Федеральное агентство связи Сибирский государственный университет телекоммуникаций и информатики Кафедра прикладной математики и кибернетики (ПМ и К)

Лабораторная работа «Виджет»

по дисциплине "Программирование для мобильных устройств"

Выполнил: студент ИВТ,

группы ИП-813

Бурдуковский И. А.

Проверила:

Ассистент кафедры ПМиК

Павлова У.В.

Новосибирск, 2021 г.

Оглавление

[Задание 3](#_Toc67656391)

[Теория 3](#_Toc67656392)

[Реализация приложения 4](#_Toc67656393)

[Код программы 4](#_Toc67656394)

# Задание

Виджет должен отображать котировки металлов на текущую дату по данным ЦБ. Данные подгружаются из http://www.cbr.ru/scripts/xml\_metall.asp?date\_req1=**13/07/2020**&date\_req2=**13/07/2020**   
На виджете отображается только данные по одному металлу. по нажатию на виджет открывается окно, где указаны данные по четырем металлам.

# Теория

[XmlPullParser](http://developer.android.com/reference/org/xmlpull/v1/XmlPullParser.html) – XML-парсер, который можно использовать для разбора XML документа. Принцип его работы заключается в том, что он пробегает весь документ, останавливаясь на его элементах. Но пробегает он не сам, а с помощью метода [next](http://developer.android.com/reference/org/xmlpull/v1/XmlPullParser.html" \l "next()" \t "_blank). Мы постоянно вызываем метод next и с помощью метода [getEventType](http://developer.android.com/reference/org/xmlpull/v1/XmlPullParser.html" \l "getEventType()" \t "_blank)проверяем, на каком элементе парсер остановился.

Основные элементы документа, которые ловит парсер:

[START\_DOCUMENT](http://developer.android.com/reference/org/xmlpull/v1/XmlPullParser.html#START_DOCUMENT)– начало документа

[START\_TAG](http://developer.android.com/reference/org/xmlpull/v1/XmlPullParser.html#START_TAG)– начало тэга

[TEXT](http://developer.android.com/reference/org/xmlpull/v1/XmlPullParser.html#TEXT)– содержимое элемента

[END\_TAG](http://developer.android.com/reference/org/xmlpull/v1/XmlPullParser.html#END_TAG)– конец тэга

[END\_DOCUMENT](http://developer.android.com/reference/org/xmlpull/v1/XmlPullParser.html#END_DOCUMENT)– конец документа

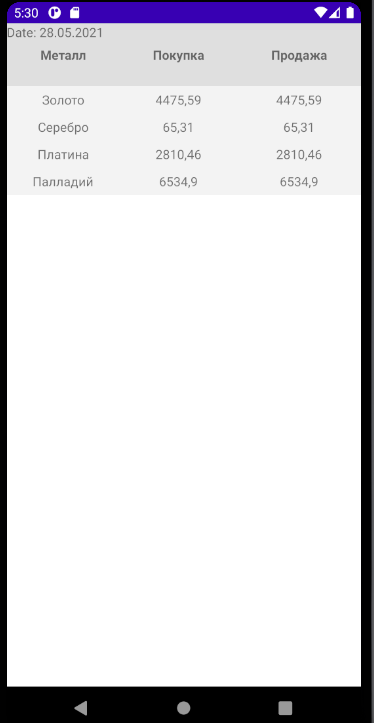
 onCreate мы получаем XmlPullParser с помощью метода prepareXpp и начинаем его разбирать. Затем в цикле while мы запускаем прогон документа, пока не достигнем конца - END\_DOCUMENT. Прогон обеспечивается методом next в конце цикла while. В switch мы проверяем на каком элементе остановился парсер.

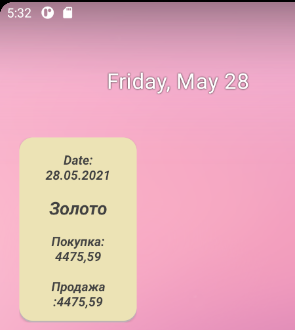
В методе prepareXpp мы подготавливаем XmlPullParser. Для этого вытаскиваем данные из папки res/xml. Это аналогично вытаскиванию строк или картинок – сначала получаем доступ к ресурсам (getResources), затем вызываем метод, соответствующий ресурсу. В нашем случае это - метод [getXml](http://developer.android.com/reference/android/content/res/Resources.html" \l "getXml(int)" \t "_blank). Но возвращает он не xml-строку , а готовый XmlPullParser.

Список, состоящий из текущего метода, метода, который его вызвал, его вызвавшего метода и т.д., называется *stack trace*.

Класс BufferedReader считывает текст из символьного потока ввода, буферизируя прочитанные символы. Использование буфера призвано увеличить производительность чтения данных из потока.

# Реализация приложения





# Код программы

**MainActivity.java**

*package com.example.widget;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.res.XmlResourceParser;  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.util.Log;  
import android.view.Gravity;  
import android.widget.ImageView;  
import android.widget.TableLayout;  
import android.widget.TableRow;  
import android.widget.TextView;  
  
import org.xmlpull.v1.XmlPullParser;  
import org.xmlpull.v1.XmlPullParserException;  
import org.xmlpull.v1.XmlPullParserFactory;  
  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStreamReader;  
import java.io.StringReader;  
import java.net.HttpURLConnection;  
import java.net.MalformedURLException;  
import java.net.URL;  
import java.text.DateFormat;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.Date;  
import java.util.Locale;  
import java.util.logging.XMLFormatter;  
  
import javax.net.ssl.HttpsURLConnection;  
  
public class MainActivity extends AppCompatActivity {   
 private TableLayout table;   
 private TableRow.LayoutParams param;   
 class Record{  
 public int type;   
 public String sell;  
 public String buy;   
 Record(int t,String s,String b){  
 type = t;  
 sell = s;  
 buy = b;  
 }  
 }  
 Date secondDate;  
 private ArrayList<Record> list = new ArrayList<>();//список  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 table = findViewById(R.id.table);   
 param = new TableRow.LayoutParams(  
 TableRow.LayoutParams.WRAP\_CONTENT,  
 TableRow.LayoutParams.WRAP\_CONTENT,  
 1f  
 );   
 SendToBase();  
 }   
 public void newRow(Record r){  
 Log.d("tet","!");  
 TextView name = new TextView(MainActivity.this);  
 TextView buy = new TextView(MainActivity.this);  
 TextView sell = new TextView(MainActivity.this);  
 name.setLayoutParams(param);  
 name.setPadding(15, 15, 15, 15);  
 name.setGravity(Gravity.CENTER);  
 buy.setLayoutParams(param);  
 buy.setPadding(15, 15, 15, 15);  
 buy.setGravity(Gravity.CENTER);  
 sell.setPadding(15, 15, 15, 15);  
 sell.setLayoutParams(param);  
 sell.setGravity(Gravity.CENTER);  
 switch (r.type){  
 case 1:name.setText("Золото");break;  
 case 2:name.setText("Серебро");break;  
 case 3:name.setText("Платина");break;  
 case 4:name.setText("Палладий");break;  
  
 }  
 buy.setText(r.buy);  
 sell.setText(r.sell);  
 TableRow row = new TableRow(MainActivity.this);  
 row.addView(name);  
 row.addView(buy);  
 row.addView(sell);  
 table.addView(row);  
  
  
 }  
  
  
 public void SendToBase()  
 {  
  
 new Thread (new Runnable() {  
 @Override  
 public void run() {  
 int k = 0;  
 while (list.size() == 0){   
 Calendar cal = Calendar.getInstance();   
 Date currentTime = cal.getTime();  
 cal.add(Calendar.DATE, -k);   
 secondDate = cal.getTime();  
 DateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy", Locale.getDefault());  
 String date2 = dateFormat.format(currentTime);  
 String date1 = dateFormat.format(secondDate);  
 URL url;  
 String line = "";  
 HttpURLConnection connection = null;  
  
 k++;  
 try {   
 url = new URL("http://www.cbr.ru/scripts/xml\_metall.asp?date\_req1=" + date1 + "&date\_req2=" + date2);   
 connection = (HttpURLConnection) url.openConnection();  
 BufferedReader br = new BufferedReader(new InputStreamReader(connection.getInputStream()));  
 line = br.readLine();  
 Log.d("HTTP-GET", line);  
  
 } catch (IOException e) {  
 e.printStackTrace();  
 } finally {  
 if (connection != null) {  
 connection.disconnect();  
 }  
 }  
 //работа с тэгами  
 final String LOG\_TAG = "myLogs";  
 try {  
 XmlPullParserFactory factory = XmlPullParserFactory  
 .newInstance();  
 factory.setNamespaceAware(true);   
 XmlPullParser xpp = factory.newPullParser();  
 xpp.setInput(new StringReader(line));  
 int type = 0;  
 String sell = "";  
 String buy = "";  
 int state = 0;  
 while (xpp.getEventType() != XmlPullParser.END\_DOCUMENT) {  
 switch (xpp.getEventType()) {   
 case XmlPullParser.START\_DOCUMENT:  
 Log.d(LOG\_TAG, "START\_DOCUMENT");  
 break;   
 case XmlPullParser.START\_TAG:  
 if (xpp.getName().equals("Record")) {  
 state = 1;  
 } else if (xpp.getName().equals("Buy")) {  
 state = 2;  
 break;  
 } else if (xpp.getName().equals("Sell")) {  
 state = 3;  
 break;  
 }   
 Log.d(LOG\_TAG, "START\_TAG: name = " + xpp.getName()  
 + ", depth = " + xpp.getDepth() + ", attrCount = "  
 + xpp.getAttributeCount());  
 for (int i = 0; i < xpp.getAttributeCount(); i++) {  
 if (xpp.getAttributeName(i).equals("Code"))  
 type = Integer.parseInt(xpp.getAttributeValue(i));  
 }  
 break;   
 case XmlPullParser.END\_TAG:  
 if (xpp.getName().equals("Record")) {  
 Log.d(LOG\_TAG, "END\_TAG: name = " + xpp.getName());  
 Record temp = new Record(type, buy, sell);  
 if (temp != null)  
 list.add(temp);  
 type = 0;  
 buy = "";  
 sell = "";  
 }  
 break;   
 case XmlPullParser.TEXT:  
 if (state == 2) buy = xpp.getText();  
 else if (state == 3) sell = xpp.getText();  
 Log.d(LOG\_TAG, "text = " + xpp.getText());  
 break;  
  
 default:  
 break;  
 }   
 xpp.next();  
 }  
 Log.d(LOG\_TAG, "END\_DOCUMENT");  
  
 } catch (XmlPullParserException e) {  
 e.printStackTrace();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
 try {  
 runOnUiThread(new Runnable() {  
  
 @Override  
 public void run() {  
 for(Record el:list){  
 newRow(el);  
 }  
 TextView text = findViewById(R.id.date);  
 DateFormat dateFormat = new SimpleDateFormat("dd.MM.yyyy", Locale.getDefault());  
 String date3 = dateFormat.format(secondDate);  
 text.setText("Date: "+ date3);  
 }  
 });  
 Thread.sleep(300);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 }).start();  
 }  
}*

**MyWidget.java**

*package com.example.widget;  
  
import android.app.PendingIntent;  
import android.appwidget.AppWidgetManager;  
import android.appwidget.AppWidgetProvider;  
import android.content.Context;  
import android.content.Intent;  
import android.os.Build;  
import android.text.Html;  
import android.util.Log;  
import android.widget.Button;  
import android.widget.RemoteViews;  
import android.widget.TextView;  
  
import org.xmlpull.v1.XmlPullParser;  
import org.xmlpull.v1.XmlPullParserException;  
import org.xmlpull.v1.XmlPullParserFactory;  
  
import java.io.BufferedReader;  
import java.io.IOException;  
import java.io.InputStreamReader;  
import java.io.StringReader;  
import java.net.HttpURLConnection;  
import java.net.URL;  
import java.text.DateFormat;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Calendar;  
import java.util.Date;  
import java.util.Locale;  
  
  
public class MyWidget extends AppWidgetProvider {  
  
 class Record{  
 public int type;  
 public String sell;  
 public String buy;  
 Record(int t,String s,String b){  
 type = t;  
 sell = s;  
 buy = b;  
 }  
 }  
 Date secondDate;  
 String ans="Err\n Err";  
 private ArrayList<Record> list = new ArrayList<>();  
  
  
  
 @Override  
 public void onUpdate(Context context, AppWidgetManager appWidgetManager, int[] appWidgetIds) {  
 for (int appWidgetId : appWidgetIds) {  
 SendToBase();  
 try {  
 Thread.sleep(1000);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 Intent intent = new Intent(context,MainActivity.class);  
 PendingIntent pendingIntent = PendingIntent.getActivity(context,0,intent,0);  
 RemoteViews views = new RemoteViews(context.getPackageName(), R.layout.my\_widget);  
 views.setOnClickPendingIntent(R.id.button,pendingIntent);  
 views.setTextViewText(R.id.button,Html.fromHtml(ans));  
 appWidgetManager.updateAppWidget(appWidgetId, views);  
 }  
 }  
  
 public void SendToBase()  
 {  
  
 new Thread (new Runnable() {  
 @Override  
 public void run() {  
 int k = 0;  
 while (list.size() == 0){  
 Calendar cal = Calendar.getInstance();  
 Date currentTime = cal.getTime();  
 cal.add(Calendar.DATE, -k);  
 secondDate = cal.getTime();  
 DateFormat dateFormat = new SimpleDateFormat("dd/MM/yyyy", Locale.getDefault());  
 String date2 = dateFormat.format(currentTime);  
 String date1 = dateFormat.format(secondDate);  
 URL url;  
 String line = "";  
 HttpURLConnection connection = null;  
  
 k++;  
 try {  
 url = new URL("http://www.cbr.ru/scripts/xml\_metall.asp?date\_req1=" + date1 + "&date\_req2=" + date2);  
  
 connection = (HttpURLConnection) url.openConnection();  
 BufferedReader br = new BufferedReader(new InputStreamReader(connection.getInputStream()));  
  
 line = br.readLine();  
 Log.d("HTTP-GET", line);  
  
 } catch (IOException e) {  
 e.printStackTrace();  
 } finally {  
 if (connection != null) {  
 connection.disconnect();  
 }  
 }  
 final String LOG\_TAG = "myLogs";  
 try {  
 XmlPullParserFactory factory = XmlPullParserFactory  
 .newInstance();  
 factory.setNamespaceAware(true);  
 XmlPullParser xpp = factory.newPullParser();  
  
 xpp.setInput(new StringReader(line));  
 int type = 0;  
 String sell = "";  
 String buy = "";  
 int state = 0;  
 while (xpp.getEventType() != XmlPullParser.END\_DOCUMENT) {  
 switch (xpp.getEventType()) {   
 case XmlPullParser.START\_DOCUMENT:  
 Log.d(LOG\_TAG, "START\_DOCUMENT");  
 break;   
 case XmlPullParser.START\_TAG:  
 if (xpp.getName().equals("Record")) {  
 state = 1;  
 } else if (xpp.getName().equals("Buy")) {  
 state = 2;  
 break;  
 } else if (xpp.getName().equals("Sell")) {  
 state = 3;  
 break;  
 }  
 Log.d(LOG\_TAG, "START\_TAG: name = " + xpp.getName()  
 + ", depth = " + xpp.getDepth() + ", attrCount = "  
 + xpp.getAttributeCount());  
 for (int i = 0; i < xpp.getAttributeCount(); i++) {  
 if (xpp.getAttributeName(i).equals("Code"))  
 type = Integer.parseInt(xpp.getAttributeValue(i));  
 }  
 break;   
 case XmlPullParser.END\_TAG:  
 if (xpp.getName().equals("Record")) {  
 Log.d(LOG\_TAG, "END\_TAG: name = " + xpp.getName());  
 Record temp = new Record(type, buy, sell);  
 if (temp != null)  
 list.add(temp);  
 type = 0;  
 buy = "";  
 sell = "";  
 }  
 break;   
 case XmlPullParser.TEXT:  
 if (state == 2) buy = xpp.getText();  
 else if (state == 3) sell = xpp.getText();  
 Log.d(LOG\_TAG, "text = " + xpp.getText());  
 break;  
  
 default:  
 break;  
 }   
 xpp.next();  
 }  
 Log.d(LOG\_TAG, "END\_DOCUMENT");  
  
 } catch (XmlPullParserException e) {  
 e.printStackTrace();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 DateFormat dateFormat = new SimpleDateFormat("dd.MM.yyyy", Locale.getDefault());  
 String date3 = dateFormat.format(secondDate);  
  
 ans = "<p>Date: " +date3+"</p>"+"<h2>Золото<h2><p>Покупка: "+list.get(0).buy+"</p><p>Продажа :"+list.get(0).sell+"</p>";  
  
 }  
 }).start();  
 }  
  
}*

**activity\_main.xml**

*<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <TableLayout  
 android:id="@+id/table"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:background="#f3f3f3"  
 tools:ignore="MissingConstraints">  
 <TableRow>  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Металл"  
 android:background="#dfdfdf"  
 android:textStyle="bold"  
 android:layout\_weight="1"  
 android:gravity="center"  
 android:padding="25dp"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Покупка"  
 android:background="#dfdfdf"  
 android:textStyle="bold"  
 android:layout\_weight="1"  
 android:gravity="center"  
 android:padding="25dp"  
 />  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Продажа"  
 android:background="#dfdfdf"  
 android:textStyle="bold"  
 android:layout\_weight="1"  
 android:gravity="center"  
 android:padding="25dp"  
 />  
 </TableRow>  
  
 </TableLayout>  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Date : 26/03/2021"  
 android:id="@+id/date"  
 tools:ignore="MissingConstraints"  
 />  
  
</androidx.constraintlayout.widget.ConstraintLayout>*

**my\_widget.xml**

*<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/test"  
 android:textColor="@color/cardview\_dark\_background"  
 android:background="@drawable/widget\_style"  
 android:padding="15dp"  
 android:textStyle="bold|italic"  
 android:layout\_centerHorizontal="true"  
 android:layout\_centerVertical="true"  
 />  
  
</RelativeLayout>*