

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
по дисциплине «Проектирование программного обеспечения
интеллектуальных систем»

Выполнил Маевский В. Г.
студент группы
121701

Проверил Бутрин С. В.

Минск 2023

Цель:Изучить построение графического пользовательского интерфейса с использование библиотеки Kivy(KivyMD)

Вариант 6

Название товара	Название производителя	УНП производителя	Количество на складе	Адрес склада
			Число или текст нет на складе	

Условия поиска и удаления:

- по названию товара или количеству на складе;
- названию производителя или УНП производителя
- по адресу склада;

Выполнение

Реализованные функции

- Графическое меню

```
def build(self):
    self.theme_cls.theme_style = "Dark"
    self.theme_cls.primary_palette = "Purple"

    button_box = MDBoxLayout(
        padding="10dp",
        spacing="5dp",
    )

    for button_text in ["+", "-", "Read / Write", "Search / Delete"]:
        button_box.add_widget(
            MDRaisedButton(
                text=button_text, on_release=self.on_button_press
            )
        )

    layout = AnchorLayout()
    self.data_tables = MDDDataTable(
        size_hint=(1, 1),
        use_pagination=True,
        rows_num=5,
        column_data=[
            ("Product Name", dp(60)),
            ("Manufacturer Name", dp(60)),
            ("Manufacturer's UCR", dp(60)),
            ("Quantity In Stock", dp(60)),
            ("Warehouse Address", dp(60)),
        ]
    )

    layout.add_widget(self.data_tables)
    layout.add_widget(button_box)
    return layout
```

- Добавление строк

Для создания формы:

```
def form_to_add_new_row(self):
    self.dialog = MDDialog(
        title="Enter Data:",
        type="custom",
        content_cls=MDBoxLayout(
            MDTextField(
                id="Product Name",
                hint_text="Product Name",
            ),
            MDTextField(
                id="Manufacturer Name",
                hint_text="Manufacturer Name",
            ),
            MDTextField(
                id="Manufacturer's UCR",
                hint_text="Manufacturer's UCR",
                max_text_length=9,
            ),
            MDTextField(
                id="Quantity In Stock",
                hint_text="Quantity In Stock",
            ),
            MDTextField(
                id="Warehouse Address",
                hint_text="Warehouse Address",
            ),
            orientation="vertical",
            spacing="8dp",
            size_hint_y=None,
            height="340dp",
        ),
    )

    buttons=[
        MDFlatButton(
            text="CANCEL",
            theme_text_color="Custom",
            text_color=self.theme_cls.primary_color,
            on_release=self.on_button_press,
        ),
        MDFlatButton(
            text="ADD",
            theme_text_color="Custom",
            text_color=self.theme_cls.primary_color,
            on_release=self.on_button_press,
        ),
    ],
)
self.dialog.open()
```

Непосредственно для добавления значений в строку:

```
def add_row(self):
    item = self.get_dialog_data()
    self.data_tables.add_row((item["Product Name"].text, item["Manufacturer Name"].text,
                               item["Manufacturer's UCR"].text, item["Quantity In Stock"].text,
                               item["Warehouse Address"].text))
    self.dialog.dismiss()
```

- Удаление строк

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

- Чтение с выбранного файла

```
def read_file(self):
    self.clear_data_base()
    file_name = self.get_dialog_data()
    if os.path.isfile(file_name["File Name"].text):
        json_file = open(file_name["File Name"].text, "r")
        data_base = j.load(json_file)
        for item in data_base["items"]:
            self.data_tables.add_row((item["Product Name"], item["Manufacturer Name"],
                                     item["Manufacturer's UCR"], item["Quantity In Stock"],
                                     item["Warehouse Address"]))
        json_file.close()
        self.dialog.dismiss()
    else:
        self.error_file_dialog()
```

- Перезапись выбранного файла

```
def write_file(self):
    file_name = self.get_dialog_data()
    with open(file_name["File Name"].text, "w+") as outfile:
        j.dump(self.get_rows(), outfile)
    self.dialog.dismiss()
```

- Поиск строк по критериям

Производит поиск по заданным критериям и выводит информацию о найденных строках с помощью метода `info_about_searched_rows()`

```
def search_and_output_rows(self):
    data = self.get_rows()
    filled_fields = self.get_dialog_data()
    section_for_search = field_for_search = None
    counter = 0
    self.new_data = []
    for field in filled_fields:
        if filled_fields[field].text:
            section_for_search = field
            if section_for_search == "Quantity In Stock":
                field_for_search = int(filled_fields[field].text)
            else:
                field_for_search = filled_fields[field].text
            break
    for item in data["items"]:
        for section in item:
            if section == section_for_search:
                if item[section] == field_for_search:
                    self.new_data.append(item)
                    counter += 1
            else:
                continue
    self.new_data = [list(i.values()) for i in self.new_data]
    self.counter = counter
    self.close_dialog()
    self.info_about_searched_rows()
```

- Удаление строк по критериям

Производит поиск по заданным критериям и удаляет найденные строки, после чего выводит количество удаленных строк с помощью метода `info_about_deleted_rows()`

```
def search_and_delete_rows(self):
    data = self.get_rows()
    filled_fields = self.get_dialog_data()
    section_for_search = field_for_search = None
    counter = 0
    new_data = {}
    new_data["items"] = []
    for field in filled_fields:
        if filled_fields[field].text:
            section_for_search = field
            if section_for_search == "Quantity In Stock":
                field_for_search = int(filled_fields[field].text)
            else:
                field_for_search = filled_fields[field].text
            break
    for item in data["items"]:
        for section in item:
            if section == section_for_search:
                if item[section] == field_for_search:
                    counter += 1
                    continue
            else:
                new_data["items"].append(item)
    self.clear_data_base()
    for item in new_data["items"]:
        self.data_tables.add_row((item["Product Name"], item["Manufacturer Name"],
                                   item["Manufacturer's UCR"], item["Quantity In Stock"], item["Warehouse Address"]))
    self.close_dialog()
    self.info_about_deleted_rows(counter)
```