Gson提供了fromJson() 和toJson() 两个直接用于解析和生成的方法，前者实现反序列化，后者实现了序列化

simpleObjToJson、simpleJsonToObj、complexObjToJson、complexJsonToObj、arrayToJson、listToJson、jsonToList、exposeObjToJson、exposeJsonToObj

// 创建gson对象，gson对象是内部无状态的，所以创建一个可以多次使用，可以想象成一个转换器

Gson gson = new Gson();

//转换成json只用直接放入对象就行，gson在内部会提取对象的类型信息

gson.toJson(1); // ==> 1

gson.toJson("abcd"); // ==> "abcd"

gson.toJson(new Long(10)); // ==> 10

int[] values = { 1 };

gson.toJson(values); // ==> [1]

// 反json化，此时需要传入类型参数,因为json中是没有保存关于java类型的信息的

int one = gson.fromJson("1", int.class);

Integer one = gson.fromJson("1", Integer.class);

Long one = gson.fromJson("1", Long.class);

String str = gson.fromJson("\"abc\"", String.class);

gson在序列化对象到json的过程中并没有保存对象的类型信息，所以在反序列化过程中，需要手动传入参数(就是你自己要清楚这个对象原来是什么)