МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра Математического обеспечения и стандартизации информационных технологий

Отчет по выполнению практического задания №6

Тема:
Хранение данных

Дисциплина:
Разработка мобильных приложений

Выполнил:
Студент группы ИКБО-03-20 Цемкало А.Р.

Принял:
Доцент кафедры МОСИТ ИИТ Чернов Е. А.

Москва 2022 г.

СОДЕРЖАНИЕ

Ход работы

Хранение наборов ключ-значение

```
sharedPreferences = getSharedPreferences(getString(R.string.preference_file_key), Context.MODE_PRIVATE);
```

Рисунок 1 – Получение дескриптора SharedPreferences

```
@Override
protected void onResume() {
    super.onResume();
    SharedPreferences.Editor editor = sharedPreferences.edit();
    editor.putInt( s: "resumes",  i: sharedPreferences.getInt( s: "resumes",  i: 0) + 1);
    editor.commit();
}
```

Рисунок 2 – Запись пар ключ-значение

Хранение файлов

activity_main.xml | MainActivity.java | FirstFragment.java | AndroidManifest.xml | integers.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.task6">

    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

```

Рисунок 3 – Получение прав для внешнего хранилища

```
saveOnDeviceButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String text = editTextDevice.getText().toString();
        String fileName = "deviceFile.txt";
        File file = new File(getFilesDir(),fileName);
        FileOutputStream fileOutputStream;
        try {
            fileOutputStream = openFileOutput(fileName,Context.MODE_PRIVATE);
            fileOutputStream.write(text.getBytes());
            fileOutputStream.close();
            System.out.println("Path to device memory: " + getFilesDir());
            System.out.println("Free place left on device: " + getFilesDir().getTotalSpace());
        } catch (IOException ioException){
            return;
        }
    }
});
```

Рисунок 4 – Реализация сохранения файлов во внутреннем хранилище

```
saveOnExternalButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (isExternalStorageReadable() && isExternalStorageWritable()){
            File root = getExternalFilesDir(Environment.DIRECTORY_DOCUMENTS);
            System.out.println(root);
            File textFile = new File(root, child: "externalFile.txt");
            try {
                FileOutputStream fos = new FileOutputStream(textFile);
                fos.write(editTextExternal.getText().toString().getBytes());
                fos.flush();
                fos.close();
                System.out.println("Path to external memory: " + getExternalFilesDir( type: null));
                System.out.println("Free place left in external memory: " + root.getTotalSpace());
            } catch (IOException ioException){
                ioException.fillInStackTrace();
            }
        }
    }
});
```

Рисунок 5 – Реализация сохранения файлов во внешнем хранилище

```
I/System.out: Path to external memory: /storage/emulated/0/Android/data/com.example.task6/files
I/System.out: Free place left in external memory: 812531712
I/System.out: Path to device memory: /data/user/0/com.example.task6/files
I/System.out: Free place left on device: 812531712
```

Рисунок 6 – Реализация запроса свободного пространства

```
deleteFromDeviceButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        File files = getFilesDir();
        File delFile = new File(files, child: "deviceFile.txt");
        delFile.delete();
    }
});
```

Рисунок 7 – Реализация удаления файла из внутреннего хранилища

```
deleteFromExternalButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        File root = getExternalFilesDir(Environment.DIRECTORY_DOCUMENTS);
        File textFile = new File(root, child: "externalFile.txt");
        textFile.delete();
    }
});
```

Рисунок 8 – Реализация удаления файла из внешнего хранилища

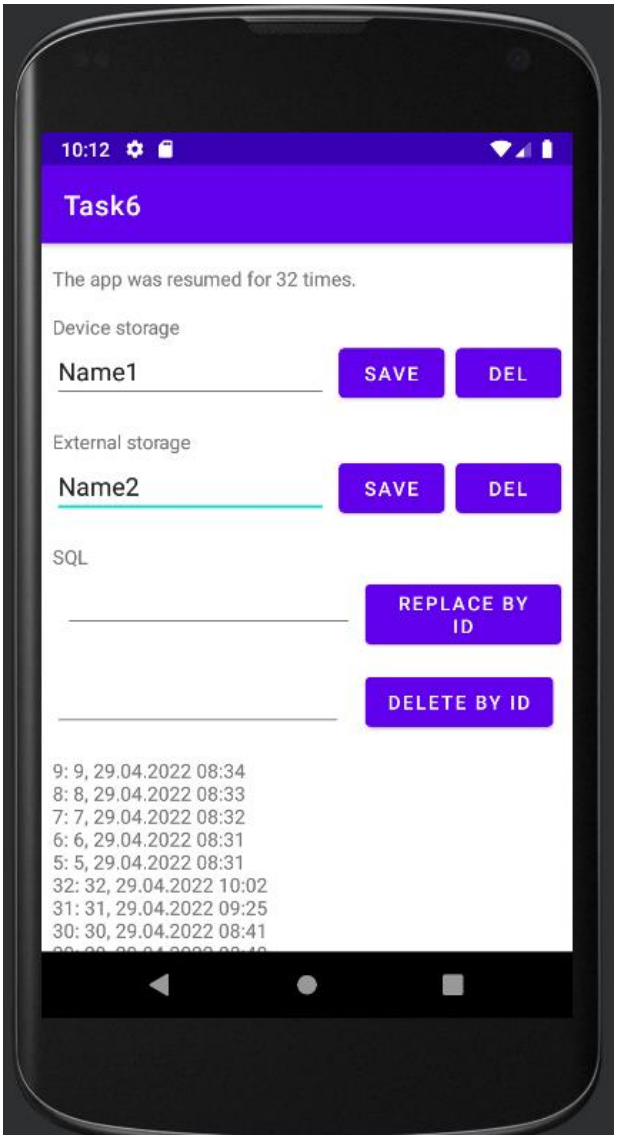

Рисунок 9 – Вид работающего приложения

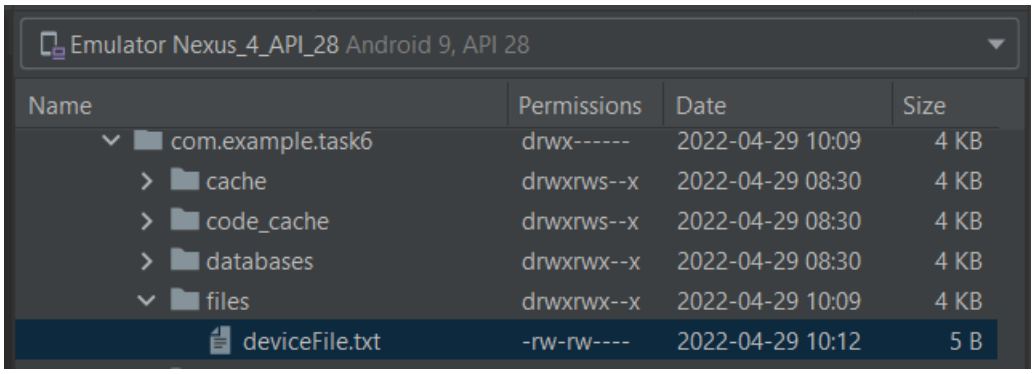

Рисунок 10 – Созданный файл во внутреннем хранилище

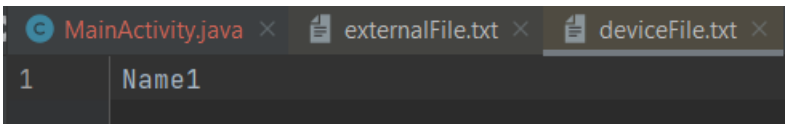

Рисунок 11 – Содержимое созданного файла во внутреннем хранилище

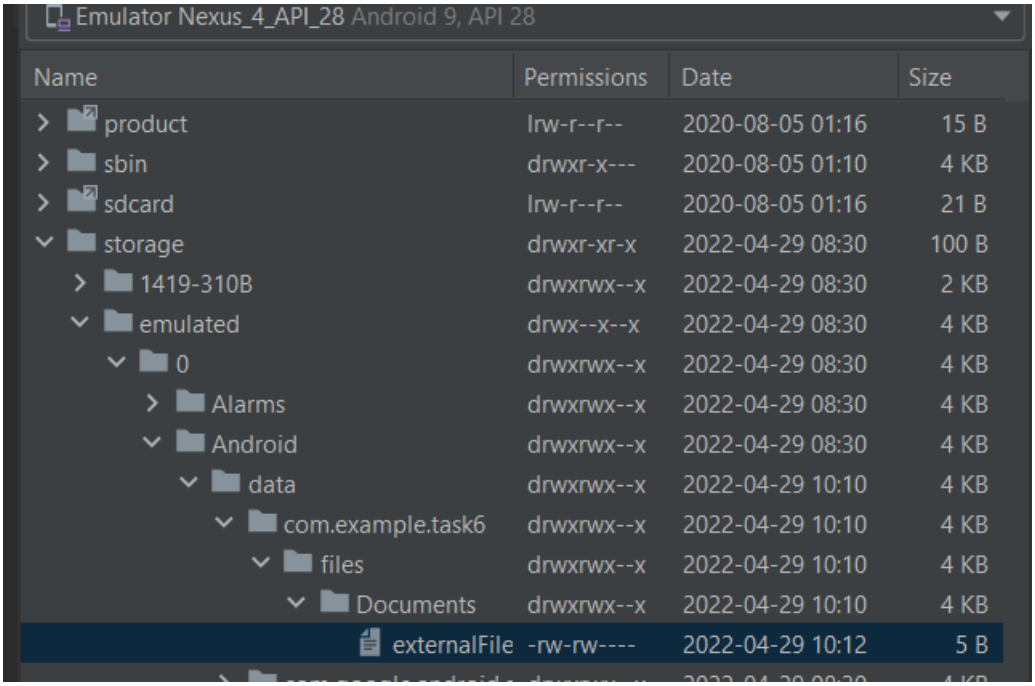

Рисунок 12 – Созданный файл во внешнем хранилище

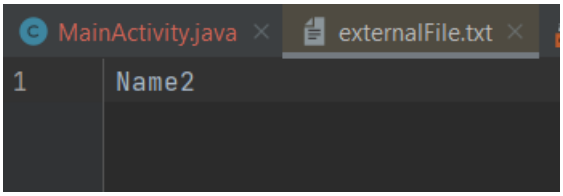

Рисунок 13 – Содержимое созданного файла во внешнем хранилище

Хранение данных в базе данных SQLite

```
import android.provider.BaseColumns;

public class FeedReaderContract {
    public FeedReaderContract() {}

    public static abstract class FeedEntry implements BaseColumns {
        public static final String TABLE_NAME = "resume_records";
        public static final String NUMBER_OF_RESUME = "number";
        public static final String DATE_OF_RESUME = "date";
    }
}
```

Рисунок 14 – Создание схемы и контракт базы данных. Реализация примера описания имени таблицы и столбцов.

```
public class FeedReaderDbHelper extends SQLiteOpenHelper {
    private static final String TEXT_TYPE = " TEXT";
    private static final String COMMA_SEP = ",";
    private static final String SQL_CREATE_ENTRIES = "CREATE TABLE " + FeedReaderContract.FeedEntry.TABLE_NAME + " (" +
            FeedReaderContract.FeedEntry._ID + " INTEGER PRIMARY KEY" + COMMA_SEP +
            FeedReaderContract.FeedEntry.NUMBER_OF_RESUME + TEXT_TYPE + COMMA_SEP +
            FeedReaderContract.FeedEntry.DATE_OF_RESUME + TEXT_TYPE + " )";

    private static final String SQL_DELETE_ENTRIES = "DROP TABLE IF EXISTS " + FeedReaderContract.FeedEntry.TABLE_NAME;
    public static final int DATABASE_VERSION = 1;
    public static final String DATABASE_NAME = "FeedReader.db";

    public FeedReaderDbHelper(Context context) {
        super(context, DATABASE_NAME, factory: null, DATABASE_VERSION);
    }
```

Рисунок 15 – Создание базы данных с использованием SQL помощника

```
public void makeRecord(Integer resumeNumber) {
    SQLiteDatabase database = mDbHelper.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(FeedReaderContract.FeedEntry.NUMBER_OF_RESUME, resumeNumber);
    values.put(FeedReaderContract.FeedEntry.DATE_OF_RESUME, dateFormat.format(Calendar.getInstance().getTime()));

    long newRowId = database.insert(
            FeedReaderContract.FeedEntry.TABLE_NAME,
            nullColumnHack: null,
            values);
}
```

Рисунок 16 – Реализация записи данных в базу данных

```
public void showRecords() {
    SQLiteDatabase database = mDbHelper.getReadableDatabase();
    String[] projection = {
            FeedReaderContract.FeedEntry._ID,
            FeedReaderContract.FeedEntry.NUMBER_OF_RESUME,
            FeedReaderContract.FeedEntry.DATE_OF_RESUME
    };
    String sortOrder = FeedReaderContract.FeedEntry.NUMBER_OF_RESUME + " DESC";
    Cursor cursor = database.query(
            FeedReaderContract.FeedEntry.TABLE_NAME,
            projection,
            selection: null,
            selectionArgs: null,
            groupBy: null,
            having: null,
            sortOrder
    );
    cursor.moveToFirst();
    StringBuilder stringBuilder = new StringBuilder();
    while(! cursor.isAfterLast()) {...}
    listItemsText.setText(stringBuilder.toString());
}
```

Рисунок 17 – Реализация чтения информации из базы данных

```
deleteButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (idOfDeletingItem.getText() != null) {
            Long rowId = Long.valueOf(String.valueOf(idOfDeletingItem.getText()));
            String selection = FeedReaderContract.FeedEntry._ID + " LIKE ?";
            String[] selectionArgs = { String.valueOf(rowId)};
            SQLiteDatabase database = mDbHelper.getWritableDatabase();
            database.delete( table: "resume_records", selection, selectionArgs);
            showRecords();
        }
    }
});
```

Рисунок 18 – Реализация удаления информации из базы данных

```
replaceButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (idOfReplacingItem.getText() != null) {
            SQLiteDatabase database = mDbHelper.getReadableDatabase();
            ContentValues values = new ContentValues();
            values.put(FeedReaderContract.FeedEntry.NUMBER_OF_RESUME, sharedPreferences.getInt( s: "resumes",  i: 0));
            values.put(FeedReaderContract.FeedEntry.DATE_OF_RESUME, dateFormat.format(Calendar.getInstance().getTime()));
            String selection = FeedReaderContract.FeedEntry._ID + " LIKE ?";
            Long rowId = Long.valueOf(String.valueOf(idOfReplacingItem.getText()));
            String[] selectionArgs = { String.valueOf(rowId) };
            int count = database.update(
                    FeedReaderContract.FeedEntry.TABLE_NAME,
                    values,
                    selection,
                    selectionArgs
            );
            showRecords();
        }
    }
});
```

Рисунок 19 – Реализация обновления базы данных

Вывод

Создано приложение с обеспечением хранения простых данных в парах ключ-значение, с сохранением произвольных файлов в файловой системе Android, с использованием базы данных SQLite.