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: И.Е.Рыбакова Проверил: С. В. Бутрин

Минск 2023

Цель: Изучить построение графического пользовательского интерфейса с использование 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:

Название турнира	Дата проведени я	Название вида спорта	ФИО победителя	Размер призовых турнира	Заработок победителя (60% от размера призовых)
					Значения поля вычисляется автоматически

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

- посмотреть таблицу
- найти отдельные записи таблицы по отдельным параметрам (по названию турнира, имени победителя, дате проведения, названию вида спорта, по миниимальным и максимальным размерам приза и зароботка победителя)
- удаление по отдельным параметрам
- добавление новых записей

Переключение между экранами осуществляется с помощью панели инструментов, реализованной с помощью объекта класса ActionBar. Примеры работы приложения:

				Table	Search	Delete
No.	Name	Date	Sport	Winner	Prize	Winner's g
9	Man gegen man	15.11.2020	wrestling	Till	404	242
10	Why am I here?	11.05.1999	golf	John	0	0
11	Nothing stands	29.07.2015	bowling	Polina	502	301
12	Red dot	21.06.2019	darts	Zhenya	5000	3000
13	Above the net	03.03.2013	volleyball	Anya	215	129
14	Black circle	04.08.2016	hockey	Aylin	670	402
15	Don't drown	02.04.2001	swimming	William	3500	2100
16	Like a fish	12.11.2019	swimming	Jerry	780	468
			Roy	ws per page 8 ▲	9-16 of 2	24 〈)

Рис. 1: Экран с таблицей

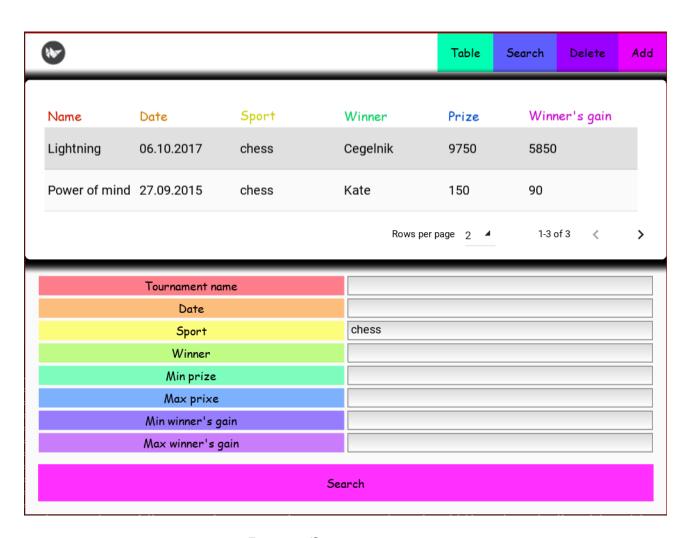


Рис. 2: Экран поиска

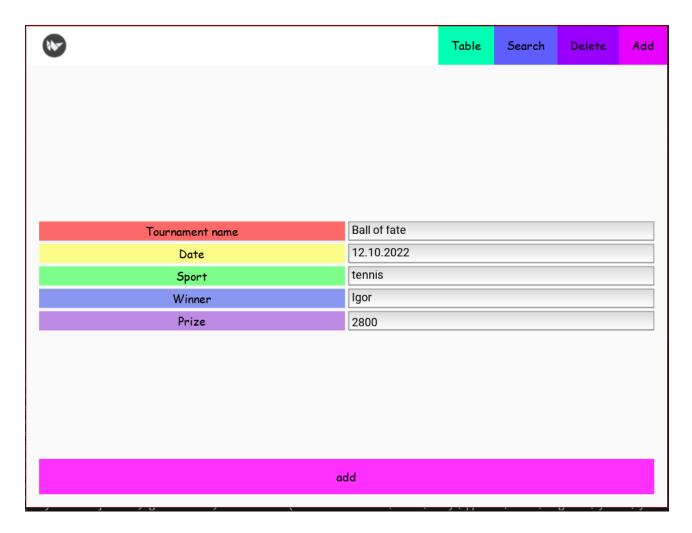


Рис. 3: Экран добавления новых записей в таблицу

```
class WindowManager(ScreenManager):
 pass
```

```
<WindowManager>:
 MenuScreen:
 SearchScreen:
 id: search_scr
 DeleteScreen:
 id: delete_scr
 AddScreen:
 id: add_scr
 DataScreen:
 id: data_scr
```

Возможность переключения между экранами была реализована с помощью объекта класса ScreenManager

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

Рис. 4: Описание таблицы

Данные таблицы хранятся в xml файле, чтение осуществляется с помощью ${\rm SAX}$ парсера.

```
class TableHandler(sax.ContentHandler):
 def init (self):
     super(TableHandler,self). init ()
     self.union = []
     self.num = "1"
     self.tournament = [self.num]
def startElement(self, name, attrs):
     self.current = name
def characters(self, content):
     if self.current == "tournaname":
         self.tournaname = content
     elif self.current == "date":
         self.date = content
     elif self.current == "sport":
         self.sport = content
     elif self.current == "winner":
         self.winner = content
     elif self.current == "prize":
         self.prize = content
 def endElement(self, name):
     if self.current == 'tournaname':
         self.tournament.append(self.tournaname)
     elif self.current == 'date':
         self.tournament.append(self.date)
     elif self.current == 'sport':
         self.tournament.append(self.sport)
     elif self.current == 'winner':
         self.tournament.append(self.winner)
     elif self.current == 'prize':
         self.tournament.append(self.prize)
         self.tournament.append(str(round(float(self.prize)*0.6)))
```