

Министерство образования Республики Беларусь  
Учреждение образования  
“Белорусский государственный университет информатики и радиоэлектроники”

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ  
Кафедра интеллектуальных информационных технологий

Отчёт по лабораторной работе №2  
Проектирование программного обеспечения в интеллектуальных системах

Выполнил: ст. гр. 121701  
Воронцов Р.Г.

Проверил: Бутрин С.В.

Минск 2023

**Цель:** изучить построение графического пользовательского интерфейса с использованием библиотеки Kivy(KivyMD)

## Вариант 5

ФИО пациента	Адрес прописки	Дата рождения	Дата приема	ФИО врача	Заключение

Условия поиска и удаления:

- фамилии пациента или адресу прописки;
- по дате рождения;
- по ФИО врача или дате последнего приема (может быть заполнен только один элемент ФИО, например имя).

### Ход работы:

Графическое меню:

```
348 def build(self):
349     self.theme_cls.theme_style = "Light"
350     self.theme_cls.primary_palette = "Blue"
351
352     button_box = MDBoxLayout(
353         padding="10dp",
354         spacing="5dp",
355     )
356
357     for button_text in ["+", "-", "Read / Write", "Find / Delete"]:
358         button_box.add_widget(
359             MDRaisedButton(
360                 text=button_text, on_release=self.on_button_press
361             )
362         )
363
364     layout = AnchorLayout()
365     self.data_tables = MDDDataTable(
366         size_hint=(1, 1),
367         use_pagination=True,
368         rows_num=6,
369         column_data=[
370             ("Patient's name", dp(25)),
371             ("Address", dp(25)),
372             ("Birth date", dp(25)),
373             ("Appointment date", dp(25)),
374             ("Doctor's name", dp(25)),
375             ("Conclusion", dp(25))
376         ]
377     )
378
379     layout.add_widget(self.data_tables)
380     layout.add_widget(button_box)
381     return layout
```

Добавление строк и добавление значений в строку:

```
259 def form_to_add_new_row(self):
260     self.dialog = MDDialog(
261         title="Enter data:",
262         type="custom",
263         content_cls=MDBoxLayout(
264             MDTextField(
265                 id="Patient's name",
266                 hint_text="Patient's name",
267             ),
268             MDTextField(
269                 id="Address",
270                 hint_text="Address",
271             ),
272             MDTextField(
273                 id="Birth date",
274                 hint_text="Birth date",
275                 max_text_length=10,
276             ),
277             MDTextField(
278                 id="Appointment date",
279                 hint_text="Appointment date",
280             ),
281             MDTextField(
282                 id="Doctor's name",
283                 hint_text="Doctor's name",
284             ),
285             MDTextField(
286                 id="Conclusion",
287                 hint_text="Conclusion",
288             ),
289             orientation="vertical",
290             spacing="8dp",
291             size_hint_y=None,
292             height="400dp",
293         ),
```

```
294         buttons=[
295             MDFlatButton(
296                 text="Cancel",
297                 theme_text_color="Custom",
298                 text_color=self.theme_cls.primary_color,
299                 on_release=self.on_button_press,
300             ),
301             MDFlatButton(
302                 text="Add",
303                 theme_text_color="Custom",
304                 text_color=self.theme_cls.primary_color,
305                 on_release=self.on_button_press,
306             ),
307         ],
308     )
309     self.dialog.open()
```

```
245 def add_row(self):
246     item = self.get_dialog_data()
247     self.data_tables.add_row((item["Patient's name"].text, item["Address"].text,
248                             item["Birth date"].text, item["Appointment date"].text,
249                             item["Doctor's name"].text, item["Conclusion"].text))
250     self.dialog.dismiss()
```

Удаление строк:

```
252     def remove_row(self):
253         if len(self.data_tables.row_data) > 0:
254             self.data_tables.remove_row(self.data_tables.row_data[-1])
255
```

Поиск и вывод информации по заданным критериям:

```
38     def search_and_output_rows(self):
39         data = self.get_rows()
40         filled_fields = self.get_dialog_data()
41         section_for_search = field_for_search = None
42         counter = 0
43         self.new_data = []
44         for field in filled_fields:
45             if filled_fields[field].text:
46                 section_for_search = field
47                 field_for_search = filled_fields[field].text
48                 break
49         for item in data["items"]:
50             for section in item:
51                 if section == section_for_search:
52                     if item[section] == field_for_search:
53                         self.new_data.append(item)
54                         counter += 1
55                     else:
56                         continue
57         self.new_data = [list(i.values()) for i in self.new_data]
58         self.counter = counter
59         self.close_dialog()
60         self.info_about_searched_rows()
```

Поиск и удаление по заданным критериям:

```
82     def search_and_delete_rows(self):
83         data = self.get_rows()
84         filled_fields = self.get_dialog_data()
85         section_for_search = field_for_search = None
86         counter = 0
87         new_data = {}
88         new_data["items"] = []
89         for field in filled_fields:
90             if filled_fields[field].text:
91                 section_for_search = field
92                 field_for_search = filled_fields[field].text
93                 break
94         for item in data["items"]:
95             for section in item:
96                 if section == section_for_search:
97                     if item[section] == field_for_search:
98                         counter += 1
99                         continue
100                     else:
101                         new_data["items"].append(item)
102         self.clear_data_base()
103         for item in new_data["items"]:
104             self.data_tables.add_row((item["Patient's name"], item["Address"],
105                                     item["Birth date"], item["Appointment date"],
106                                     item["Doctor's name"], item["Conclusion"]))
107         self.close_dialog()
108         self.info_about_deleted_rows(counter)
```

Чтение и запись с файла:

```
200     def read(self):
201         self.clear_data_base()
202         file_name = self.get_dialog_data()
203         if os.path.isfile(file_name["File Name"].text):
204             json_file = open(file_name["File Name"].text, "r")
205             data_base = j.load(json_file)
206             for item in data_base["items"]:
207                 self.data_tables.add_row((item["Patient's name"], item["Address"],
208                                         item["Birth date"], item["Appointment date"],
209                                         item["Doctor's name"], item["Conclusion"]))
210             json_file.close()
211             self.dialog.dismiss()
212         else:
213             self.error_file_dialog()
214
215     def write(self):
216         file_name = self.get_dialog_data()
217         with open(file_name["File Name"].text, "w") as outfile:
218             j.dump(self.get_rows(), outfile, sort_keys=True, indent=4)
219         self.dialog.dismiss()
```