

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«УФИМСКИЙ ГОСУДАРСТВЕННЫЙ НЕФТЯНОЙ  
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра вычислительной техники и инженерной кибернетики

## **РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ**

Учебно-методическое пособие  
для выполнения лабораторных работ

Уфа, 2019

В учебно-методическом пособии представлены варианты заданий лабораторных работ по дисциплине «Разработка мобильных приложений» и методические рекомендации по их выполнению.

Пособие предназначено для проведения лабораторных занятий со студентами направления подготовки 09.03.01 - «Информатика и вычислительная техника» по профилю «Программное обеспечение средств вычислительной техники и автоматизированных систем». Допускается использование данного учебно-методического пособия для проведения занятий по смежным дисциплинам или для смежных направлений и профилей обучения. Соответствует ФГОС 3++.

*Публикуется в авторской редакции.*

Составитель: Дружинская Е.В., старший преподаватель кафедры ВТИК

Рецензенты: Зарипов Д.М., доцент кафедры ВТИК,  
Имаева Л.Р., старший преподаватель кафедры ВТИК

## Содержание

Введение .....	4
1 Лабораторная работа №1. Жизненный цикл активности и фрагментов....	5
2 Лабораторная работа №2. Создание однооконного приложения.....	7
3 Лабораторная работа №3. Создание многооконного приложения.....	9
4 Лабораторная работа №4. Использование ресурсов устройства в мобильном приложении.....	10
5 Лабораторная работа №5. Потоки и графика.....	14
6 Лабораторная работа №6. Использование коллекций в приложении .....	15
7 Лабораторная работа №7. Использование библиотеки SQLite.....	17
8 Лабораторная работа №8. Разработка клиент-серверных приложений.....	18

**Введение.**

С развитием высоких технологий в нашу жизнь прочно вошли мобильные устройства, позволяющие «иметь в кармане» все необходимые коммуникации и информационную поддержку. И, конечно же, вместе с «умными» телефонами очень актуальной стала разработка приложений для них.

На сегодняшний день самая распространённая мобильная операционная система Android управляет наибольшим числом мобильных устройств: смартфонами, планшетами, очками, часами и даже автонавигацией. Многие лидеры рынка мобильной техники доверяют свои продукты именно этой системе.

Корпорация Samsung не только использует мобильную систему Android на своих устройствах, но и запустила в многих странах свою социальную образовательную программу по обучению школьников мобильной разработке под операционную систему Android. В России программа для школьников работает с 2014 года, а с 2019 года программа была интегрирована в студенческую IT академию Samsung в виде отдельного трека «Мобильная разработка».

Дисциплина «Разработка мобильных приложений» реализует программу IT академии Samsung. При изучении дисциплины используется смешанное обучение: электронный учебник и тестовый контроль по модулям от компании Samsung, лекционные, лабораторные занятия и выполнение курсового проекта проводится в вузе.

Дисциплина относится к вариативной части дисциплин учебного плана и читается в шестом семестре.

Данное учебно-методическое пособие содержит варианты заданий для проведения лабораторных занятий по дисциплине и методические рекомендации по их выполнению.

## Лабораторная работа 1 :Жизненный цикл активности и фрагментов.

### Задание 1. Отслеживание состояния активности

Создайте проект *FirstProject* и в файле кода MainActivity.java и измените метод onCreate().

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toast.makeText(this, "create",
Toast.LENGTH_SHORT).show();
}
```

Здесь super.onCreate() — это вызов конструктора класса-родителя Activity, а класс android.widget.Toast (класс всплывающих окон) — используется для вывода сообщения «create» на экран.

Переопределите остальные методы жизненного цикла Android-приложения. Необходимо отследить выполнение всех методов: onCreate(), onRestoreInstanceState(), onStart(), onResume(), onSaveInstanceState(), onPause(), onRestart(), onStop(), onDestroy().

Сохраните и запустите проект на планшете. Понаблюдайте за всплывающими окнами в процессе работы приложения.

Изучите другие объекты класса Toast и посмотрите их работу в вашем приложении.

### Задание 2.

Продублируйте сообщения, размещаемые в Toast в окне отладки LogCat (используйте класс Log):

```
Log.d("condition", "create");
```

При работе используйте один и тот же тег для всех состояний и выполните поиск по этому тегу в протоколе работы приложения.



***Задание 3.***

Добавьте в активность статический фрагмент и отслеживайте его жизненный цикл с помощью Toast и Log.

*Для выполнения работы используйте презентации:*

*3. Компоненты Android*

*5. Отладка приложений.*

## Лабораторная работа №2. Создание однооконного приложения.

### Задание 1.

1. Создайте приложение с пустой активностью EmptyActivity. В файле манифеста закрепите у **активности** горизонтальное расположение, добавив атрибут

```
android:screenOrientation="landscape"
```

2. Измените корневую разметку на LinearLayout в горизонтальной ориентации.

3. В корневой контейнер добавьте еще три LinearLayout, равного размера. Равномерность размеров дочерних разметок выполните расстановкой одинаковых весов элементов

```
android:layout_weight="1"
```

Измените в макете активности цвета добавленных контейнеров в цвета французского флага (синий, белый, красный).

4. В **белый** контейнер в центре поместите кнопку <Button> с надписью «НАЖМИ!», в **красный** контейнер разместите календарь, в **синий** – TableLayout из двух рядов: в первой строке поместите TextView в ширину родительского контейнера, во второй - две RadioButton («да» и «нет»).

5. Запрограммируйте кнопку в белом LinearLayout так, чтобы при нажатии на неё в TextView помещалась запись с включенной RadioButton. Обработчик нажатия кнопки выполните в виде метода класса MainActivity и поставьте в xml-описании кнопки значением свойства onClick:

```
android:onClick="mySetText"
```

6. Измените кнопку <Button> на кнопку с изображением <ImageButton> и замените текст подписи кнопки на изображение *ic\_launcher.png* из папки *miptar* ресурсов проекта.

### Задание 2.

Создайте приложение для решения неравенства или уравнения (по вариантам). Для ввода коэффициентов используйте текстовые поля ввода <EditText>, саму переменную размещайте в текстовой константе <TextView>.

Предусмотрите соблюдение ОДЗ. <sup>8</sup>Алгоритм решения помещается в обработчике кнопки <Button>, при нажатии на которую выводится множество корней уравнения или неравенства.

1.  $Ax^2 + Bx + C > 0$
2.  $A|x + B| > 0$
3.  $\text{Log}_A x + \text{Log}_A(x-1) < C$
4.  $A \cdot B^x + C < D$
5.  $(x + A)^B > C$
6.  $A|x| + B > 0$
7.  $Ax^2 + Bx + C < 0$
8.  $A|x| + B < 0$
9.  $A \cdot B^x + C > D$
10.  $\text{Log}_A x - \text{Log}_A(x-1) < C$

*Для выполнения работы используйте презентации*

*4. Управляющие элементы активности*

*6. Наследование и полиморфизм в Android-приложениях*



### **Лабораторная работа №3. Создание многооконного приложения.**

#### ***Задание 1.***

В приложение из задания 2 лабораторной работы №2 добавьте новую активность, назовите класс `StartActivity.java`. В файле манифеста переместите регистрацию этой активности в начало списка всех активностей. Подготовьте ещё 2 активности: `About` и `Solution`.

#### ***Задание 2.***

На активности `StartActivity` расположите кнопки «Математика» и «Выход» и `ImageView` со своей фотографией.

На активности `About` поместите свою фотографию, сведения о себе и кнопку «Назад».

На активности `Solution` поместите поля для ввода коэффициентов задачи, решаемой в `MainActivity` и кнопку «Передать».

#### ***Задание 3.***

Реализуйте переходы со стартовой активности:

- по кнопке «Математика» на активность `Solution`;
- по фотографии на активность `About`;
- по кнопке «Выход» завершение работы приложения.

#### ***Задание 4.***

Запрограммируйте кнопку «Передать» в активности `Solution` на запуск `MainActivity` с передачей ей введённых значений и отображении их в соответственных полях вызываемой активности.

#### ***Задание 5\*.***

Реализуйте возврат результатов вычисления из `MainActivity` в `Solution` с выводом их в соответственном поле.

*Для выполнения работы используйте презентацию 6. Наследование и полиморфизм в Android-приложениях.*

## **Лабораторная работа №4. Использование ресурсов устройства в мобильном приложении**

Создайте приложение из одной активности с тремя кнопками: «Задание1», «Задание 2» и «Задание 3».

Для выполнения задания 1 в файле манифеста в корневом элементе `<manifest>` добавьте разрешения на выход в интернет

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

### ***Задание 1. Интеграция браузера в приложение***

1. Создайте новую активность, корневую разметку экрана замените на WebView. Измените текст в TextView на надпись «Мой внутренний браузер», увеличьте размер шрифта, установите выравнивание текста по центру.

2. WebView задайте атрибут id, в коде объявите поле класса активности (WebView wv;) и в методе onCreate() найдите этот виджет по id.

3. Настройте WebView в качестве браузера для выполнения интернет-запросов

```
wv.getSettings().setJavaScriptEnabled(true);
```

4. Установите стартовую страницу браузера

```
wv.loadUrl("https://rusoil.net");
```

5. Для отображения результатов запросов в этом же браузере необходимо создать объект-клиента WebView. Для этого в классе активности создайте внутренний класс, наследник WebViewClient и переопределите в нём метод `shouldOverrideUrlLoading()` на отображение результатов в текущем окне

```
private class MyClient extends WebViewClient{
    @Override
    public boolean shouldOverrideUrlLoading(WebView view, WebResourceRequest request) {
        view.loadUrl(request.toString());
        return true;
    }
}
```

6. В методе onCreate() установите в качестве клиента для WebView объект описанного класса

```
wv.setWebViewClient(new MyClient());
```

7. Переопределите в активности <sup>11</sup>обработчик кнопки «назад» (метод `onKeyDown()`), чтобы он закрывал интернет-страницу, а не приложение.

```
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if ((keyCode == KeyEvent.KEYCODE_BACK) && wv.canGoBack()) {
        wv.goBack();
        return true;
    }
    return super.onKeyDown(keyCode, event);
}
```

8. В стартовой активности создайте намерение (Intent) на запуск этой активности по кнопке «Задание 1».

### *Задание 2. Использование камеры в приложении*

1. Создайте в приложении новую активность с кнопкой (Button) и картинкой (ImageView).

2. В классе активности создайте переменные кнопки и картинки, а так же константу для получения отклика неявного намерения

```
static final int REQUEST_IMAGE_CAPTURE = 1;
```

3. На кнопку установите обработчик нажатия, при котором будет вызвано неявное намерение к активному видеоустройству на телефоне (`MediaStore.ACTION_IMAGE_CAPTURE`)

```
Intent i = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
```

4. Если будет найдено такое устройство, то необходимо его активировать и получить с него снимок, иначе вывести сообщение о невозможности сделать фото.

```
if (i.resolveActivity(getPackageManager()) != null) {
    startActivityForResult(i, REQUEST_IMAGE_CAPTURE);
} else {
    Toast.makeText(getApplicationContext(), "Нечем снимать", Toast.LENGTH_LONG).show();
}
```

5. В методе `onActivityResult()` реализуйте получения отклика от камеры

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
        Bundle extras = data.getExtras();
        Bitmap imageBitmap = (Bitmap)extras.get("data");
        iv.setImageBitmap(imageBitmap);
    }
}
```

6. На кнопку «Задание 2» в <sup>12</sup>стартовой активности создайте намерение для запуска этой активности.

### ***Задание 3. Сохранение данных приложения в файле.***

1. Создайте активность с кнопкой. В классе активности объявите целочисленную переменную для подсчёта нажатий кнопки. На кнопку задайте обработчик, который будет считать нажатия и выводить значение в качестве подписи кнопки.

2. В классе активности объявите файловую переменную, в методе onCreate() выполните проверку на наличие файла на устройстве и создайте его при первом запуске приложения (используйте метод *createNewFile()*). При повторных запусках настройте Scanner для чтения целочисленного значения из файла в первоначальное значение счётчика нажатий.

3. Добавьте в активность кнопку «Сохранить счёт» и запрограммируйте её на перезапись результата в файл.

Для работы с файлами можно использовать наследников класса File: для записи FileWriter с методом write(); для чтения FileReader с методом read().

```
FileWriter fileWriter;  
int rezult;  
//...  
{  
    try {  
        fileWriter = new FileWriter( fileName: "in.txt");  
        fileWriter.write(rezult);  
        fileWriter.flush();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

Данные классы наследуются от классов входных / выходных потоков InputStreamReader / OutputStreamWriter:

```
try {  
    BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(  
        openFileOutput( name: "in.txt", MODE_PRIVATE)));  
    bw.write( str: rezult+"");  
    bw.close();  
} catch (FileNotFoundException e) {  
    e.printStackTrace();  
} catch (IOException e) {  
    e.printStackTrace();  
}
```

4. На кнопку «Задание 3» в <sup>13</sup>стартовой активности создайте намерение для запуска этой активности.

*Для выполнения работы используйте презентацию 1. Программирование Java.*

## Лабораторная работа №5. Потоки и графика

### Задание 1. Организация фонового потока

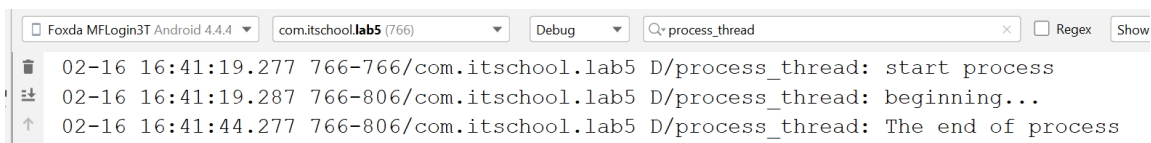
А) Создайте экран с двумя кнопками. Первую кнопку запрограммируйте на подсчёт количества нажатий на неё и вывод результата в Toast. На вторую кнопку запустите подсчёт суммы всех значений диапазона целых чисел `int`

```
long f = 0;
for(int i = Integer.MIN_VALUE; i < Integer.MAX_VALUE; f += i++);
```

По окончании вычисления выведите в текст кнопки надпись «ГОТОВО» и полученный результат.

Б) «Заверните» процесс вычисления в отдельный поток (используйте любой способ организации потока, во время защиты нужно будет заменить способ на другой). Убедитесь в невозможности передачи вычисленного значения в объекты главного потока.

Выполните логирование выполнения процесса в фоне, при этом проверяйте работоспособность приложения кнопкой «кликера».



В) Создайте вложенный класс - наследник `AsyncTask<Void,Void,Long>` и «перенесите» логирование процесса в основной поток, разместив комментарии в текстовое поле на экране или в Toast (переопределите методы `onPreExecute()`, `onPostExecute(Long l)`). Проверьте возможность передачи значения суммы из потока в UI, выведя вместе с текстом полученное значение суммы.

### Задание 2. Рисование на Canvas

Напишите класс, расширяющий `View`, напишите конструктор с настройками кисти и переопределите метод `onDraw()` так, чтобы на экране была нарисована открытка «С Днём защитника Отечества». На открытке должны быть использованы заливка, стандартные графические примитивы и на одном из объектов настроена любая анимация.

### Задание 3. Рисование на SurfaceView

На экране разместите кнопку и `SurfaceView`. Запрограммируйте кнопку на запуск анимации в `SurfaceView`: на поверхности «летают» разноцветные круги разных диаметров (цветные мыльные пузыри).

Для выполнения работы используйте презентации

7. Потоки

8. Графика.

## Лабораторная работа №6. Использование коллекций в приложении

### Задание 1. Использование *ListActivity*

Создайте активность, расширяющую *ListActivity*. В папке *res* проекта добавьте файл ресурсов с массивом `<string-array>`, запишите в него список городов страны по вариантам. В качестве источника данных для отображения в адаптере установите этот список, выберите шаблон отображения `android.R.layout.simple_list_item_1`. В строковых ресурсах добавьте для каждого города его описание и запрограммируйте обработчик выбора пункта на запуск новой активности с описанием города.

Варианты задания:

1. Франция
2. Испания
3. Португалия
4. Никарагуа
5. Монголия
6. Великобритания
7. Малайзия
8. Литва
9. Словакия
10. Польша
11. Эстония
12. Румыния
13. Молдова
14. Хорватия
15. Япония
16. Китай
17. Намибия
18. Мадагаскар
19. Белоруссия
20. Бангладеш
21. Алжир
22. Ангола
23. Греция
24. Киргизия
25. Мали
26. Непал
27. Южная Корея
28. КНДР

***Задание 2. Создание собственного шаблона для ListView***

В папке ресурсов res/layout создайте новый файл разметки, состоящий из текстового поля и картинки. Используйте данный шаблон для отображения списка стран с их флагами из задания 1.

***Задание 3. Использование RecyclerView.***

Выполните задание 2 с использованием RecyclerView.

***Задание 4. Использование Spinner***

Для каждого города подготовьте списки достопримечательностей в виде строковых массивов в ресурсах приложения. Создайте новую активность, разместите на неё Spinner и ListView. В Spinner выведите названия городов, в ListView - список достопримечательностей города, выбранного в Spinner.

*Для выполнения работы используйте презентацию 9. Списочные структуры данных.*



## **Лабораторная работа №7. Использование библиотеки SQLite**

### ***Задание.***

Выполните проектирование базы данных «Расписание занятий». Создайте приложение «Расписание занятий» с использованием локальной базы SQLite для работы с построенной базой данных. Выполните вывод расписания на день; поиск занятия по названию, по преподавателю, по аудитории. Предусмотрите функции добавления, изменения и удаления записи в базу данных.

*Для выполнения работы используйте презентацию 11. Библиотека SQLite*

## Лабораторная работа №8. Разработка клиент-серверных приложений

### ***Задание 1.***

1. В Eclipse создайте сервер с использованием библиотеки SpringBoot (STS). Запрограммируйте его на получение от клиента запроса со строковым аргументом `userName` - именем пользователя и отправкой ответа: «Welcome, `userName`», где в качестве `userName` подставляется полученное имя.

### ***Задание 2.***

Выполните реализацию клиента для построенного сервера с использованием библиотеки OkHttp3.

На активности расположите три элемента

*TextView* с надписью «Представьтесь, пожалуйста»

*EditText* для ввода имени

*Button* для отправки запроса.

После получения ответа в *TextView* разместите полученный ответ, *EditText* и *Button* сделайте невидимыми.

### ***Задание 3.***

Выполните реализацию нового клиента для построенного сервера с использованием библиотеки Retrofit.

### ***Задание 4\*.***

Выполнить регистрацию через ресурс Firebase (<https://firebase.google.com/>)

*Для выполнения работы используйте*

*\* презентацию 13. Клиент-серверные приложения,*

*\* материалы учебника (темы 5.3-5.4),*

*\* обучающее видео*

*<https://yandex.ru/video/preview?filmId=11821433904715268702&noreask=1&path=wizard&text=firebase>*