

Санкт-Петербургский колледж информационных технологий

ОТЧЕТ

по приложению "Заметки"

по МДК01.03. «Разработка мобильных приложений»

Специальность 09.02.07

«Информационные системы и программирование»

Специализация:

«Администратор баз данных»

Выполнил:

студент 2-го курса группы 204 Колинько Роман

Преподаватель:

Фомин Александр Валерьевич

Санкт-Петербург, 2022

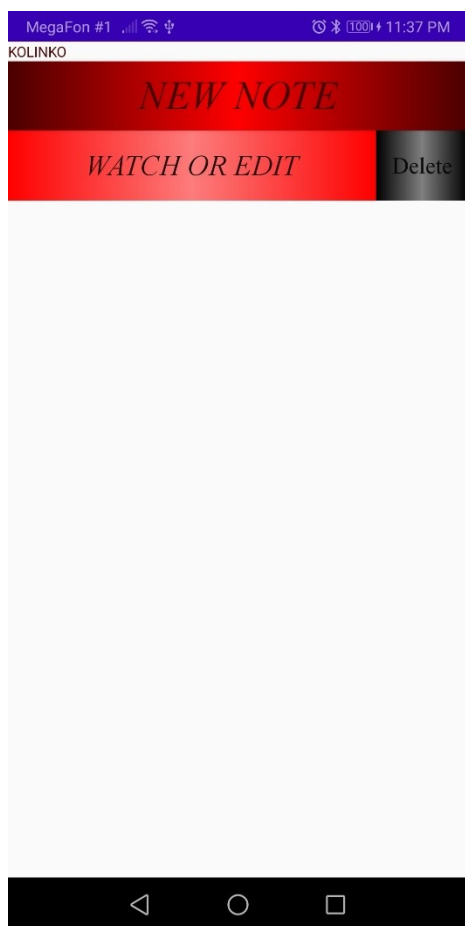
ОПИСАНИЕ

Приложение представляет собой несколько активити, в которых можно создавать, редактировать и удалять заметки.

В последующих работах оно будет улучшено при помощи использования в нем базы данных.

Работа приложения показана в "Приложении" к отчету.

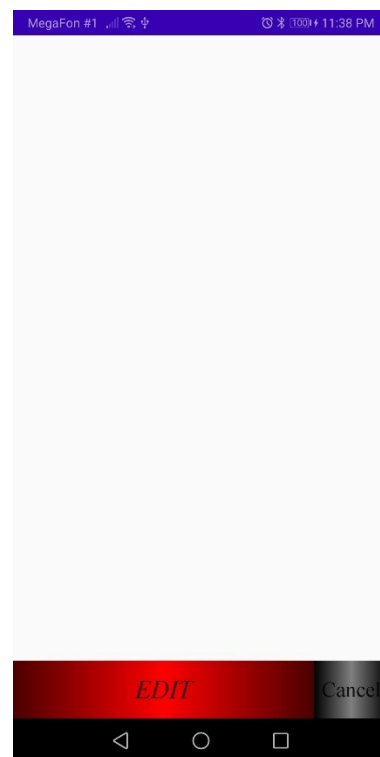
```
package="com.example.romankolinnotes">
```



Страница для выбора
возможности создания новой
заметки/действий с заметкой



Страница создания
заметки



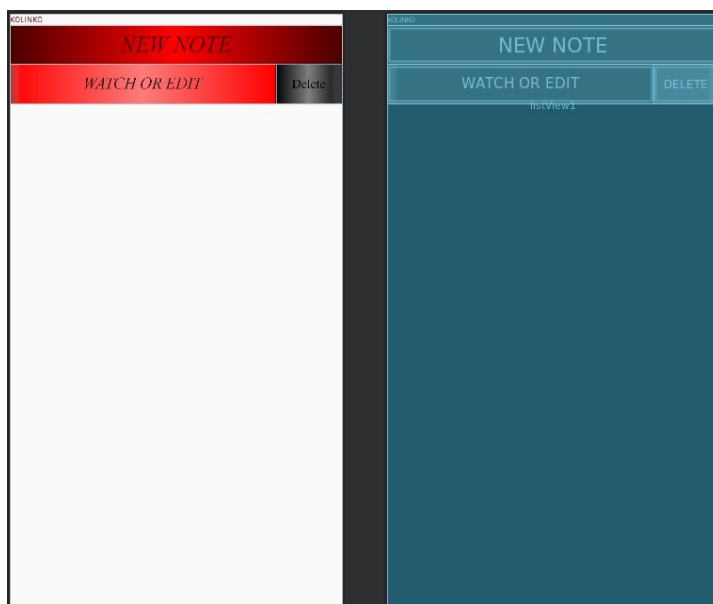
Страница редактирования
заметки

ИНТЕРФЕЙС

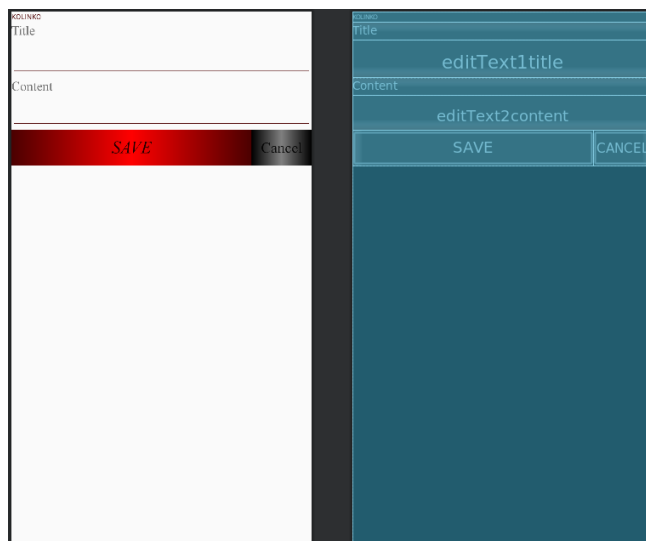
Интерфейс, полностью написанный в XML, представляет собой для первой активности: кнопки создания новой заметки, редактирования и удаления заметки; для второй: два TextView с подписями "Title" и "Content", два EditText для ввода заголовка и содержания заметки, кнопки сохранения и отмены создания заметки; для третьей: два EditText для изменения заголовка и содержания заметки, кнопки сохранения и отмены редактирования заметки.

Элементы сгруппированы в нескольких LinearLayout.

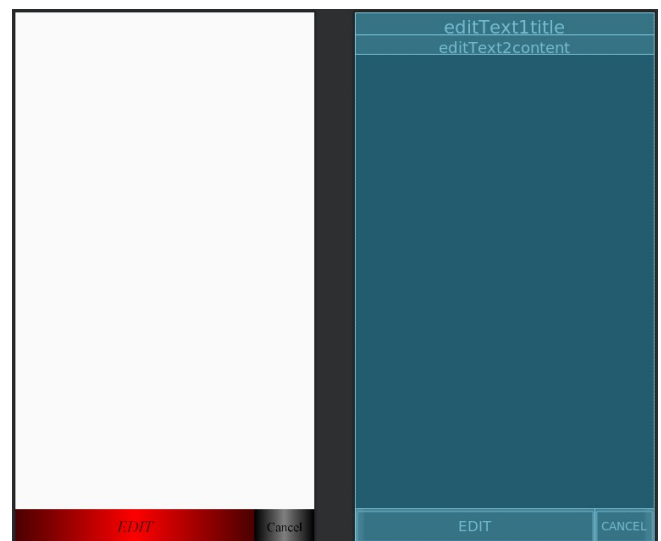
В строковых ресурсах находится отображаемый на элементах текст, а в графических – их цвета (которые были настроены в drawable). Используются шрифты TimesNewRoman и TimesNewRomanItalic.



Первая активности



Вторая активности



Третья активности

```

<!-- KOLINKO-->
    <string name="app_name">Notes</string>
    <string name="creator">KOLINKO</string>
    <string name="newnote">New note</string>
    <string name="watchoredit">Watch or edit</string>
    <string name="edit">Edit</string>
    <string name="delete">Delete</string>
    <string name="title">Title</string>
    <string name="content">Content</string>
    <string name="save">Save</string>
    <string name="cancel">Cancel</string>

```

Строковые ресурсы

```

<!-- KOLINKO-->
    <color name="purple_200">#FFBB86FC</color>
    <color name="purple_500">#FF6200EE</color>
    <color name="purple_700">#FF3700B3</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="maroon">#4B0000</color>
    <color name="red">#FF0000</color>
    <color name="lightred">#80FF0000</color>
    <color name="darkgrey">#808080</color>
    <color name="lightgrey">#AAAAAA</color>

```

Графические ресурсы

```

▼ drawable
  cursor.xml
  deletecancel.xml
  edit.xml
  ic_launcher_background.xml
  ic_launcher_foreground.xml (v24)
  newnotesave.xml

```

Drawable

ЛОГИКА

Заметки формируются при помощи класса "Note", который позволяет создавать заголовок и содержание заметки и на котором основывается адаптер, добавляющий заметки в ListView, что делает каждую заметку элементом списка, благодаря чему при нажатии на любую из заметок программно определяется ее номер, который используется дальше при совершении действия с заметкой (переход на страницу редактирования, удаление).

```
public class Note
{
    //    KOLINKO
    public String title;
    public String content;

    public String toString()
    {
        return title;
    }
}
```

Класс "Note"

```
KOLINKO
note = new ArrayAdapter<Note>(context: this, android.R.layout.simple_list_item_1);
listView1.setAdapter(note);
```

Создание адаптера и добавление его в ListView

```
KOLINKO
listView1.setOnItemClickListener((adapterView, vie, number, l) -> notepos = number);
```

Определение позиции нажатой заметки (в onCreate)

Передача данных между активити осуществляется при помощи Intent. Например, нажимая на первой активити на кнопку создания новой заметки вторая активити принимает позицию заметки, а также пустые поля для заголовка и содержания заметки, после заполнения которых и нажатия на кнопку сохранения вторая активити передает первой содержание этих полей.

Также происходит с редактированием заметки, но поля передаются не пустые (если они, конечно, были чем-то заполнены).

```
public void onclicknewnote(View newnote)
{
    KOLINKO
    Note newnot = new Note();
    newnot.title = "";
    newnot.content = "";
    note.add(newnot);

    Intent newnoteactnew = new Intent( packageContext this, TheNewNoteActivity.class);
    notepos = note.getPosition(newnot);
    newnoteactnew.putExtra( name: "newnoteactnotepos", notepos);
    newnoteactnew.putExtra( name: "newnoteacttitl", newnot.title);
    newnoteactnew.putExtra( name: "newnoteactcont", newnot.content);

    startActivityForResult(newnoteactnew, requestCode: 1);
}
```

Намерение работы с полями ввода текста на второй активити

```
KOLINKO
Intent noteact = getIntent();
notepos = noteact.getIntExtra( name: "newnoteactnotepos", defaultValue: 0);
editText1title.setText(noteact.getStringExtra( name: "newnoteacttitl"));
editText2content.setText(noteact.getStringExtra( name: "newnoteactcont"));
```

Принятие намерения второй активити

```
public void onclicksave(View save)
{
    KOLINKO
    Intent noteactsav = new Intent();
    noteactsav.putExtra( name: "noteactnotepos", notepos);
    noteactsav.putExtra( name: "noteacttitl", editText1title.getText().toString());
    noteactsav.putExtra( name: "noteactcont", editText2content.getText().toString());

    setResult(RESULT_OK, noteactsav);
    finish();
}
```

Передача данных на первую активити

```
if (rescod == -1)
{
    KOLINKO
    notepos = newnoteact.getIntExtra( name: "noteactnotepos", defaultValue: 0);
    String titl = newnoteact.getStringExtra( name: "noteacttitl");
    String cont = newnoteact.getStringExtra( name: "noteactcont");

    Note newnot = note.getItem(notepos);
    newnot.title = titl;
    newnot.content = cont;
}
```

Принятие данных первой активити

Удаление заметки происходит при помощи нажатия на кнопку и подтверждения действия в диалоговом окне.

```
if (notepos >= 0)
{
    KOLINKO
    AlertDialog.Builder del = new AlertDialog.Builder( context: TheNoteActivity.this).setTitle("Note's delete");
    del.setMessage("Do you wanna delete this note?");
    del.setPositiveButton( text: "No", ((dialogInterface, n) -> dialogInterface.cancel());
    del.setNegativeButton( text: "Yes", ((dialogInterface, y) ->
    {
        Note delnote = note.getItem(notepos);
        note.remove(delnote);
        note.notifyDataSetChanged();

        notepos = -1;
    }));
    del.show();
}
```

ПРИЛОЖЕНИЕ

