

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления
Кафедра Интеллектуальных информационных технологий

ОТЧЕТ
к лабораторной работе 2
по дисциплине «Проектирование программного обеспечения
интеллектуальных систем»

Выполнил:
Проверил:

И.Д.Кондратьева
С. В. Бутрин

Минск 2023

Цель: Изучить построение графического пользовательского интерфейса с использованием библиотеки Kivy (<https://kivy.org/>)

Задача: разработать оконное приложение с одним главным окном и несколькими дочерними диалогами. Вызов диалогов осуществляется через соответствующие пункты меню. Команды меню должны дублироваться на панели инструментов.

Вариант:

Вариант 13

Номер поезда	Станция отправления	Станция прибытия	Дата и время отправления	Дата и время прибытия	Время в пути
					(не вводится руками, а вычисляется автоматически)

Условия поиска и удаления:

- по номеру поезда или дате отправления;
- по времени отправления (задается верхний и нижний предел) или времени прибытия (задается верхний и нижний предел);
- по станции отправления или станции прибытия;
- по времени в пути

В приложении имеются 4 главных экрана, позволяющие выполнять следующие операции:

- посмотреть таблицу
- найти отдельные записи таблицы по отдельным параметрам (по номеру поезда, станции отправления, станции прибытия, минимальной и максимальной дате и времени отправления, минимальной и максимальной дате и времени прибытия, времени в пути)
- удаление по отдельным параметрам (по номеру поезда, станции отправления, станции прибытия, минимальной и максимальной дате и времени отправления, минимальной и максимальной дате и времени прибытия, времени в пути)
- добавление новых записей (номер поезда, станция отправления, станция прибытия, дата и время отправления, дата и время прибытия)

Переключение между экранами осуществляется с помощью панели инструментов, реализованной с помощью объекта класса ActionBar.

Примеры работы приложения:

MyApp

Table

Search

Delete

Add

No.	Train	Departure station	Arrival station	Departure time	Arrival time	Travel time
1	874B	Minsk	Gomel	25.10 13:15	25.10 16:08	2 hours 53 minutes
2	201C	Moskow	Vitebsk	14.09 22:56	15.09 03:45	4 hours 49 minutes
3	888	Grodno	New York	11.03 11:32	01.05 16:01	1 month 21 days 4 hours 29 minutes
4	314A	Bereza	Rogochov	13.07 15:06	13.07 17:12	2 hours 6 minutes
5	404B	Mogilev	Peterburg	01.02 04:14	02.02 22:56	1 day 18 hours 42 minutes

Rows per page

5

1-8 of 5

<

>

Рис. 1: Экран с таблицей

MyApp

Table

Search

Delete

Add

Номер поезда

Станция отправления

Станция прибытия

Минимальная дата и время отправления

Максимальная дата и время отправления

Минимальная дата и время прибытия

Максимальная дата и время прибытия

Время в пути

Search

Рис. 2: Экран поиска

MyApp

TableSearchDeleteAdd

Номер поезда

Станция отправления

Станция прибытия

Дата и время отправления

Дата и время прибытия

add

Рис. 3: Экран добавления новых записей в таблицу

Возможность переключения между экранами была реализована с помощью объекта класса ScreenManager.

Таблица с данными выводится с помощью объекта класса MDDDataTable. Записи в таблице записаны на разных страницах, которые можно листать.

```
self.data_tables = MDDDataTable(  
    use_pagination=True,  
    rows_num = 8,  
    elevation=7,  
    column_data=[  
        ("[font=Comic][color=#cc74f2]No.[/color][[/font]", dp(7), None, "Custom tooltip"),  
        ("[font=Comic][color=#be51ed]Train[/color][[/font]", dp(12)),  
        ("[font=Comic][color=#aa1ae8]Departure station[/color][[/font]", dp(21)),  
        ("[font=Comic][color=#8900c4]Arrival station[/color][[/font]", dp(21)),  
        ("[font=Comic][color=#6e1d91]Departure time[/color][[/font]", dp(23)),  
        ("[font=Comic][color=#5b117a]Arrival time[/color][[/font]", dp(21)),  
        ("[font=Comic][color=#2f0342]Travel time[/color][[/font]", dp(40))],  
    row_data=sax.work_parser(),  
    self.root.ids.data_scr.ids.data_layout.add_widget(self.data_tables)
```

Рис. 4: Описание таблицы

Данные таблицы хранятся в xml файле, чтение осуществляется с помощью SAX парсера.

```
<info>  
  <train>  
    <number_of_train>8748</number_of_train>  
    <departure>Minsk</departure>  
    <arrival>Gomel</arrival>  
    <departure_data_time>25.10 13:15</departure_data_time>  
    <arrival_data_time>25.10 16:08</arrival_data_time>  
  </train>  
  <train>  
    <number_of_train>2010</number_of_train>  
    <departure>Moscow</departure>  
    <arrival>Vitebsk</arrival>  
    <departure_data_time>14.09 22:56</departure_data_time>  
    <arrival_data_time>15.09 03:45</arrival_data_time>  
  </train>  
  <train>  
    <number_of_train>888</number_of_train>  
    <departure>Grodno</departure>  
    <arrival>New York</arrival>  
    <departure_data_time>11.03 11:32</departure_data_time>  
    <arrival_data_time>01.05 16:01</arrival_data_time>  
  </train>  
  <train>  
    <number_of_train>314A</number_of_train>  
    <departure>Bereza</departure>  
    <arrival>Rogochov</arrival>  
    <departure_data_time>13.07 15:06</departure_data_time>  
    <arrival_data_time>13.07 17:12</arrival_data_time>  
  </train>  
  <train>  
    <number_of_train>4048</number_of_train>  
    <departure>Mogilev</departure>  
    <arrival>Peterburg</arrival>  
    <departure_data_time>01.02 04:14</departure_data_time>  
    <arrival_data_time>02.02 22:56</arrival_data_time>  
  </train>  
</info>
```

```

class TableHandler(sax.ContentHandler):
    def __init__(self):
        super(TableHandler, self).__init__()
        self.union = []
        self.num = "1"
        self.train = [self.num]

    def startElement(self, name, attrs):
        self.current = name

    def characters(self, content):
        if self.current == "number_of_train":
            self.number_of_train = content
        elif self.current == "departure":
            self.departure = content
        elif self.current == "arrival":
            self.arrival = content
        elif self.current == "departure_data_time":
            self.departure_data_time = content
        elif self.current == "arrival_data_time":
            self.arrival_data_time = content

    def endElement(self, name):
        if self.current == 'number_of_train':
            self.train.append(self.number_of_train)
        elif self.current == 'departure':
            self.train.append(self.departure)
        elif self.current == 'arrival':
            self.train.append(self.arrival)
        elif self.current == 'departure_data_time':
            self.train.append(self.departure_data_time)
        elif self.current == 'arrival_data_time':
            self.train.append(self.arrival_data_time)
            travel_time = time.time_of_travel(time.to_date(self.departure_data_time), time.to_date(self.arrival_data_time))
            self.train.append(time.from_time_to_string(travel_time))
        self.current = ''
        if len(self.train) == 7:
            self.union.append(self.train)
            self.num = str(int(self.num)+1)
            self.train=[self.num]

```