**package** non.walltimer;  
  
**import** android.bluetooth.BluetoothAdapter;  
**import** android.bluetooth.BluetoothDevice;  
**import** android.bluetooth.BluetoothSocket;  
**import** android.content.Intent;  
**import** android.net.Uri;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.TextView;  
**import** android.widget.TimePicker;  
  
**import** com.google.android.gms.appindexing.Action;  
**import** com.google.android.gms.appindexing.AppIndex;  
**import** com.google.android.gms.appindexing.Thing;  
**import** com.google.android.gms.common.api.GoogleApiClient;  
  
**import** java.io.IOException;  
**import** java.io.InputStream;  
**import** java.io.OutputStream;  
**import** java.lang.reflect.Method;  
**import** java.util.Set;  
**import** java.util.UUID;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 */\*\*  
 \* ATTENTION: This was auto-generated to implement the App Indexing API.  
 \* See https://g.co/AppIndexing/AndroidStudio for more information.  
 \*/* **private** GoogleApiClient **client**;  
  
 *//int REQUEST\_ENABLE\_BT=1;  
 //Button butSend=(Button) findViewById(R.id.buttonSend);* @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 TimePicker time = (TimePicker) findViewById(R.id.***timePick***);  
 time.setIs24HourView(**true**);  
  
 BluetoothAdapter mBluetoothAdapter = BluetoothAdapter.*getDefaultAdapter*();  
 **int** REQUEST\_ENABLE\_BT = 1;  
 Button butSend = (Button) findViewById(R.id.***buttonSend***);  
 **if** (mBluetoothAdapter == **null**) {*// Device does not support Bluetooth* butSend.setText(**"No Bt"**);  
 butSend.setClickable(**false**);  
 }  
 **if** (!mBluetoothAdapter.isEnabled()) {  
 Intent enableBtIntent = **new** Intent(BluetoothAdapter.***ACTION\_REQUEST\_ENABLE***);  
 startActivityForResult(enableBtIntent, REQUEST\_ENABLE\_BT);  
 }  
 Set<BluetoothDevice> pairedDevices = mBluetoothAdapter.getBondedDevices();  
 **for** (BluetoothDevice device : pairedDevices) {  
 String deviceName = device.getName();  
 String deviceHardwareAddress = device.getAddress(); *// MAC address* }  
 *// ATTENTION: This was auto-generated to implement the App Indexing API.  
 // See https://g.co/AppIndexing/AndroidStudio for more information.* **client** = **new** GoogleApiClient.Builder(**this**).addApi(AppIndex.***API***).build();  
 }  
  
 @Override  
 **protected void** onActivityResult(**int** req, **int** res, Intent enIntent) {  
 **int** REQUEST\_ENABLE\_BT = 1;  
 Button butSend = (Button) findViewById(R.id.***buttonSend***);  
 **super**.onActivityResult(req, res, enIntent);  
 **if** (req == REQUEST\_ENABLE\_BT) {  
 **if** (res == ***RESULT\_CANCELED***) {  
 butSend.setText(**"No Bt"**);  
 butSend.setClickable(**false**);  
 } **else** {  
 butSend.setText(**"Послать"**);  
 butSend.setClickable(**true**);  
 }  
 }  
 }  
  
 **public void** onBtSClk(View view) {  
 **boolean** exit=**false**;  
 TextView outlog = (TextView) findViewById(R.id.***textView***);  
 outlog.setText(**""**);  
 TimePicker time = (TimePicker) findViewById(R.id.***timePick***);  
 **int** hset = time.getCurrentHour(), mset = time.getCurrentMinute();*//Get timer values. Using old function due to low required Api.  
 //int hset = time.getHour(), mset = time.getMinute();* BluetoothAdapter mBluetoothAdapter = BluetoothAdapter.*getDefaultAdapter*();  
  
 Set<BluetoothDevice> pairedDevices = mBluetoothAdapter.getBondedDevices();  
 BluetoothDevice device = **null**;  
 **if** (pairedDevices.size() > 0) {  
 *// There are paired devices.* **for** (BluetoothDevice deviceinlist : pairedDevices) {  
 **if**(deviceinlist.getName().equals(**"HC-06"**)) {*//You should connect phone to HC-06 at first. Default password 1234.* device = deviceinlist;  
 }  
 }  
 }**else** {  
 outlog.setText(**"Some ship happened in list of paired devices"**);  
 exit=**true**;  
 }  
 **if**(device.equals(**null**)){  
 outlog.setText(**"Some ship happened in matching HC-06"**);  
 exit=**true**;  
 }  
 *//BluetoothDevice device = mBluetoothAdapter.getRemoteDevice("20:13:06:18:10:64");//You can connect without getting list of paired devices if you know MAC of your HC-06. (On different modules they're different).* UUID MY\_UUID = UUID.*fromString*(**"00001101-0000-1000-8000-00805F9B34FB"**);  
 BluetoothSocket Soc = **null**;  
 **if**(!exit)  
 **try** {*//try to create socket. As default HC-06 is host (or server).* Soc = device.createRfcommSocketToServiceRecord(MY\_UUID);  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 outlog.setText(**"Some ship happened during creating socket"**);  
 exit=**true**;  
 }  
 mBluetoothAdapter.cancelDiscovery();*//stop all discoveries in order to get less troubles during connection* **int** count=-1;  
 **if**(!exit)*//5 times try to connect. Works much(!) better then just one time.* **do** {  
 count++;  
 exit=**false**;  
 **try** {  
 Soc.connect();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 outlog.setText(**"Some ship happened during connecting"**);  
 exit = **true**;  
 }  
 }**while** ((exit)&&(count<5));  
  
 OutputStream outS = **null**;  
 **if**(!exit)  
 **try** {*//Creating output stream of socket* outS = Soc.getOutputStream();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 outlog.setText(**"Some ship happened during getting output stream"**);  
 exit=**true**;  
 }  
 **if**(!exit)  
 **try** {*//Try to write time to ouput stream  
 //outS.write(hset \* 60 + mset);//Sends integet* outS.write(Integer.*toString*(hset \* 60 + mset).getBytes());*//Sends string* } **catch** (IOException e) {  
 e.printStackTrace();  
 outlog.setText(**"Some ship happened during writing to output stream"**);  
 exit=**true**;  
 }  
 **try** {*//Flushing* outS.flush();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 outlog.setText(**"Some ship happened during flushing to output stream"**);  
 exit=**true**;  
 }  
 **if**(!exit){  
 outlog.setText(**"Time is Send"**);*//Everything is good)* **try** {  
 Soc.close();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 outlog.setText(**"Time is Send, but some ship happened during closing socket"**);*//But socket isn't closed. Disable bt to ensure, that connection lost.* }  
 }  
 }  
}