1、Android 的四大组件为：Activity,Service,ContentProvider,BroadcastReceiver;他们的作用如下：

activity 显示界面（显示的界面都是继承 activity 完成的）
service 服务（后台运行的，可以理解为没有界面的 activity）
Broadcast Receiver 广播（做广播，通知时候用到）
Content Provider 数据通信（数据之间通信，同个程序间数据，或者是不同程序间通信）

2、(对应代码文件夹 Second)

```
public void onClick(View v) {
        String quantity=tv2.getText().toString();
        switch (v.getId()) {
        case R.id.button1:
            int num =Integer.parseInt(quantity);
            if (num>0) {
                tv2.setText(String.valueOf(--num));
                tv4.setText("$"+String.valueOf(num*5));
            }
            break;
        case R.id.button2:
            int plus =Integer.parseInt(quantity);

                tv2.setText(String.valueOf(++plus));
                tv4.setText("$"+String.valueOf(plus*5));
            break;
        case R.id.order:
            new AlertDialog.Builder(this).setTitle("系统提示").setMessage("您选择了"+tv2.getText().toString()+"件共"+tv4.getText().toString()+"确定吗？").setPositiveButton("确定", new DialogInterface.OnClickListener() {

                @Override
                public void onClick(DialogInterface dialog, int which) {   }}).setNegativeButton("取消", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    }}).show();
            break;
        default:
            break;
        }
    }
```

3、(对应代码文件夹 Third)

```
public void onClick(View v) {
    String num  =tv2.getText().toString();
    int score = Integer.parseInt(num);
    String num_team2    =tv4.getText().toString();
    int score2 = Integer.parseInt(num_team2);
```

```
        switch (v.getId()) {
            case R.id.button1:
                tv2.setText(String.valueOf(score+3));
            break;
            case R.id.button2:
                tv2.setText(String.valueOf(score+2));
                break;
            case R.id.button3:
                break;
            case R.id.button4:
                tv4.setText(String.valueOf(score2+3));
                break;
            case R.id.button5:
                tv4.setText(String.valueOf(score2+2));
                break;
            case R.id.button6:
                break;
case R.id.reset:
                tv2.setText("0");
                tv4.setText("0");
                Editor editor=sp.edit();
                editor.clear();
                break;
        default:
            break;
        }

    }
    @Override
    protected void onPause() {
        super.onPause();
         Editor editor=sp.edit();
         editor.putString("score", tv2.getText().toString());
         editor.putString("score2", tv4.getText().toString());
         editor.commit();

    }
```

4、 (对应代码文件夹 Forth)update.setOnClickListener(new OnClickListener() {

```
            @Override
            public void onClick(View v) {
             String ett=et.getText().toString();
```

```
                if (!et.equals("")) {
                    lists.add(ett);
                    adapter.notifyDataSetChanged();
                }
            }
        });
```

5、Intent i = new Intent(一个 Activity.this,另一个 Activity.class);
startActivity(i);

6、Intent i = new Intent();
i.setAction(Intent.*ACTION_CALL*);
i.setData(Uri.*parse*("tel:10086"));
startActivity(i);

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

7、

```
public class MySqlite extends SQLiteOpenHelper{
    public MySqlite(Context context, String name, CursorFactory factory,
            int version) {
        super(context, name, factory, version);
    }
    private String createTable(){
        return "create table animals ("
                + "id  INTEGER PRIMARY KEY AUTOINCREMENT, "
                + "name VARCHAR(100), "
                + "description varchar(1000))";    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(createTable());
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
{
        db.execSQL("drop table if exists zoo");
        onCreate(db);
    }
}
```

在 Activity 的 onCreate()方法中加入

```
SQLiteOpenHelper dbHelper=new MySqlite(this, "zooOpenHelper.db", null, 1);
        SQLiteDatabase db = dbHelper.getWritableDatabase();
```

8、客户端：

```
1.  protected void connectServerWithTCPSocket() {
```

```
2.
3.          Socket socket;
4.          try {// 创建一个 Socket 对象，并指定服务端的 IP 及端口号
5.              socket = new Socket("192.168.1.32", 1989);
6.              // 创建一个 InputStream 用户读取要发送的文件。
7.              InputStream inputStream = new FileInputStream("e://a.txt");
8.              // 获取 Socket 的 OutputStream 对象用于发送数据。
9.              OutputStream outputStream = socket.getOutputStream();
10.             // 创建一个 byte 类型的 buffer 字节数组，用于存放读取的本地文件
11.             byte buffer[] = new byte[4 * 1024];
12.             int temp = 0;
13.             // 循环读取文件
14.             while ((temp = inputStream.read(buffer)) != -1) {
15.                 // 把数据写入到 OuputStream 对象中
16.                 outputStream.write(buffer, 0, temp);
17.             }
18.             // 发送读取的数据到服务端
19.             outputStream.flush();
20.
21.             /** 或创建一个报文，使用 BufferedWriter 写入,看你的需求 **/
22. //          String socketData = "[2143213;21343fjks;213]";
23. //          BufferedWriter writer = new BufferedWriter(new OutputStreamWrite
    r(
24. //                  socket.getOutputStream()));
25. //          writer.write(socketData.replace("\n", " ") + "\n");
26. //          writer.flush();
27.             /***********************************************/
28.         } catch (UnknownHostException e) {
29.             e.printStackTrace();
30.         } catch (IOException e) {
31.             e.printStackTrace();
32.         }
33.
34.     }
```

服务端：

[java] view plaincopy

```
1.  public void ServerReceviedByTcp() {
2.      // 声明一个 ServerSocket 对象
3.      ServerSocket serverSocket = null;
4.      try {
```

```
5.          // 创建一个 ServerSocket 对象，并让这个 Socket 在 1989 端口监听
6.          serverSocket = new ServerSocket(1989);
7.          // 调用 ServerSocket 的 accept()方法，接受客户端所发送的请求，
8.          // 如果客户端没有发送数据，那么该线程就停滞不继续
9.          Socket socket = serverSocket.accept();
10.         // 从 Socket 当中得到 InputStream 对象
11.         InputStream inputStream = socket.getInputStream();
12.         byte buffer[] = new byte[1024 * 4];
13.         int temp = 0;
14.         // 从 InputStream 当中读取客户端所发送的数据
15.         while ((temp = inputStream.read(buffer)) != -1) {
16.             System.out.println(new String(buffer, 0, temp));
17.         }
18.         serverSocket.close();
19.     } catch (IOException e) {
20.         e.printStackTrace();
21.     }
22. }
```

9、StrictMode 用途：最常用来捕捉应用程序的主线程，它将报告与线程及虚拟机相关的策略违例。一旦检测到策略违例（policy violation），你将获得警告，其包含了一个栈 trace 显示你的应用在何处发生违例。除了主线程，我们还可以在 Handler，AsyncTask，AsyncQueryHandler，IntentService 等 API 中使用 StrictMode。StrictMode 的线程策略主要用于检测磁盘 IO 和网络访问，而虚拟机策略主要用于检测内存泄漏现象；

意义：能让开发者尽快地了解程序的瑕疵，以提交程序的质量。

Lint 用途：Android lint 是在 ADT 16 提供的新工具，它是一个代码扫描工具，能够帮助我们识别代码结构存在的问题，主要包括：

1）布局性能（以前是 layoutopt 工具，可以解决无用布局、嵌套太多、布局太多）

2）未使用到资源

3）不一致的数组大小

4）国际化问题（硬编码）

5）图标的问题（重复的图标，错误的大小）

6）可用性问题（如不指定的文本字段的输入型）

7）manifest 文件的错误

意义：Android lint 是对 android 开发者很有帮助的一款工具，对于项目打包发布前优化代码、查找没用到的资源、查找错误等非常有帮助