|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети**

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 **Прикладная информатика**

Android приложение «Помощник учителя»

Фрагмент исходного кода

Листов 6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-82Б |  |  | И.С. Марчук |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Руководитель |  |  |  | М.М. Фомин |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

*2023 г.*

**Фрагмент программы из файла DataBaseOpenHelper.java, отвечающий за работу с таблицей групп учеников:**

//класс

public long createClass(String name) {

SQLiteDatabase db = this.getWritableDatabase();

ContentValues values = new ContentValues(); values.put(SchoolContract.TableClasses.COLUMN\_CLASS\_NAME

, name);

long temp =

db.insert(SchoolContract.TableClasses

.NAME\_TABLE\_CLASSES, null, values

return temp;

}

public Cursor getLearnersClases() {

SQLiteDatabase db = this.getReadableDatabase();

Cursor cursor = db.query(

SchoolContract.TableClasses.NAME\_TABLE\_CLASSES,

null, null, null, null, null, null

);

return cursor;

}

public Cursor getLearnersClases(long classId) {

SQLiteDatabase db = this.getReadableDatabase();

Cursor cursor = db.query(

SchoolContract.TableClasses.NAME\_TABLE\_CLASSES,

null,

SchoolContract.TableClasses.KEY\_ROW\_ID + " = " +

classId,

null, null, null, null);

return cursor;

}

public int setClassName(long classId, String name) {

SQLiteDatabase db = this.getReadableDatabase();

ContentValues contentName = new ContentValues();

contentName.put(

SchoolContract.TableClasses.COLUMN\_CLASS\_NAME,

name);

int answer = db.update(

SchoolContract.TableClasses.NAME\_TABLE\_CLASSES, contentName,

SchoolContract.TableClasses.KEY\_ROW\_ID + " = " +

classId,

null);

return answer;

}

public int deleteClass(long classesId) {

SQLiteDatabase db = this.getReadableDatabase();

int answer = db.delete(

SchoolContract.TableClasses.NAME\_TABLE\_CLASSES,

SchoolContract.TableClasses.KEY\_ROW\_ID + " = " +

classesId,

null);

return answer;

}

**Фрагмент описания таблицы ученических групп из файла Schoolcontract.java, описывающего структуру БД:**

public static class TableClasses extends Table {

public static final String NAME\_TABLE\_CLASSES =

"classes";

public static final String COLUMN\_CLASS\_NAME =

"className";

public static final String CREATE\_TABLE\_STRING =

"CREATE TABLE " + SchoolContract.TableClasses.NAME\_TABLE\_CLASSES

+" ( " + SchoolContract.TableClasses.KEY\_ROW\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " + SchoolContract.TableClasses.COLUMN\_CLASS\_NAME + " VARCHAR );";

}

**Фрагмент, описывающий работу диалогового окна для редактирования группы учеников из файла EditLearnersClassDialogFragment.java:**

public class EditLearnersClassDialogFragment extends

DialogFragment {

@Override

public Dialog onCreateDialog(Bundle savedInstanceState){

// начинаем строить диалог

android.app.AlertDialog.Builder builder =

new android.app.AlertDialog.Builder(getActivity());

// layout диалога

View dialogLayout = requireActivity()

.getLayoutInflater().inflate(

R.layout.learners\_classes\_out\_dialog\_edit\_class, null);

builder.setView(dialogLayout);

// кнопка отмены

View neutralButton = dialogLayout

.findViewById(R.id

.learners\_classes\_out\_dialog\_edit\_class\_button\_cancel);

neutralButton.setOnClickListener(view -> dismiss());

// текстовое поле имени

final EditText editName = dialogLayout.

findViewById(R.id.

learners\_classes\_out\_dialog\_edit\_class\_name\_input);

// входные данные предыдущее название

editName.setText(requireArguments().getString("name"));

//при нажатии...

// согласие

dialogLayout.findViewById(

R.id.learners\_classes\_out\_dialog\_edit\_class\_button\_save)

.setOnClickListener(view -> {

if (editName.getText().toString().trim().length()

== 0) {

Toast.makeText(getActivity(), R.string.learners\_classes\_out\_activity\_toast\_empty\_name, Toast.LENGTH\_SHORT).show();

} else {

//вызываем в активности метод по созданию

// класса и передаем ей имя

((EditLearnersClassDialogInterface)

getActivity())

.editLearnersClass(

editName.getText().toString().trim()

);

dismiss();

}

});

final AppCompatActivity context =

(AppCompatActivity) getActivity();

//удаление

dialogLayout.findViewById(R.id

.learners\_classes\_out\_dialog\_edit\_class\_button\_remove)

.setOnClickListener(view -> {

// вызываем в активности метод по удалению

// класса и передаем id

((EditLearnersClassDialogInterface)

getActivity()).removeLearnersClass();

dismiss();

});

// наконец создаем диалог и возвращаем его

Dialog dialog = builder.create();

dialog.getWindow().setBackgroundDrawableResource(

android.R.color.transparent);

return dialog;

}

}

interface EditLearnersClassDialogInterface {

void editLearnersClass(String name);

void removeLearnersClass();

}