения

**private** String description; // Описание

**private** Participant[] participants; // Участники

**public** Event(String data, String startTime, **int** duration,

String venue, String description, Participant[] participants) {

**this**.data = data;

**this**.startTime = startTime;

**this**.duration = duration;

**this**.venue = venue;

**this**.description = description;

**this**.participants = participants;

}

**public** Event() {

}

**public** String getData() {

**return** **this**.data;

}

**public** **void** setData(String data) {

**this**.data = data;

}

**public** String getStartTime() {

**return** **this**.startTime;

}

**public** **void** setStartTime(String startTime) {

**this**.startTime = startTime;

}

**public** **int** getDuration() {

**return** **this**.duration;

}

**public** **void** setDuration(**int** duration) {

**this**.duration = duration;

}

**public** String getVenue() {

**return** **this**.venue;

}

**public** **void** setVenue(String venue) {

**this**.venue = venue;

}

**public** String getDescription() {

**return** **this**.description;

}

**public** **void** setDescription(String description) {

**this**.description = description;

}

**public** Participant[] getParticipants() {

**return** **this**.participants;

}

**public** **void** setParticipants(Participant[] participants) {

**this**.participants = participants;

}

@Override

**public** String toString() {

StringBuffer buf = **new** StringBuffer();

/\*buf.append("Дата: " + this.data + "\n");

buf.append("Время начала: " + this.startTime + "\n");

buf.append("Продолжительность: " + this.duration + "\n");

buf.append("Место проведения: " + this.venue + "\n");

buf.append("Описание: " + this.description + "\n");

buf.append("Участники: ");\*/

**for**(**int** i = 0; i < **this**.participants.length; i++) {

buf.append(**this**.participants[i].toString());

}

**return** buf.toString();

}

}

listController.java

package application;

import javafx.beans.property.SimpleStringProperty;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.collections.transformation.FilteredList;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.TextFieldTableCell;

public class listController {

@FXML

private TableColumn<Event, String> date;

@FXML

private Button EditButton;

@FXML

private TableColumn<Event, String> venue;

@FXML

private TextField dateButton;

@FXML

private TableColumn<Event, String> description;

@FXML

private TableColumn<Event, String> participant;

@FXML

private TableColumn<Event, String> duration;

@FXML

private TextField venueButton;

@FXML

private TextField durationButton;

@FXML

private TextField SearchField;

@FXML

private Button FindButton;

@FXML

private Button DeleteButton;

@FXML

private Button AddButton;

@FXML

private TextField startTimeButton;

@FXML

private TableColumn<Event, String> startTime;

@FXML

private TextField descriptionButton;

@FXML

private Label joobleButton;

@FXML

private TableView<Event> TableView;

@FXML

void editCompany(ActionEvent event) {

Event event2 = TableView.getSelectionModel().getSelectedItem();

event2.setData(edittedCell.getNewValue().toString());

}

@FXML

void delete(ActionEvent event) {

ObservableList<Event> selectedRows, allEvents;

allEvents = TableView.getItems();

//this gives us the rows that were selected

selectedRows = TableView.getSelectionModel().getSelectedItems();

//loop over the selected rows and remove the Person objects from the table

for (Event person: selectedRows)

{

allEvents.remove(person);

}

}

@FXML

void edit(ActionEvent event) {

TableView.refresh();

}

ObservableList<Event> items = FXCollections.observableArrayList();

@FXML

void add(ActionEvent event) {

// Участники

Participant par1[] = new Participant[2];

par1[0] = new Participant("Роман","Жолин",25);

par1[1] = new Participant("Евгений","Розумовский",24);

Participant[] par2 = new Participant[2];

par2[0] = new Participant("Игнат", "Третьяков", 18);

par2[1] = new Participant("Ростислав", "Малахов", 19);

items.add(new Event(dateButton.getText(), startTimeButton.getText(), Integer.parseInt(durationButton.getText()), venueButton.getText(),

descriptionButton.getText(), par1));

dateButton.clear();

startTimeButton.clear();

durationButton.clear();

venueButton.clear();

descriptionButton.clear();

}

private FilteredList<Event> filterdata;

@FXML

void SearchBtn(ActionEvent event) {

SearchField.textProperty().addListener((o, ov, nv) -> {

filterdata.setPredicate((Event e) -> {

String newVal=nv.toLowerCase();

return e.getData().toLowerCase().contains(newVal);

//return false;

});

});

TableView.setItems(filterdata);

}

@FXML

void initialize() {

filterdata = new FilteredList<Event>(items ,e->true);

TableView.itemsProperty().setValue(items);

date.setCellValueFactory(param-> new SimpleStringProperty(param.getValue().getData()));

startTime.setCellValueFactory(param-> new SimpleStringProperty(param.getValue().getStartTime()));

duration.setCellValueFactory(param-> new SimpleStringProperty(String.valueOf(param.getValue().getDuration())));

venue.setCellValueFactory(param-> new SimpleStringProperty(param.getValue().getVenue()));

description.setCellValueFactory(param-> new SimpleStringProperty(param.getValue().getDescription()));

participant.setCellValueFactory(param-> new SimpleStringProperty(param.getValue().toString()));

TableView.setEditable(true);

date.setCellFactory(TextFieldTableCell.forTableColumn());

}

}

Participant.java

**package** application;

**import** java.io.Serializable;

**public** **class** Participant **implements** Serializable {

**private** String name;

**private** String surname;

**private** **int** age;

**public** Participant(String name, String surname, **int** age){

**this**.name = name;

**this**.surname = surname;

**this**.age = age;

}

**public** Participant(){

}

**public** String getName() {

**return** **this**.name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getSurname() {

**return** **this**.surname;

}

**public** **void** setSurname(String surname) {

**this**.surname = surname;

}

**public** **int** getAge() {

**return** **this**.age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

@Override

**public** String toString() {

StringBuffer buf = **new** StringBuffer();

buf.append("Имя: " + **this**.name + "\n");

buf.append("Фамилия: " + **this**.surname + "\n");

buf.append("Возраст: " + **this**.age + "\n");

**return** buf.toString();

}

}

Sample.fxml

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<?import java.lang.\*?>

<?import javafx.scene.control.\*?>

<?import javafx.scene.layout.\*?>

<?import javafx.scene.text.\*?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.Label?>

<?import javafx.scene.control.TableColumn?>

<?import javafx.scene.control.TableView?>

<?import javafx.scene.control.TextField?>

<?import javafx.scene.layout.AnchorPane?>

<?import javafx.scene.text.Font?>

<?import javafx.scene.text.Text?>

<AnchorPane maxHeight=*"-Infinity"* maxWidth=*"-Infinity"* minHeight=*"-Infinity"* minWidth=*"-Infinity"* prefHeight=*"583.0"* prefWidth=*"1281.0"* xmlns=*"http://javafx.com/javafx/8"* xmlns:fx=*"http://javafx.com/fxml/1"* fx:controller=*"application.listController"*>

<children>

<AnchorPane maxHeight=*"-Infinity"* maxWidth=*"-Infinity"* minHeight=*"-Infinity"* minWidth=*"-Infinity"* opacity=*"0.84"* prefHeight=*"585.0"* prefWidth=*"1280.0"* style=*"-fx-background-color: #0b0b0b; -fx-border-color: #ffa600; -fx-border-width: 5px;"*>

<children>

<AnchorPane layoutX=*"5.0"* layoutY=*"57.0"* opacity=*"0.79"* prefHeight=*"525.0"* prefWidth=*"1275.0"* style=*"-fx-background-color: #ffffff;"*>

<children>

<TableView fx:id=*"TableView"* prefHeight=*"328.0"* prefWidth=*"722.0"*>

<columns>

<TableColumn fx:id=*"date"* onEditCommit=*"#editCompany"* prefWidth=*"63.99999713897705"* text=*"Date"* />

<TableColumn fx:id=*"startTime"* prefWidth=*"90.39999389648438"* text=*"Start time"* />

<TableColumn fx:id=*"duration"* prefWidth=*"123.0"* text=*"Duration (Minutes)"* />

<TableColumn fx:id=*"venue"* prefWidth=*"59.0"* text=*"Venue"* />

<TableColumn fx:id=*"description"* prefWidth=*"79.0"* text=*"Description"* />

<TableColumn fx:id=*"participant"* prefWidth=*"302.0"* text=*"Participant"* />

</columns>

</TableView>

<Button fx:id=*"DeleteButton"* layoutX=*"764.0"* layoutY=*"192.0"* mnemonicParsing=*"false"* onAction=*"#delete"* prefHeight=*"51.0"* prefWidth=*"217.0"* text=*"Delete Event"*>

<font>

<Font size=*"24.0"* />

</font>

</Button>

<Button fx:id=*"EditButton"* layoutX=*"1027.0"* layoutY=*"192.0"* mnemonicParsing=*"false"* onAction=*"#edit"* prefHeight=*"51.0"* prefWidth=*"200.0"* text=*"Edit"*>

<font>

<Font size=*"24.0"* />

</font>

</Button>

<Button fx:id=*"AddButton"* layoutX=*"1026.0"* layoutY=*"422.0"* mnemonicParsing=*"false"* onAction=*"#add"* prefHeight=*"51.0"* prefWidth=*"207.0"* text=*"Add Event"*>

<font>

<Font size=*"24.0"* />

</font>

</Button>

<Button fx:id=*"FindButton"* layoutX=*"1029.0"* layoutY=*"98.0"* mnemonicParsing=*"false"* text=*"Find"*>

<font>

<Font size=*"24.0"* />

</font>

</Button>

<Text layoutX=*"805.0"* layoutY=*"61.0"* strokeType=*"OUTSIDE"* strokeWidth=*"0.0"* text=*"Enter company name which u want to find"*>

<font>

<Font size=*"19.0"* />

</font>

</Text>

<TextField fx:id=*"dateButton"* layoutX=*"764.0"* layoutY=*"299.0"* prefHeight=*"26.0"* prefWidth=*"207.0"* promptText=*"Date (yyyy-mm-dd)"* />

<TextField fx:id=*"descriptionButton"* layoutX=*"764.0"* layoutY=*"460.0"* prefHeight=*"26.0"* prefWidth=*"207.0"* promptText=*"Description"* />

<TextField fx:id=*"startTimeButton"* layoutX=*"766.0"* layoutY=*"340.0"* prefHeight=*"26.0"* prefWidth=*"207.0"* promptText=*"Start time (HH:MM)"* />

<TextField fx:id=*"venueButton"* layoutX=*"764.0"* layoutY=*"422.0"* prefHeight=*"26.0"* prefWidth=*"207.0"* promptText=*"Venue"* />

<TextField fx:id=*"durationButton"* layoutX=*"766.0"* layoutY=*"382.0"* prefHeight=*"26.0"* prefWidth=*"207.0"* promptText=*"Duration"* />

<TextField fx:id=*"SearchField"* layoutX=*"764.0"* layoutY=*"111.0"* onKeyReleased=*"#SearchBtn"* prefHeight=*"26.0"* prefWidth=*"211.0"* promptText=*"Date"* />

</children>

</AnchorPane>

<Label fx:id=*"joobleButton"* layoutX=*"573.0"* layoutY=*"6.0"* opacity=*"0.84"* prefHeight=*"51.0"* prefWidth=*"135.0"* text=*"PLANNER"* textFill=*"#a89e9e"*>

<font>

<Font name=*"Calibri"* size=*"35.0"* />

</font>

</Label>

</children>

</AnchorPane>

</children>

</AnchorPane>

**ВАРІАНТИ ВИКОРИСТАННЯ**

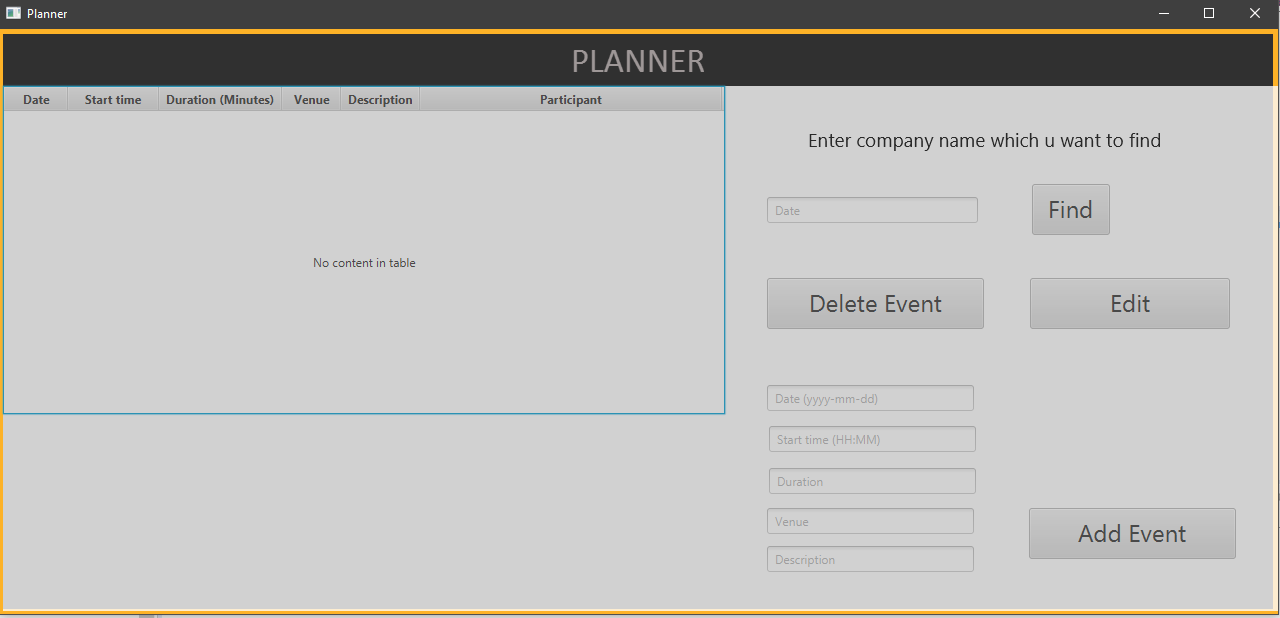


Рисунок 1 – Головне меню програми(вікно в якому відбувається додавання, видалення, редагування та пошук елементів у контейнері.)

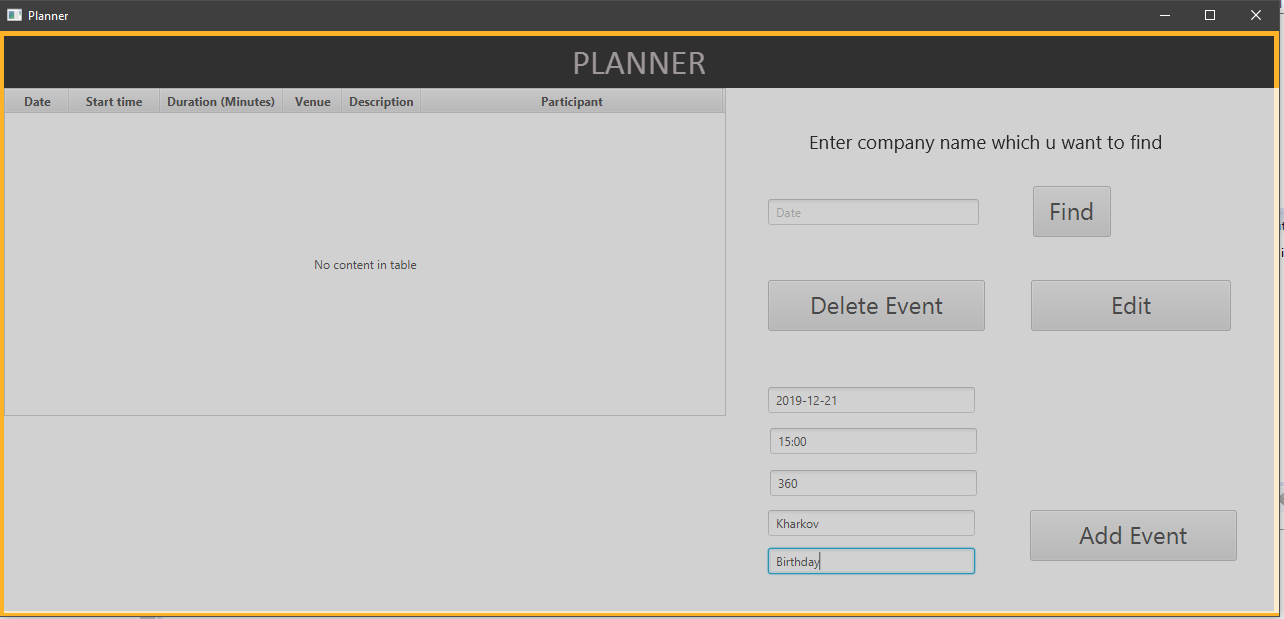


Рисунок 2 – Додавання елементу в контейнер

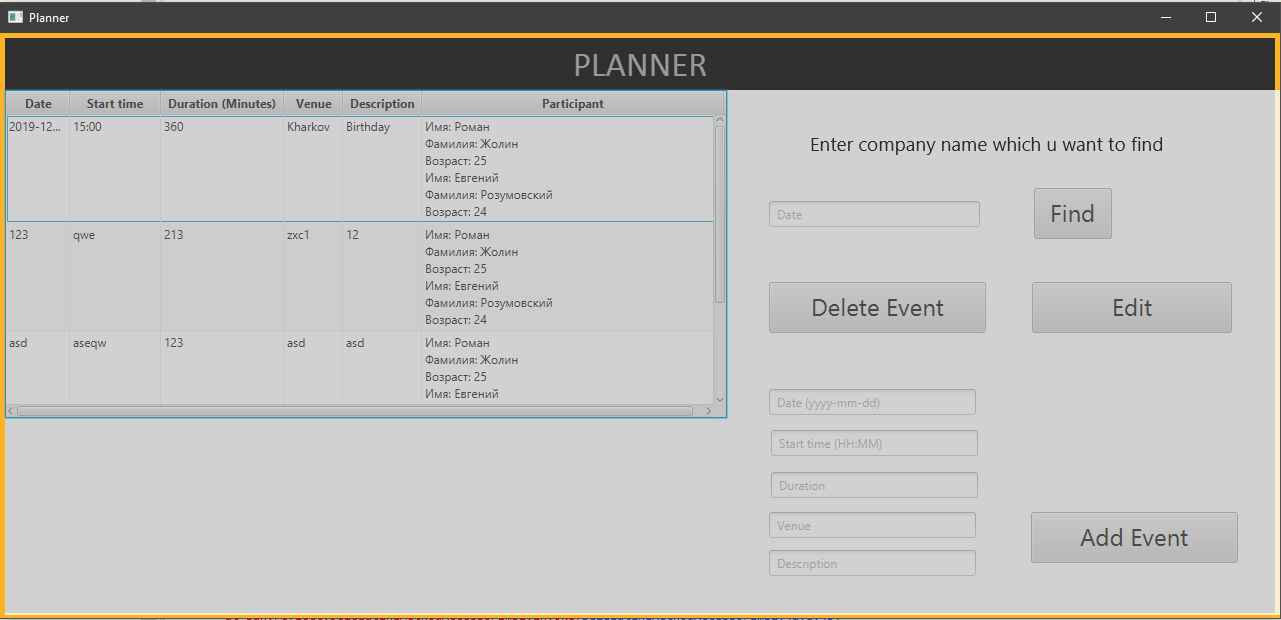


Рисунок 4 – Результат додавання

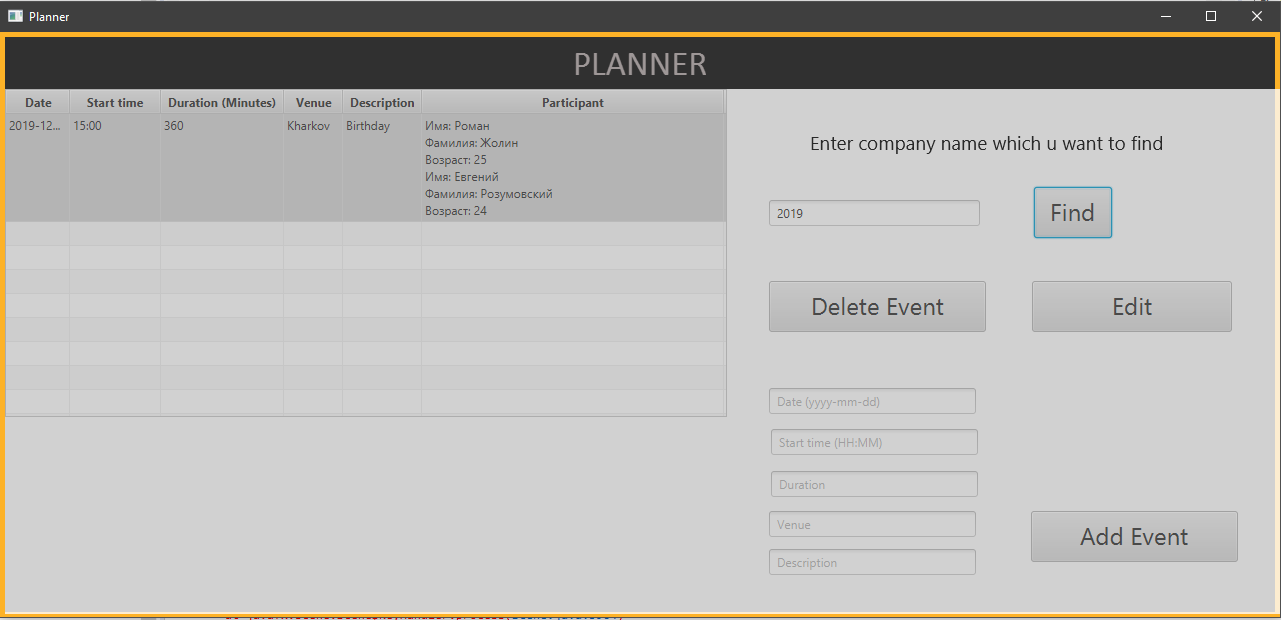


Рисунок 5 – Пошук елементів за заданою датою

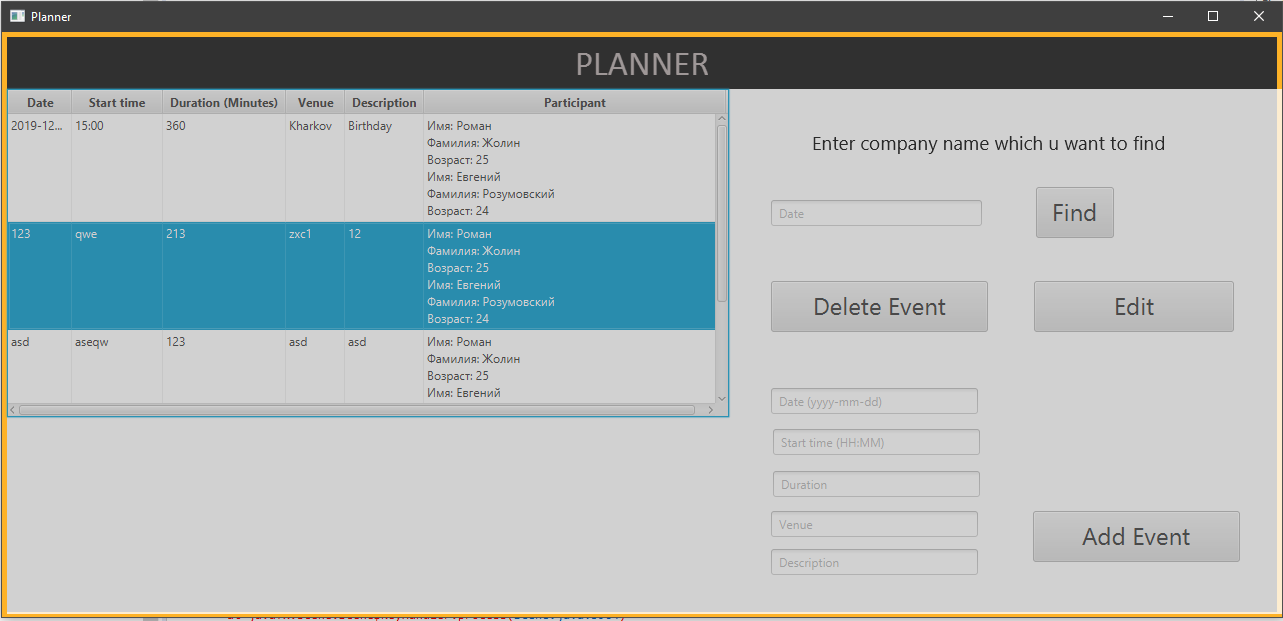


Рисунок 6 – Видалення елементу

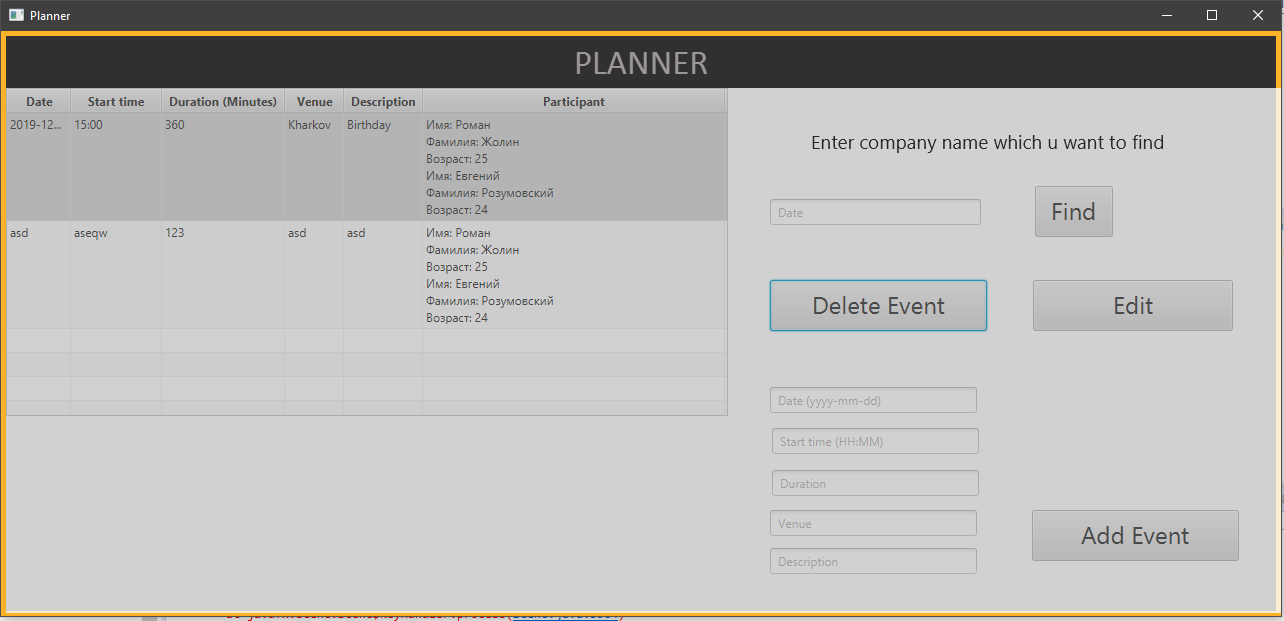


Рисунок 7 – результат видалення елементу

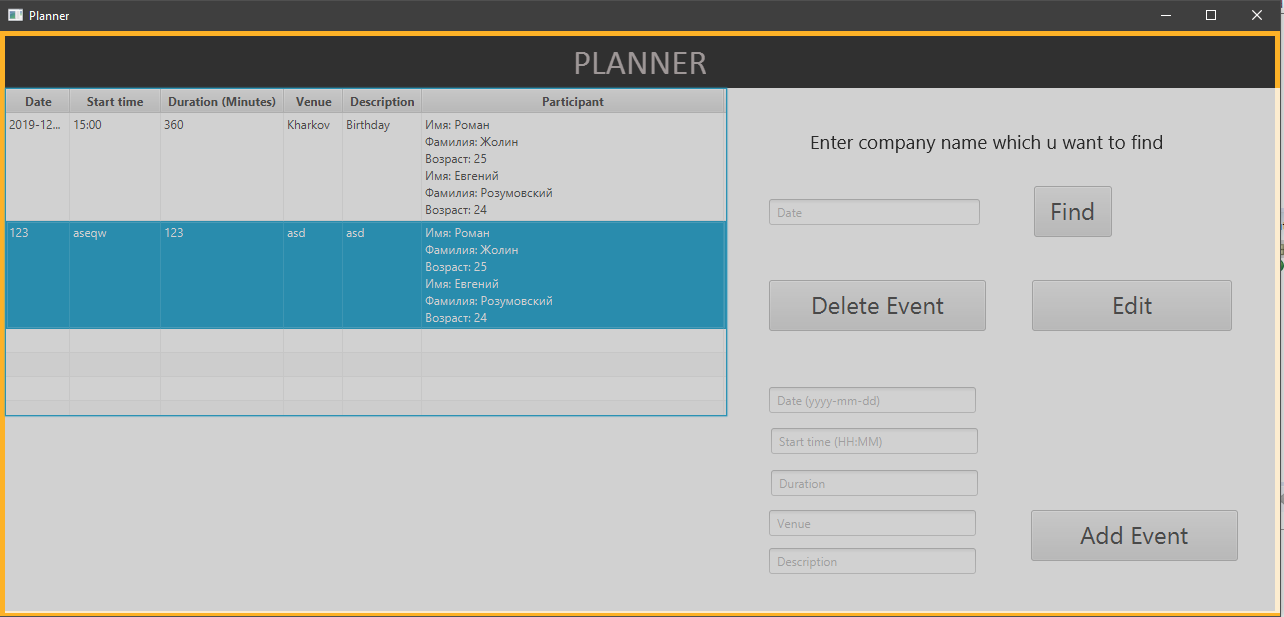


Рисунок 8 – дата 2 елементу контейнеру була відредагована.

Програму можна використовувати задля створення бази даних. Завдяки параметризації зв’язного списку, базу даних можна використати для будь-яких типів даних. Переважно у нашому варіанті - планувальник, в якому представляються різноманітні вакансії. Також для вибору доступно багато інших можливостей. Реалізовано меню для поліпшення користування програмою.

**ВИСНОВКИ**

При виконанні лабораторної роботи набуто практичних навичок щодо розробки параметризованих класів. Завдяки цієї можливості в JAVA, можливо створювати колекції та інші класи на основі будь-яких типів. Також навчився обробляти параметризовані контейнери.