

**Белорусский государственный университет информатики и
радиоэлектроники**

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

Проектирование программного обеспечения интеллектуальных систем

ЛАБОРАТОРНАЯ РАБОТА №2

По дисциплине “Проектирование программного обеспечения
интеллектуальных систем”

Выполнил:

Селицкий Б.С.

Проверил:

Бутрин С. В.

Группа:

121701

Минск

2023

Цель: Изучить построение графического пользовательского интерфейса с использованием библиотеки Kivy.

Задание: 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 1:

Условия поиска и удаления:

- по среднему баллу (средний балл задается нижним пределом и верхним пределом) и предмету.
- по номеру группы;
- по баллу (балл задается нижним и верхним пределом) и по предмету.

Описание программы: Программа состоит из 3 основных модулей: main, model, view

Модуль main содержит класс App, в котором описаны внешние параметры главного окна.

```
class App(MDApp):  
  
    def build(self):  
        Window.size = (800, 500)  
        self.theme_cls.theme_style = "Light"  
        self.theme_cls.primary_palette = "Cyan"  
        self.theme_cls.material_style = "M3"  
        self.title = "PP01S 2"  
        sm = Controller()  
        return sm
```

Модуль model содержит класс Model, в котором описаны подклассы Exam (экзамен) и Person (человек), общий принцип работы окна и записи данных.

```
class Model:
    class Exam:
        def __init__(self, name: str, mark: int):
            self.name = name
            self.mark = mark

    class Person:
        def __init__(self, name: str, group: int, identifier: int):
            self.name = name
            self.group = group
            self.identifier = identifier
            self.exams: List[Model.Exam] = []

    def __init__(self):
        self.json_dir = "/db.json"
        self.persons: List[Model.Person] = self.parse_student()

    def save(self):
        students: List = []

        for i in self.persons:
            person: dict = {
                "id": i.identifier,
                "name": i.name,
                "group": i.group,
                "courses": []
            }
            for j in i.exams:
                person["courses"].append({"name": j.name, "mark": j.mark})
```

Модуль view содержит класс RemoveScreen и другие подобные классы, в которых описаны принципы работы отдельных элементов окна .

```
class RemoveScreen(Screen):
    def __init__(self, controller, **kwargs):
        super(RemoveScreen, self).__init__(**kwargs)
        self.controller = controller

        self.students = self.controller.get_student_names()

        self.data_table = Table([
            ("No.", dp(50)),
            ("Name", dp(50)),
            ("Group", dp(50)),
        ], self.students, check=True)

        self.buttons = edit_menu_button_layout(self.controller)
        self.buttons.pos_hint = {'center_x': 0.53, 'center_y': 0.5}

        self.add_widget(self.data_table)
        self.add_widget(self.buttons)
```