

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Емельянов Сергей Геннадьевич

Должность: ректор

Дата подписания: 15.05.2024 08:55:59

Уникальный программный ключ:

9ba7d3e34c012eba476ffd2d064cf2781957ba730df3374d16f3c0ce536f0fc6

МИНОБРАЗОВАНИЯ РОССИИ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«Юго-Западный государственный университет»

(ЮЗГУ)

Кафедра программной инженерии

УТВЕРЖДАЮ

Проректор по учебной работе

О.Г. Локтионова

« 25 » 10

2022 г.



АДМИНИСТРИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Методические указания для выполнения практических заданий для магистров направления подготовки 02.04.03 Математическое обеспечение и администрирование информационных систем

Курск 2022

УДК 004

Составитель А.И. Катыхин

Рецензент

Кандидат технических наук, Киселев А.В.

Администрирование информационных систем: методические указания для выполнения практических заданий для магистров направления подготовки 02.04.03 Математическое обеспечение и администрирование информационных систем/ Юго-Зап. гос. ун-т; сост.: А.И. Катыхин, Курск, 2022. 28 с.

Содержат методические указания для выполнения практических заданий.

Методические указания по структуре, содержанию и стилю изложения материала соответствуют методическим и научным требованиям, предъявляемым к учебным и методическим пособиям.

Предназначены для студентов направления подготовки 02.04.03 Математическое обеспечение и администрирование информационных систем.

Текст печатается в авторской редакции

Подписано в печать

Формат 60 x 84 1/16.

Усл. печ. л. Уч.- изд. л. Тираж 50 экз. Заказ 2143 Бесплатно.

Юго-Западный государственный университет.

305040, Курск, ул. 50 лет Октября, 94.

Содержание

Практическая работа №1. Взаимодействие с сервером	4
Практическая работа №2. Хранение данных. Настройки и внешние файлы.	9
Практическая работа №3. Уведомления	18
Список литературы	28

Практическая работа №1. Взаимодействие с сервером.

Цель работы:

Изучить работу с потоками. Научиться работать с мультимедиа файлами. Изучить работу с классом AsyncTask

Теоретические сведения

Главный и обычный потоки

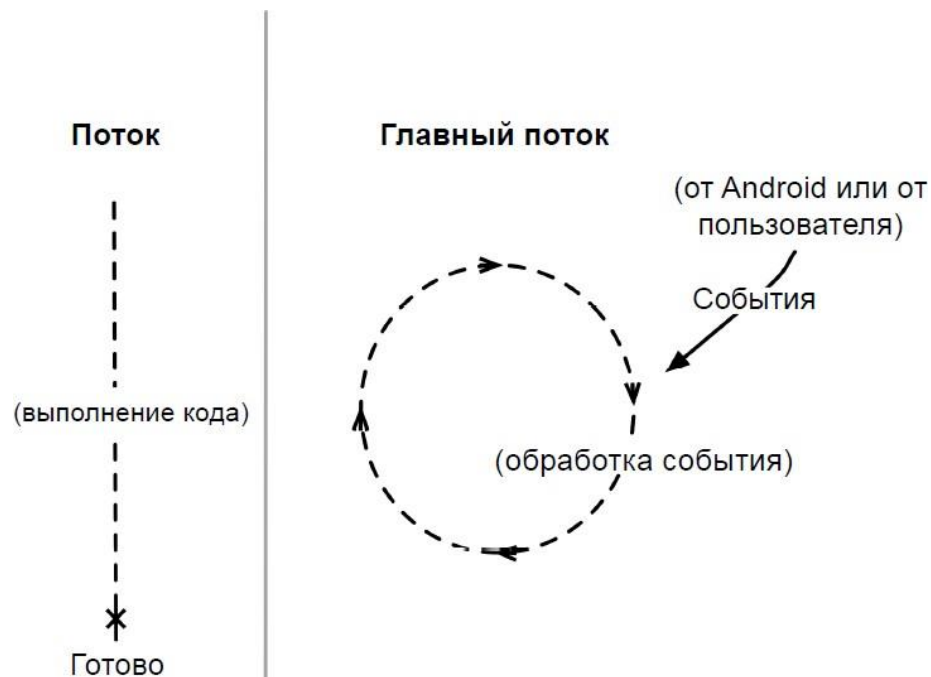


Рисунок 1.1 – Графическое представление потоков

Асинхронные потоки в Android

AsyncTask-это специальный абстрактный класс, предоставляющий набор методов для реализации:

- onPreExecute-для размещения инициализирующего кода (UI поток)
- doInBackground-для размещения тяжелого кода, который будет выполняться в другом потоке
- onProgressUpdate-для информирования о прогрессе (UI поток)
- onPostExecute-для обработки результата, возвращенного doInBackground(UI поток) и вспомогательных методов
- isCancelled-для получения информации об отмене задачи

- `publishProgress`-для перевода сообщения о прогрессе в UI поток с последующим вызовом `onProgressUpdate`

Последовательность выполнения методов `AsyncTask`

`onPreExecute`

`doInBackground`

`onPostExecute`

`publishProgress`

`onProgressUpdate`

Правила использования `AsyncTask`:

Объект `AsyncTask` должен быть создан в UI-поток

- Метод `execute` должен быть вызван в UI-поток
- Метод `execute` может быть запущен только один раз
- Не вызывайте методы `onPreExecute`, `doInBackground`, `onPostExecute` и `onProgressUpdate`

Передача данных в `AsyncTask`

Объявляем класс

```
Class MyAsyncTask extends AsyncTask<String, Integer, Long>{
```

```
...
```

```
}
```

- Первый параметр используется методом `doInBackground` `protected Long doInBackground(String... urls)`
- Второй параметр используется методом `onProgressUpdate` `protected void onProgressUpdate(Integer... progress)`
- Третий параметр используется методом `onPostExecute` `protected void onPostExecute(Long result)`

Промежуточные данные

Последовательность действий для передачи промежуточных данных в основной поток программы:

- В методе `doInBackground` вызываем метод `publishProgress`

- В методе onProgressUpdate обрабатываем переданный в publishProgress параметр и выводим прогресс

Метод get

- Возвращает результат выполнения метода doInBackground
- Вызывается из UI потока

```
MyAsyncTaskat = newMyAsyncTask();
```

```
...
```

```
result= at.get();
```

Задание на лабораторную работу Задание 1.

Рассмотрите пример передачи данных.

```
MyAsyncTaskat = newMyAsyncTask();
```

```
at.execute("url1", "url2");
```

```
doInBackground(String... urls)
```

Задание 2. Рассмотрите пример вывода промежуточных данных.

```
@Override
```

```
Protected void doInBackground(String... urls)
```

```
    { try{
```

```
        int cnt= 0;
```

```
        for(Stringurl: urls)
```

```
        {
```

```
            // обрабатываем первый параметр
```

```
            ...
```

```
            // выводим промежуточные результаты
```

```
            cnt++;
```

```
            publishProgress(cnt);
```

```
        }
```

```
        TimeUnit.SECONDS.sleep(1);
```

```
    } catch(InterruptedException)
```

```

        { e.printStackTrace();
        }
        return null;
    }
    @Override
    protected void onProgressUpdate(Integer... values) {
        super.onProgressUpdate(values);
        tv.setText("обработано " + values[0] + " параметров");
    }
}

```

Задание 3. Проверьте пример создания простой асинхронной задачи

```

public class MainActivity extends
Activity { MyAsyncTask at;
TextView tv;

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    tv= (TextView)
    findViewById(R.id.tv); MyAsyncTask
    at = new MyAsyncTask();at.execute();
}
}

```

Задание 4. На основании изученных примеров разработать приложение сохраняющее статистику проигрываемых песен на радио Мегабайт. Для сохранения песни и названия необходимо создать базу данных, содержащую таблицу со следующими полями:

- 1) ID
- 2) Исполнитель

- 3) Название трека
- 4) Время внесения записи

При включении приложения необходимо производить проверку подключения к Интернету. В случае если подключение отсутствует – выводить всплывающее сообщение (Toast) с предупреждением о запуске в автономном режиме (доступен только просмотр внесенных ранее записей).

После включения приложение должно производить асинхронный опрос сервера с интервалом 20 секунд. Если название трека не совпадает с последней записью в таблице необходимо произвести запись в БД.

URL адрес, по которому можно получить информацию о текущем треке и исполнителе:

http://media.ifmo.ru/api_get_current_song.php

Формат возвращаемых данных – JSON.

В случае успешного выполнения запроса результат будет иметь вид:

```
{“result”: “success”, “info” : “Исполнитель – Название трека” }
```

В случае ошибки API вернет следующую строку:

```
{“result”: “error”, “info” : “Информация об ошибке” }
```

Для успешного взаимодействия с API при обращении к странице необходимо передавать логин (login) и пароль(password) как POST-параметры.

login: 4707login

password: 4707pass

Приложение должно содержать активити, позволяющее просматривать внесенные в базу данных записи.

Практическая работа №2. Хранение данных. Настройки ивнешние файлы.

Цель работы

Изучить инструменты хранения данных, а также работу с внешними файлами.

Теоретические сведения

Варианты взаимодействия с сервером

1. Сокеты

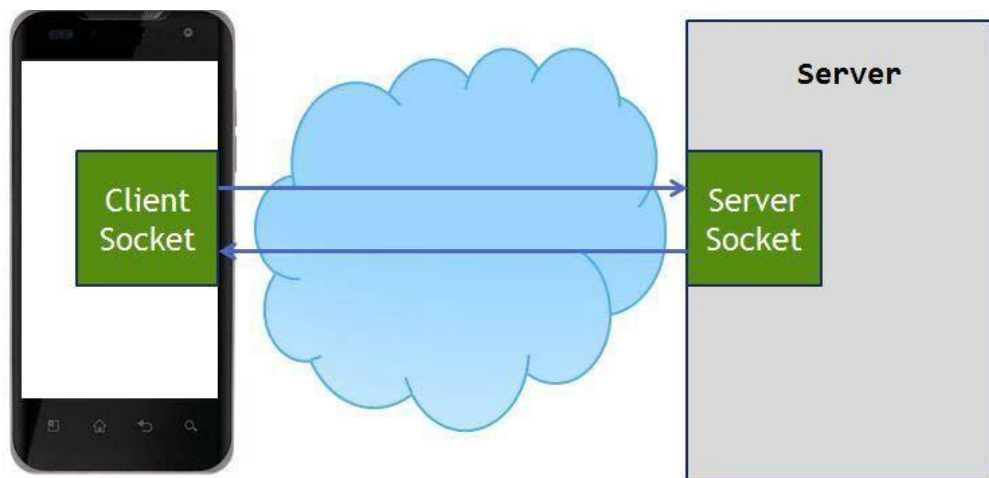


Рисунок 2.1 – Схема взаимодействия клиентского сокета с серверным

Используется в приложениях, где важна скорость доставки сообщения, важен порядок доставки сообщений и необходимо держать стабильное соединение с сервером. Такой способ зачастую реализуется в мессенджерах и играх.

2. Частые опросы

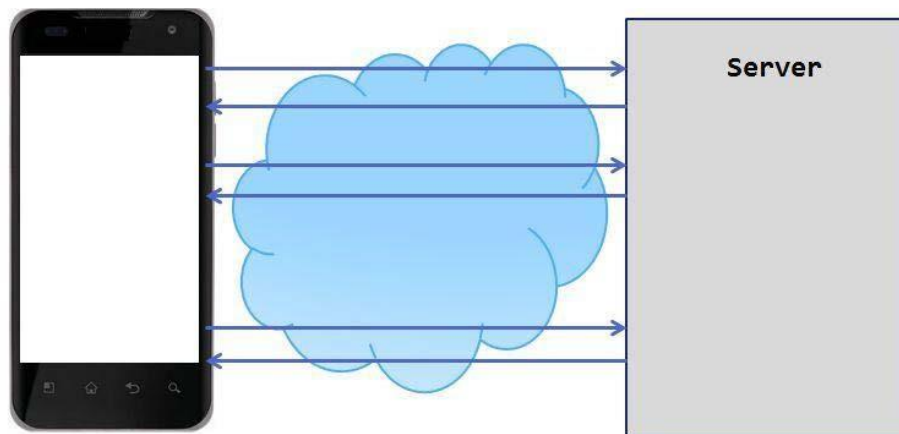


Рисунок 2.2 – Схема реализации частых опросов

Клиент посылает запрос на сервер и запрашивает новые данные. Сервер отвечает на запрос клиента и отдает все, что у него накопилось к этому моменту.

3. Длинные опросы

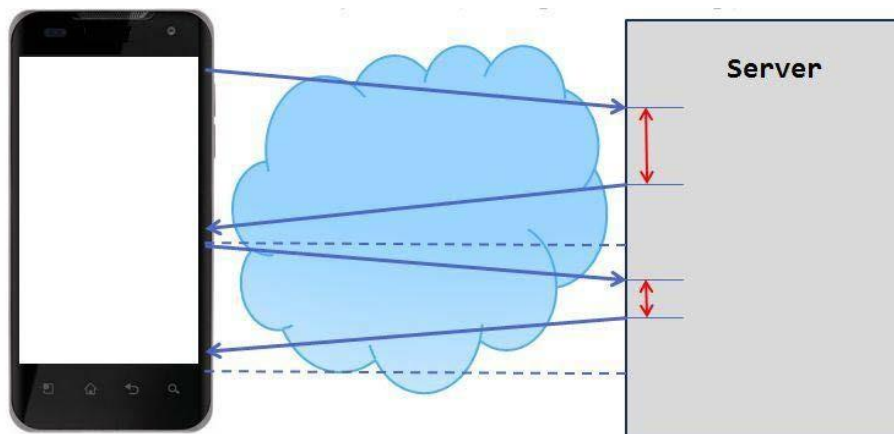


Рисунок 2.3 – Схема реализации длинных опросов

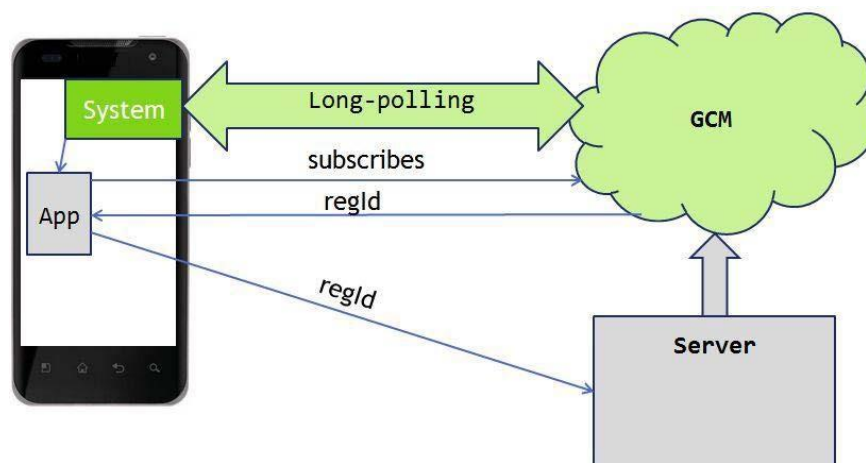


Рисунок 2.4 – Механизм реализации длинных опросов
Метод длинных опросов

- Реализация этого подхода достаточно сложна на мобильном клиенте в первую очередь из-за нестабильности мобильного соединения.
- При данном подходе расходуется меньше трафика, чем при обычном polling'e, т.к. сокращается количество установок соединений с сервером.
- Механизм long polling, или пуш-уведомлений (push notifications), реализован в самой платформе Android. (ваше приложение подписывается у сервиса Google Cloud Messaging (GCM) на получение пуш-уведомлений)

Таблица 2.1 – Библиотеки для работы с сетью

java.net	Содержит классы, связанные с сетевыми функциями, в том числе сокет, потоки и датаграммы, протокол IP, а также общие средства для работы с HTTP. Это многоцелевой ресурс для работы с сетями.
java.io	Пакет не относится непосредственно к сетям. Его классы используются сокетом и соединениями, содержащимися в других пакетах Java. Они используются также для обмена с локальными файлами (что часто происходит при взаимодействии с сетью).
java.nio	Содержит классы, которые служат буфером для определенных типов данных. Удобен для организации сетевой связи между двумя конечными точками средствами Java.
org.apache.*	Набор пакетов, которые обеспечивают точный контроль и функции для HTTP-коммуникаций на основе Apache - популярного веб-сервера с открытым исходным кодом.
android.net	Содержит дополнительные сокет доступа к сети в дополнение к основным классам java.net.*. Этот пакет включает в себя класс URI, который часто используется в разработке приложений Android, не связанных с сетью.
android.net.http	Содержит классы для работы с сертификатами SSL.
android.net.wifi	Содержит классы для реализации всех аспектов WiFi (802.11 Wireless Ethernet) на платформе Android.

Разрешения

- android.permission.INTERNET
- android.permission.ACCESS_NETWORK_STATE

Задание на лабораторную работу

Задание 1. Изучите пример подключения к

```

сети.public void myClickHandler(View view) {
...
ConnectivityManager connMgr = (ConnectivityManager)
    getSystemService(Context.CONNECTIVITY_SERVICE);
NetworkInfo networkInfo =
connMgr.getActiveNetworkInfo();if (networkInfo != null &&
networkInfo.isConnected()) {
    // fetch data
} else {
    // display error
}
...
}

```

Задание 2. Изучите код приложений

URLConnection

```

private String downloadUrl(String myurl) throws IOException
{
    InputStream is = null;
    // Only display the first 500 characters of the retrieved
    // web page content.
    int len = 500;
    try {
        URL url = new URL(myurl);
        HttpURLConnection conn =
            (HttpURLConnection)
url.openConnection();
        conn.setReadTimeout(10000 /* milliseconds */);
        conn.setConnectTimeout(15000 /* milliseconds */);
        conn.setRequestMethod("GET"); conn.setDoInput(true);
        // Starts the query conn.connect();
        int response = conn.getResponseCode();

```

```

        Log.d(DEBUG_TAG, "The response is: " + response); is
        =conn.getInputStream();
        // Convert the InputStream into a string
        String contentAsString = readIt(is, len);
        return contentAsString;
    // Makes sure that the InputStream is closed after the app is
    // finished using it.
    } finally {
        if (is != null) {
            is.close();
        }
    }
}

```

Преобразование полученной информации к типу String

```

public String readIt (InputStream stream, int len) throws IOException,
UnsupportedEncodingException {
    Reader reader = null;
    reader = new InputStreamReader(stream, "UTF-8");
    char[] buffer = new char[len];
    reader.read(buffer); return
    new String(buffer);
}

```

Http GET запрос

- Создаем HttpClient


```
HttpClient client = new DefaultHttpClient();
```
- Создаем объект HttpGet


```
HttpGet request = new HttpGet("http://www.example.com");
```
- Выполняем HTTP запрос

```

    HttpResponse response;
try {
    response = client.execute(request);
    Log.d("Response of GET request", response.toString());
} catch (ClientProtocolException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (IOException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

```

Взаимодействие с сервером через сокеты

```

public class Requester extends Thread {
    Socket requestSocket;
    String message;
    StringBuilder returnStringBuffer = new
    StringBuilder();
    Message lmsg;
    int ch;
    @Override public void run()
    {
        try {
            this.requestSocket = new Socket("remote.servername.com",
            13);
            InputStreamReader isr = new
            InputStreamReader(this.requestSocket.getInputStream(), "ISO-8859-
            1");
            while ((this.ch = isr.read()) != -1) {
                this.returnStringBuffer.append((char) this.ch);
            }
            this.message = this.returnStringBuffer.toString();
            this.lmsg = new Message();

```

```

        this.lmsg.obj = this.message;
        this.lmsg.what = 0;
        h.sendMessage(this.lmsg);
        this.requestSocket.close();
    }
    catch (Exception ee) {
        Log.d("sample application", "failed to read data"
            +ee.getMessage());
    }
}
}
}

```

Задание 3. Работа с внешними файлами.

Разработать мобильное приложение, позволяющее пользователю асинхронно скачивать файлы журнала Научно-технический вестник. Файлы хранятся на сервере в формате PDF и расположены по адресу:

http://ntv.ifmo.ru/file/journal/идентификатор_журнала.pdf

Не для всех ID имеются журналы, поэтому необходимо предусмотреть сообщение об отсутствии файла. В случае если файл не найден, ответ от сервера будет содержать главную страницу сайта.

Определить существует ли файл можно по возвращаемому сервером заголовку (параметр content-type).

Примеры ссылок:

<http://ntv.ifmo.ru/file/journal/1.pdf> – возвращен PDF файл

<http://ntv.ifmo.ru/file/journal/2.pdf> – файл не найден, возвращена главная страница сайта

Файлы должны храниться на устройстве в папке, создаваемой при первом запуске приложения (путь до папки и ее название определите самостоятельно).

После окончания загрузки файла должна становиться доступной кнопка «Смотреть» и кнопка «Удалить».

При нажатии на кнопку «Смотреть» должно происходить открытие сохраненного на устройстве файла. Предусмотреть ошибку, если на устройстве не установлено приложение, открывающее PDF файлы.

При нажатии на кнопку «Удалить» загруженный файл должен удаляться с устройства.



Рисунок 2.5 – Интерфейс приложения

Задание 4. Хранение и чтение настроек.

При запуске приложения пользователю должно выводиться всплывающее полупрозрачное уведомление (popupWindow), с краткой инструкцией по использованию приложения (можете написать случайный текст), чекбоксом «Больше не показывать» и кнопкой «ОК».

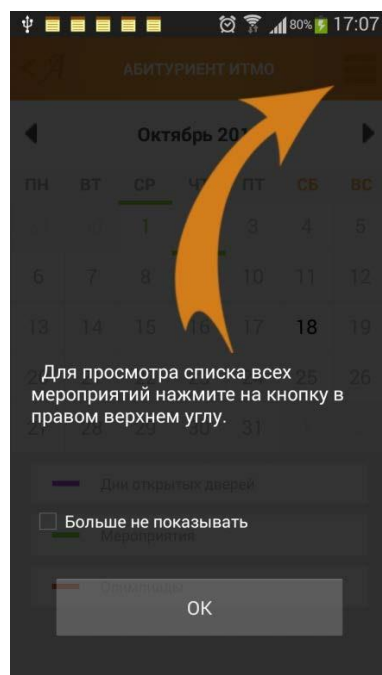


Рисунок 2.6. – Результат работы приложения

Если чекбокс был отмечен и нажата кнопка ОК, необходимо произвести сохранение данного параметра используя SharedPreferences. При следующем запуске приложения производить проверку параметра, и не выводить всплывающее сообщение, если чекбокс был отмечен.

Лабораторная работа №3. Уведомления.

Цель работы

Изучить инструменты хранения данных, а также работу с внешними файлами.

Теоретические сведения

Варианты хранения данных на устройстве

- Preferences (аналог INI-файлам в ОС Windows)
- SQLite - база данных, таблицы
- обычные файлы - внутренние и внешние (на SD карте)

Preferences

Значения сохраняются в виде пары: имя, значение

- Для хранения используется файл формата XML
- Путь к сохраняемым настройкам
/data/data/YOUR_PACKAGE_NAME/shared_prefs/YOUR_PREFS_NAME.xml

Отличие методов `getPreferences` и `getSharedPreferences`

```
public SharedPreferences getPreferences(int mode) {  
    return getSharedPreferences(getLocalClassName(), mode);  
}
```

Метод `getSharedPreferences`

```
public abstract SharedPreferences getSharedPreferences (String name, int mode)
```

- Name – имя файла с настройками
- Mode – режим

MODE_PRIVATE – режим по умолчанию. Файл доступен из только из текущего приложения.

MODE_WORLD_READABLE (deprecated in API level 17) – разрешено чтение всеми приложениями.

MODE_WORLD_WRITEABLE - (deprecated in API level 17) –

разрешена запись всеми приложениями.

MODE_MULTI_PROCESS – режим для взаимодействия с файлом из нескольких процессов

Создание экрана настроек

pref.xml

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen
xmlns:android="http://schemas.android.com/apk/res/android">
    <CheckBoxPreference
        android:key="notif"
        android:summary="Enable notifications"
        android:title="Notifications">
    </CheckBoxPreference>
    <EditTextPreference
        android:key="address"
        android:summary="Address for notifications"
        android:title="Address">
    </EditTextPreference>
</PreferenceScreen>
```

Создание экрана настроек

PrefActivity.java

```
public class PrefActivity extends PreferenceActivity
{
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        addPreferencesFromResource(R.xml.pref);
    }
}
```

Создание экрана настроек

main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android=http://schemas.android.com/apk/res/android
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <TextView
        android:id="@+id/tvInfo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="">
    </TextView>
</LinearLayout>

```

Создание экрана настроек

MainActivity.java

```

public class MainActivity extends Activity
{
    TextView tvInfo;
    SharedPreferences sp;
    /** Called when the activity is first created. */
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        tvInfo = (TextView) findViewById(R.id.tvInfo);
        sp = PreferenceManager.getDefaultSharedPreferences(this);

        // полная очистка настроек
        // sp.edit().clear().commit();
    }
    protected void onResume() {
        Boolean notif = sp.getBoolean("notif", false);

```

```

String address = sp.getString("address", "");
String text = "Notifications are " + ((notif) ? "enabled, address = " +
address : "disabled");
tvInfo.setText(text);
super.onResume();
}
public boolean onCreateOptionsMenu(Menu menu) {
    MenuItem mi = menu.add(0, 1, 0, "Preferences");
    mi.setIntent(new Intent(this, PrefActivity.class));
    return super.onCreateOptionsMenu(menu);
}
}

```

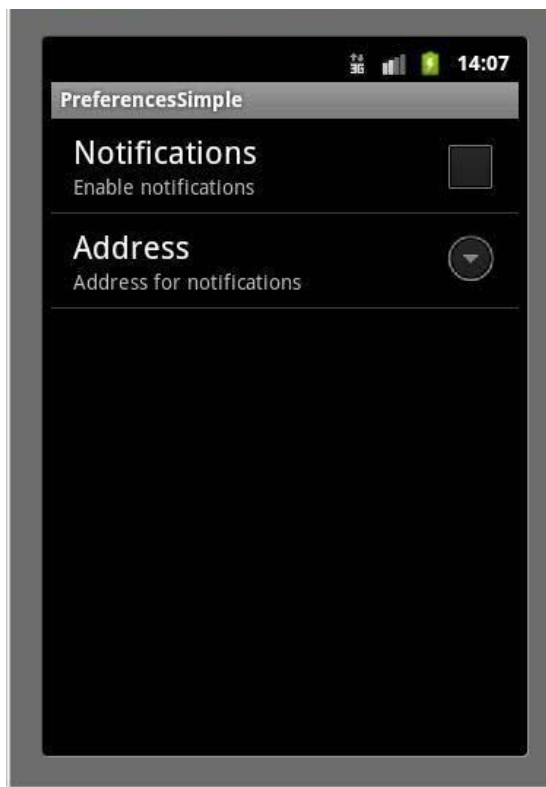


Рисунок 3.1 – Интерфейс приложения

Работа с файлами

Основные используемые классы

- `BufferedReader`, `BufferedWriter`
- `InputStreamReader`, `OutputStreamWriter`

- openFileInput, openFileOutput\
- FileReader, FileWriter
- File

Пример

```
public class MainActivity extends Activity
    { final String LOG_TAG =
      "myLogs"; final String FILENAME
      = "file";
      final String DIR_SD = "MyFiles";
      final String FILENAME_SD = "fileSD";
      /** Called when the activity is first created. */
      @Override
      public void onCreate(Bundle savedInstanceState) {
          super.onCreate(savedInstanceState);
          setContentView(R.layout.main);
      }
      public void onclick(View v)
          { switch (v.getId()) {
            case R.id.btnWrite:
                writeFile()
                ;break;
            case R.id.btnRead:
                readFile()
                ;break;
            case
              R.id.btnWriteS
              D:
                writeFileSD();
                break;
```

```

        case
            R.id.btnReadS
            D:
                readFileSD();
                break;
    }
}

```

Запись файла в память устройства

```

void writeFile()
{ try {
    // отрываем поток для записи
    BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(
        openFileOutput(FILENAME, MODE_PRIVATE)));
    // пишем данные
    bw.write("Содержимое файла");
    // закрываем поток
    bw.close();
    Log.d(LOG_TAG, "Файл записан");
} catch (FileNotFoundException e)
{ e.printStackTrace(); }
    catch (IOException e) {
        e.printStackTrace();
    }
}

```

Чтение файла из памяти устройства

```

void readFile()
{ try {
    // открываем поток для чтения BufferedReader br
    = new BufferedReader(new InputStreamReader(
        openFileInput(FILENAME)));
}
}

```

```

String str = "";
// читаем содержимое
while ((str = br.readLine()) != null)
    {Log.d(LOG_TAG, str);
}
} catch (FileNotFoundException e)
    {e.printStackTrace();
}
} catch (IOException e) {
    e.printStackTrace();
}
}

```

Запись на SD карту

```

void writeFileSD() {
    // проверяем доступность SD
    if (!Environment.getExternalStorageState().equals(
        Environment.MEDIA_MOUNTED)) {
        Log.d(LOG_TAG, "SD-карта не доступна: "
            +Environment.getExternalStorageState());
        return;
    }

    // получаем путь к SD
    File sdPath = Environment.getExternalStorageDirectory();
    // добавляем свой каталог к пути
    sdPath = new File(sdPath.getAbsolutePath() + "/" + DIR_SD);
    // создаем каталог sdPath.mkdirs();
    // формируем объект File, который содержит путь к файлу
    File sdFile = new File(sdPath,
        FILENAME_SD);try {

```



```

        // открываем поток для записи
        BufferedWriter bw = new BufferedWriter(new FileWriter(sdFile));
        // пишем данные
        bw.write("Содержимое файла на SD");
        // закрываем поток
        bw.close();
        Log.d(LOG_TAG, "Файл записан на SD: " +
            sdFile.getAbsolutePath());
    }

    catch (IOException e)
    {
        e.printStackTrace();
    }
}

```

Чтение с SD карты

```

void readFileSD() {
    // проверяем доступность SD
    if (!Environment.getExternalStorageState().equals(
        Environment.MEDIA_MOUNTED)) {
        Log.d(LOG_TAG, "SD-карта не доступна: "
            +Environment.getExternalStorageState());
        return;
    }

    // получаем путь к SD
    File sdPath = Environment.getExternalStorageDirectory();
    // добавляем свой каталог к пути
    sdPath = new File(sdPath.getAbsolutePath() + "/" + DIR_SD);
    // формируем объект File, который содержит путь к файлу
    File sdFile = new File(sdPath, FILENAME_SD); try {
        // открываем поток для чтения
        BufferedReader br = new BufferedReader(new FileReader(sdFile));

```

```
String str = "";
// читаем содержимое
while ((str = br.readLine()) != null)
{Log.d(LOG_TAG, str);
}
} catch (FileNotFoundException e)
{e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}
```

Разрешение на работу с файлами на SD карте

- android.permission.READ_EXTERNAL_STORAGE
- android.permission.WRITE_EXTERNAL_STORAGE

Задание на лабораторную работу

Задание 1. Разработать мобильное приложение, позволяющее устанавливать напоминания.

Требования к приложению

- 1) Создание напоминаний и сохранение их в базу данных. В БД должны храниться следующие значения (Заголовок, Текст уведомления, дата уведомления)
- 2) Просмотр установленных уведомлений
- 3) Удаление уведомлений
- 4) Дата напоминания должна устанавливаться с помощью TimePickerDialog и DatePickerDialog
- 5) Стилизовать напоминание в Notification Center и Status bar (установить собственный лого)
- 6) При нажатии на уведомление в Notification Center переходить в активити приложения с полным текстом уведомления.

Классы для создания приложения: Notification, NotificationManager, PendingIntent, BroadcastReceiver, AlarmManager

Список литературы

1. Куркин, А.В. Программирование под платформу Android [Текст]: учебное пособие / Куркин А.В. – СПб.: Университет ИТМО, 2015. - 35 с.
2. Григорьева, И.И. Разработка мобильных приложений [Текст]: учебное пособие / И.И. Григорьева, С.С. Самборецкий. – Тюмень, 2015. - 125 с.
3. Дэрсси, Л. Android за 24 часа. Программирование приложений под операционную систему Google [Текст] / Л.Дэрсси, Ш.Кондер. - ; М.: Рид Групп, 2011. - 464 с.
4. Колисниченко, Д. Программирование для Android [Текст]: самоучитель / Д. Колисниченко. – М.: БХВ-Петербург - Москва, 2012. - 272 с.
5. Майер, Р. Android 4. Программирование приложений для планшетных компьютеров и смартфонов [Текст] / Р. Майер. – М.: Эксмо, 2013. - 816 с.
6. Медникс, З. Программирование под Android [Текст] / З.Медникс, Л.Дорнин, Б.Мик, М.Накамура. – СПб.: Питер, 2013. – 560 с.
7. Ретабоуил, С. Android NDK. Разработка приложений под Android на C/C++ [Текст] / С. Ретабоуил. – М.: ДМК Пресс, 2012. - 496 с.
8. Цехнер, М. Программирование игр под Android [Текст] / М. Цехнер. – СПб.: Питер, 2013. – 688 с.

Удалённое управление системой и резервное копирование

Цель работы: получить навыки удалённого управления системой и резервного копирования.

Используемое программное обеспечение

Для выполнения данной лабораторной работы Вам понадобится компьютер с операционной системой Microsoft Windows не старше XP и установленным программным обеспечением Microsoft Virtual PC 2007 SP1 (с ОС Windows 7 и Windows Virtual PC) и 2 подготовленные виртуальные машины с Windows Server 2008 и Windows Vista.

Remote Server Administration Tools

Для удаленного администрирования предыдущих серверных операционных систем семейства Windows (т.е. Windows 2000 Server и Windows Server 2003) на рабочую станцию администратора нужно было установить пакет инструментов из файла Adminpak.msi с дистрибутива соответствующего сервера. С выходом Windows Server 2008 все несколько изменилось и теперь для этого используется пакет RSAT, который можно свободно загрузить с сервера Microsoft. Он существует в версиях для Windows Vista SP1 и Windows 7.

Задание. Из файла, прикладываемого к описанию работы, установите на рабочую станцию с Windows 7 утилиты удаленного администрирования (для этого вам понадобится зайти под учетной записью с административными привилегиями).

Прим. Если на виртуальной машине установлена ОС Windows 7 без SP1, то предварительно нужно поставить этот пакет обновлений. При этом надо читать, что для его установки потребуется не менее 6Гб свободного пространства на диске, где установлена ОС.

После установки, средства администрирования из состава RSAT необходимо активировать из "Панели управления". Активированные приложения появятся в папке «Администрирование» на локальном компьютере.

Служба удаленного терминала

Также удаленное управление сервером можно осуществлять через удаленный терминал. В этом случае, управляющие утилиты (и другие приложения, запускаемые из терминала), выполняются на сервере, а на клиентский компьютер передается только картинка рабочего стола пользователя. Служба терминалов использует протокол RDP (Remote Desktop Protocol), кроме того, в Windows Server 2008 появилась возможность использования удаленного терминала через web-интерфейс.

Начнем с того, что в Server Manager добавляем серверу роль Terminal Services. Для выполнения первой части лабораторной работы нам понадобится только служба Terminal Server. Если ее отметить, то появится предупреждение о том, что установка служб удаленного терминала на контроллер домена нежелательна по соображениям безопасности: удаленные пользователи получат доступ к консоли контроллера домена. Но так как наша виртуальная сеть включает только один сервер, это предупреждение сейчас игнорируем.

Следующее окно мастера установки роли позволяет указать способ выполнения аутентификации при подключении. Более безопасный тип – Network Level Authentication, но он поддерживается не во всех случаях. Выберем его, т.к. у нас новая версия клиентской ОС (Windows Vista) и подключаемся мы фактически из локальной сети, поэтому проблем с совместимостью быть не должно.

После этого будет запрошен тип лицензирования устанавливаемого сервера терминалов. В нашем случае отметим первый вариант и получим пробную 120-дневную лицензию на использование сервера терминалов (для целей обучения этого будет вполне достаточно).

Последнее окно мастера позволяет указать пользователей и группы, которым разрешен терминальный доступ. Здесь для выполнения лабораторной работы также достаточно оставить предлагаемый вариант – только группу Администраторы. После этого начнется установка выбранной службы. Для ее окончания нужно будет перезагрузить виртуальную машину, после чего снова войти под администратором и закончить процедуру установки.

Теперь, чтобы работать на сервере и на рабочей станции под разными учетными записями, создайте нового доменного пользователя adms с паролем 123qweQWE и включите его в группу Администраторы (делается это с помощью оснастки Active Directory Пользователи и компьютеры).

С рабочей станции подключитесь к серверу терминалов (Пуск - Все программы - Стандартные - Подключение к удаленному рабочему столу). Для этого в окне терминального клиента сначала кажите имя или ip-адрес сервера, а потом учетную запись, от имени которой будет производиться подключение. Обратите внимание, что по умолчанию считается, что вводимая учетная запись – локальная. Поэтому при использовании доменных записей надо полностью вводить имя домена и имя пользователя, например, Test\adms.

Административная оснастка Служба удалённого терминала (Пуск – Средства администрирования – Службы терминалов) позволяет отслеживать текущие сессии, пользователей, работающих в данных сессиях и запускаемые ими процессы (вкладка Процессы). Пользователю можно отправить сообщение, при наличии соответствующих прав завершить выполнение отдельного процесса или сессии пользователя вообще, выполнить другие действия.

В терминальной сессии доступен буфер обмена, через который можно передавать данные на удаленный (локальный) компьютер. Аналогичным образом можно копировать и файлы. Зачастую администратору надо сделать так, чтобы пользователь, не выбирая компьютер и параметры подключения, мог быстро подключиться к нужному серверу и получить нужное ему приложение. Такую настройку можно сделать с помощью оснастки TS RemoteApp Manager. Выбрав ссылку Добавить программы RemoteApp, можно добавить приложение, «публикуемое» на терминальном сервере. Когда приложение добавлено, можно создать для него .rdp файл (ссылка Создать .rdp file). Теперь клиент, получивший этот файл, может его запустить, ввести имя пользователя и пароль (если это предполагают настройки) и получить доступ к нужному приложению, как будто оно запущено локально.

Задание. «Опубликуйте» приложение. Проверьте работу сделанных настроек. Что будет отображаться на сервере в оснастке Службы удалённых терминалов, когда пользователь запустил удаленное приложение? А когда закрыл его?

Посмотрите описания опубликованного приложения в TS RemoteApp Manager и свойства .rdp файла.

Как Вы убедились в ходе выполнения задания, если пользователь закрывает окно опубликованного приложения или окно терминального клиента, его сессия не заканчивается. Таким образом, удаленный пользователь может запустить какую-то программу на выполнение, а спустя некоторое время (часы, дни ...) – подключиться, чтобы проверить результаты. В одних случаях, это может быть полезно. Но в других – может потребоваться автоматически закрывать неактивные в течение определенного времени сессии (для экономии ресурсов сервера, из-за ограниченного числа терминальных лицензий). Такую настройку можно сделать в оснастке Active Directory Пользователи и компьютеры в свойствах учетной записи пользователя.

Доступ к службе удаленного терминала через web-интерфейс

Теперь рассмотрим, как настроить работу с удаленным терминалом через web-интерфейс. Это может потребоваться в случаях, когда пользователи по каким-то причинам не могут использовать обычный терминальный клиент (фильтрация трафика провайдером и т.д.). Для начала, нужно установить дополнительную службу. В оснастке Server Manager запустите добавление служб для роли Terminal Services. В окне отметьте службу TS Web Access. Появится запрос на добавление связанных компонент – модулей IIS, обеспечивающих работу с динамически формируемым содержимым (ASP.Net, ISAPI и т.д.).

После успешной установки, в оснастке Internet Information Services (IIS) Manager проверьте, что у web-сайта Default Web Site появился раздел TS.

На рабочей станции запустите Internet Explorer и в строке адреса наберите `http://<имя сервера>/ts` В открывшемся окне перейдите на вкладку Remote Desktop (рис.9.10). При

необходимости добавьте сайт в список доверенных, чтобы с него можно было запускать активное содержимое. Для этого в меню **Сервис** выберите **Свойства обозревателя**, на вкладке **Безопасность** откройте список надежных узлов и добавьте в него наш узел. После этого попробуйте подключиться к серверу. В остальном работа через web-интерфейс ничем особо не отличается от работы с использованием терминального клиента.

Регулярное резервное копирование информации с серверов и локальных жестких дисков предотвращает утрату и повреждение данных из-за поломки жесткого диска, отключения питания, воздействия вирусов и т.д. Резервное копирование при грамотном планировании и наличии надежного оборудования позволяет безболезненно справиться с последствиями катастрофы.

Графическое инструментальное средство Windows 2003 Backup предназначено для автоматического и ручного резервного копирования и восстановления файлов, расположенных на разделах файловых систем FAT и NTFS.

Перед тем как приступить к резервному копированию файлов, нужно разработать стратегию, отвечающую потребностям Вашей организации и гарантирующую восстановление утраченных данных. Эффективное архивирование и восстановление информации — одна из самых важных задач администратора.

По степени важности (следовательно, и по частоте создания резервных копий) папки и файлы можно разделить на три категории:

- важные – их резервные копии создаются всегда;
- полезные – их резервные копии создаются изредка;
- малозначимые – их резервные копии не создаются никогда.

Отбирая файлы для резервного копирования, учитывайте следующие правила:

- всегда создавайте резервные копии файлов, жизненно важных для работы Вашей организации; реестров всех главных и резервных контроллеров домена (каждый контроллер домена имеет свою копию базы данных каталогов; резервное копирование реестра контроллера домена предотвращает потерю информации об учетных записях пользователей и защите).

- резервные копии файлов, изменяемых редко или не представляющих особой ценности, следует создавать лишь время от времени.

- не сохраняйте временные файлы, так как они постоянно изменяются, и вряд ли могут быть использованы для восстановления данных.

Выбор типа резервного копирования

Программа архивации предлагает пять вариантов резервного копирования: *обычное, копирующее, инкрементальное, разностное и ежедневное*. Выбор стратегии резервного копирования определяется тем, сколько времени отводится на сохранение данных и каковы требования к скоростям поиска резервных копий и восстановления файлов.

Таблица 1 – Варианты резервного копирования

Варианты резервного копирования	Характеристика
Обычное или полное	Архивирует выбранные файлы и помечает их как сохраненные. Обычное резервное копирование позволяет быстро восстанавливать файлы, так как наиболее свежие файлы находятся на последней ленте. Для создания первой резервной копии всегда следует применять обычное резервное копирование всех файлов
Инкрементальное или добавочное	Архивирует только файлы, созданные или измененные с момента выполнения последнего обычного или инкрементального резервного копирования. Эти файлы помечаются флажком архивации. Если Вы сочетаете обычное и инкрементальное резервное копирование, то воссоздание информации начинается с восстановления последней обычной резервной копии, а затем последовательно восстанавливаются файлы инкрементальных копий

Разностное	Архивирует файлы, созданные или измененные со времени последнего обычного (или инкрементального) резервного копирования. Файлы при этом не помечаются флажком архивации. При комбинации обычного и разностного резервного копирования для восстановления данных требуются лишь 2 ленты: с последней обычной и с последней разностной копиями
Копирующее	Архивирует выбранные файлы, не помечая их флажком архивации. Тем самым не оказывает влияния на операции обычного и инкрементального резервного копирования и может применяться для промежуточного сохранения данных
Ежедневное копирование	Архивирует выбранные, файлы, которые были изменены во время ежедневного копирования. Файлы не помечаются флажком архивации. Эта операция полезна, например, когда Вы берете работу на дом и хотите быстро выбрать файлы, над которыми сегодня работали

Журнал резервного копирования (backup log) — это текстовый файл, в котором регистрируются операции резервного копирования. Он полезен при восстановлении данных. Его можно либо распечатать, либо посмотреть в любом текстовом редакторе. Журнал хранится на диске, поэтому в случае повреждения каталога архива на ленте обратитесь к нему, чтобы найти нужный файл.

Программа резервного копирования запускается в меню Пуск - Программы - Стандартные - Служебные - Архивация данных. Программа содержит 3 мастера: архивации, восстановления и аварийного восстановления. Во вкладке «Архивация» можно выбрать параметры архивации: объекты архивации и назначение архивации. Можно также выбрать дополнительные параметры архивации. Во вкладке «Восстановление управления носителем» можно выбрать необходимые для восстановления объекты. Во

вкладке «Запланированные задания» можно создать задание архивации, путем вызова мастера архивации.

Создайте план резервного копирования и выполните резервное копирование различного типа с помощью программы Backup Utility (Пуск - Программы - Стандартные - Служебные)

- Backup (Архивация данных):
- Полное
- Инкрементное
- Ежедневное

Контрольные вопросы

1. Назовите виды резервного копирования.
2. Чем отличается полное копирование от инкрементного?
3. Что такое разностное копирование?
4. Что такое удалённое управление компьютером?
5. Какие инструменты администрирования в Windows используются для управления удалёнными подключениями к серверу?
6. Что такое диспетчер служб терминалов?

Список литературы

1. Бобынцев, Д. О. Основы администрирования информационных систем : учебное пособие : [16+] / Д. О. Бобынцев, А. Л. Марухленко, Л. О. Марухленко и др. – Москва ; Берлин : Директ-Медиа, 2021. – 201 с. : ил., табл. – Режим доступа: по подписке. –

URL: <https://biblioclub.ru/index.php?page=book&id=598955> (дата обращения: 20.01.2021). – Библиогр. в кн. – ISBN 978-5-4499-1674-7. – DOI 10.23681/598955. – Текст : электронный.

2. Проскуряков, А. В. Компьютерные сети: основы построения компьютерных сетей и телекоммуникаций : [16+] / А. В. Проскуряков ; Министерство науки и высшего образования Российской Федерации, Южный федеральный университет, Инженерно-технологическая академия. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2018. – 202 с. : ил. – Режим доступа: по подписке. –

URL: <https://biblioclub.ru/index.php?page=book&id=561238> (дата обращения: 20.01.2021). – Библиогр.: с. 195-196. – ISBN 978-5-9275-2792-2. – Текст : электронный.

3. Басыня, Е. А. Системное администрирование и информационная безопасность : учебное пособие : [16+] / Е. А. Басыня ; Новосибирский государственный технический университет. – Новосибирск : Новосибирский государственный технический университет, 2018. – 79 с. : ил. – Режим доступа: по подписке. –

URL: <https://biblioclub.ru/index.php?page=book&id=575325> (дата обращения: 20.01.2021). – Библиогр. в кн. – ISBN 978-5-7782-3484-0. – Текст : электронный.

4. Олифер, Виктор Григорьевич. Компьютерные сети. Принципы, технологии, протоколы : учебник для вузов / В. Г. Олифер, Н. А. Олифер. - 4-е изд. - Санкт-Петербург : Питер, 2015. - 943 с. - (Учебник для вузов). - Библиогр.: с. 917. - Алф. указ.: с. 918. - ISBN 978-5-496-00004-8 : 455.99 р. - Текст : непосредственный.