



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИИТ)

Кафедра математического обеспечения и стандартизации ИТ

ОТЧЁТ ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №6

по дисциплине

«Разработка мобильных приложений»

Отчет представлен к

рассмотрению:

Студенты группы ИНБО-04-20

«28» февраля 2022 г.

(подпись)

Ло В.Х.

Преподаватель

« »

2022 г.

(подпись)

Фандеев И.И.

Москва, 2022г.

1. Реализовать получение дескриптора SharedPreferences

Чтобы получить дескриптор SharedPreferences, нам нужно создать фрагмент, а затем отредактировать его следующим образом.

```
Context context = getActivity();  
SharedPreferences sharedPref = context.getSharedPreferences(getString(R.string.preference_file_key), Context.MODE_PRIVATE);
```

2. Реализовать запись пар ключ-значение

Для записи пар ключ-значение Хранилище файлов нам нужно отредактировать в файле фрагмента следующим образом

```
SharedPreferences sharedPreferences = getActivity().getPreferences(Context.MODE_PRIVATE);  
int defaultValue = getResources().getInteger(R.string.saved_high_score_default);  
long highscore = sharedPreferences.getInt(getString(R.string.saved_high_score), defaultValue);
```

3. Реализовать получение прав для внешнего хранилищ

Чтобы получить разрешения для внешнего хранилища, нам нужно отредактировать файл манифеста следующим образом.

```
<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
    package="com.example.mysixthapplication"  
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />  
    <application  
        android:allowBackup="true"  
        android:icon="@mipmap/ic_launcher"  
        android:label="mysixthapplication"  
        android:roundIcon="@mipmap/ic_launcher_round"  
        android:supportRtl="true"  
        android:theme="@style/Theme.Mysixthapplication">  
        <activity  
            android:name=".MainActivity"  
            android:exported="true">  
            <intent-filter>  
                <action android:name="android.intent.action.MAIN" />  
  
                <category android:name="android.intent.category.LAUNCHER" />  
            </intent-filter>  
        </activity>  
    </application>  
</manifest>
```

4. Реализовать сохранение файлов во внутреннем хранилище

Чтобы сохранить файлы во внутренней памяти, нам нужно отредактировать файл фрагмента следующим образом.

```
File file = new File(context.getFilesDir(), filename);
String filename = "mylife";
String string = "hello world!";
FileOutputStream outputStream;

try {
    outputStream = openFileOutput(filename, Context.MODE_PRIVATE);
    outputStream.write(string.getBytes());
    outputStream.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

5. Реализовать сохранение файлов во внешнем хранилище

Чтобы сохранить файл на внешнее хранилище, нам нужно проверить доступность внешнего хранилища следующим образом.

```
public boolean isExternalStorageWritable() {
    String state = Environment.getExternalStorageState();
    if(Environment.MEDIA_MOUNTED.equals(state)){
        return true;
    }
    return false;
}

public boolean isExternalStorageReadable(){
    String state = Environment.getExternalStorageState();
    if(Environment.MEDIA_MOUNTED.equals(state) || Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)){
        return true;
    }
    return false;
}
```

это пример общедоступного фотоальбома

```

public File getAlbumStorageDir(String albumName){
    File file = new File(Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_PICTURES), albumName);
    if(!file.mkdirs()){
        Log.e(LOG_TAG, msg: "Directory not created");
    }
    return file;
}

```

это пример личного фотоальбома

```

public File getAlbumStorageDir(Context context,String albumName) {
    File file = new File(context.getExternalFilesDir(Environment.DIRECTORY_PICTURES), albumName);
    if (!file.mkdirs()) {
        Log.e(LOG_TAG, msg: "Directory not created");
    }
    return file;
}

```

6. Реализовать запрос свободного пространства

Для проверки свободного места делаем следующее

```

public long getFreeSpace() {
    File file;
    return file.getFreeSpace();
}

public long getTotalSpace (){
    File file;
    return file.getTotalSpace();
}

```

7. Реализовать удаление файла

Чтобы удалить файл, мы делаем следующее

```

File dir = getFilesDir();
File file = new File(dir, child: "my_filename");
boolean deleted = file.delete();

```

8. Создать схемы и контракт базы данных. Реализовать пример описания имени таблицы и столбцов

Чтобы создать схемы и контракт базы данных, мы делаем следующее

```

public final class FeedReaderContract{
    public FeedReaderContract(){
    }
    public static abstract class FeedEntry implements BaseColumns {
        public static final String TABLE_NAME = "entry";
        public static final String COLUMN_NAME_ENTRY_ID = "entryid";
        public static final String COLUMN_NAME_TITLE = "title";
        public static final String COLUMN_NAME_SUBTITLE = "subtitle";
    }
}

```

9. Создать базы данных с использованием SQL помощника

Чтобы создать базы данных с помощью SQLhelper, мы делаем следующее

```

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, "
    + FeedReaderContract.FeedEntry.COLUMN_NAME_ENTRY_ID + TEXT_TYPE + COMMA_SEP + FeedReaderContract.FeedEntry.COLUMN_NAME_TITLE
    + TEXT_TYPE + COMMA_SEP + " )";
private static final String SQL_DELETE_ENTRIES = "DROP TABLE IF EXISTS " + FeedReaderContract.FeedEntry.TABLE_NAME;

```

```

public class FeedReaderDbHelper extends SQLiteOpenHelper{
    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);
    }

    public void onCreate(SQLiteDatabase db){
        db.execSQL(SQL_DELETE_ENTRIES);
    }

    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion){
        db.execSQL(SQL_DELETE_ENTRIES);
        onCreate(db);
    }

    public void onDowngrade(SQLiteDatabase db, int oldVersion, int newVersion){
        onUpgrade(db, oldVersion,newVersion);
    }
}

FeedReaderDbHelper mDbHelper = new FeedReaderDbHelper((getContext()));

```

10. Реализовать запись данных в базу данных

Для записи данных в базу делаем следующее

```
FeedReaderDbHelper mDbHelper = new FeedReaderDbHelper((getContext()));

SQLiteDatabase db = mDbHelper.getReadableDatabase();
ContentValues values = new ContentValues();
values.put(FeedReaderContract.FeedEntry.COLUMN_NAME_ENTRY_ID, id);
values.put(FeedReaderContract.FeedEntry.COLUMN_NAME_TITLE, title);
values.put(FeedReaderContract.FeedEntry.COLUMN_NAME_CONTENT, content);

long newRowId;
newRowId = db.insert(FeedReaderContract.FeedEntry.TABLE_NAME, FeedReaderContract.FeedEntry.COLUMN_NAME_NULLABLE, values);
```

11. Реализовать чтение информации из базы данных

Чтобы прочесть информацию из базы данных, мы делаем следующее

```
FeedReaderDbHelper mDbHelper = new FeedReaderDbHelper((getContext()));
String[] projection = {
    FeedReaderContract.FeedEntry._ID, FeedReaderContract.FeedEntry.COLUMN_NAME_TITLE, FeedReaderContract.FeedEntry.COLUMN_NAME_UPDATED
};

String sortOrder = FeedReaderContract.FeedEntry.COLUMN_NAME_UPDATED + " DESC";

Cursor c = db.query(FeedReaderContract.FeedEntry.TABLE_NAME, projection, selection, selectionArgs, null, null, having: null, sortOrder);

c.moveToFirst();
long itemId = c.getLong(c.getColumnIndexOrThrow(FeedReaderContract.FeedEntry._ID));
```

12. Реализовать удаление информации из базы данных

Для удаления информации из базы делаем следующее

```
String selection = FeedReaderContract.FeedEntry.COLUMN_NAME_ENTRY_ID + " LIKE ?";
String[] selectionArgs = {String.valueOf(rowId)};
db.delete(table_name, selection, selectionArgs);
```

13. Реализовать обновление базы данных.

Для обновления базы делаем следующее

```
SQLiteDatabase db = mDbHelper.getReadableDatabase();
ContentValues values = new ContentValues();
values.put(FeedReaderContract.FeedEntry.COLUMN_NAME_TITLE, title);
String selection = FeedReaderContract.FeedEntry.COLUMN_NAME_ENTRY_ID + " LIKE ?";
String[] selectionArgs = {String.valueOf(rowId)};
int count = db.update(FeedReaderDbHelper.FeedEntry.TABLE_NAME, values, selection, selectionArgs);
```