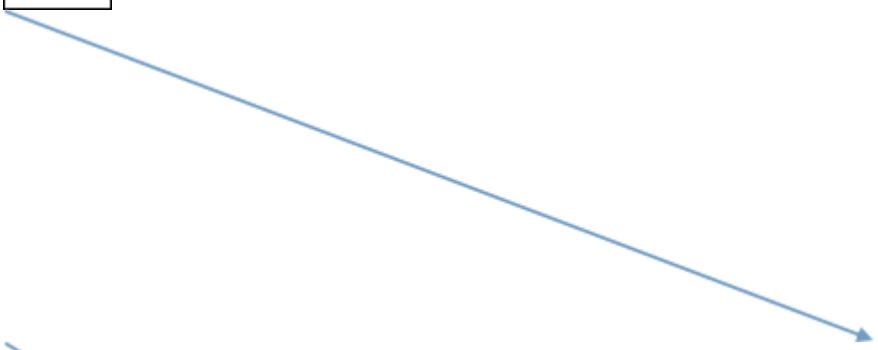
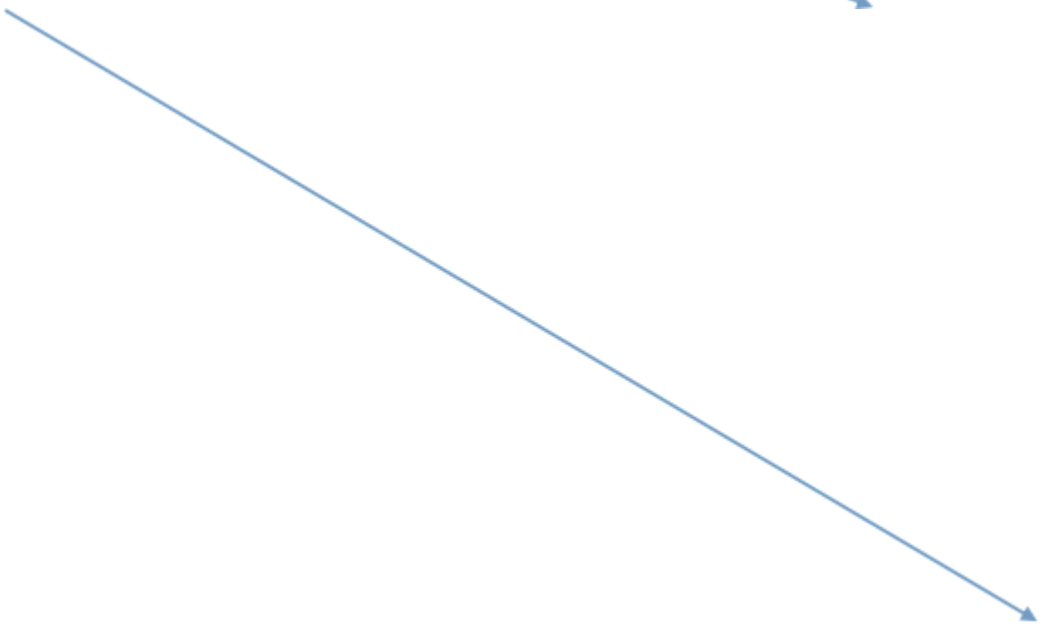
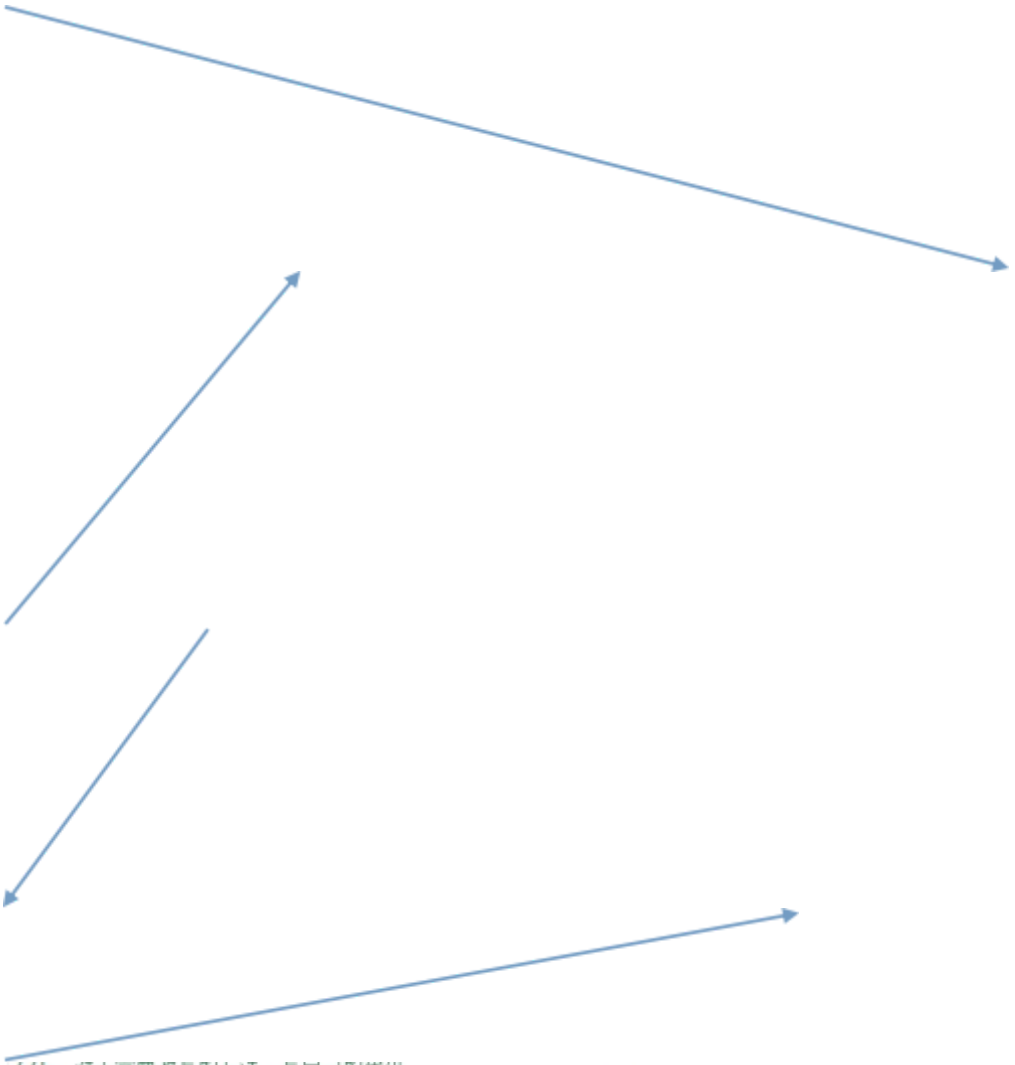
MyDataSource类

```
public Connection getConnection() throws SQLException {
    Connection conn = null;
    if(pool.size()>0){
        conn =  pool.removeFirst();//从池中取出一个连接
        MyConnection myConn = new MyConnection(conn,pool);//得到一个包装后的MyConnection对象
        return myConn;
    }else{
        //等待
        //新创建一个连接
        throw new RuntimeException("服务器忙。。。");
    }
}
```

6
5
4
3
2
1







```java
//5、对于需要改写的方法，写自己的代码
public class MyConnection implements Connection{

    private Connection oldConnection;//com.mysql.jdbc.Connection
    private LinkedList<Connection> pool;//连接池对象
    public MyConnection(Connection oldConnection,LinkedList<Connection> pool){
        this.oldConnection = oldConnection;//得到com.mysql.jdbc.Connection
        this.pool = pool;//得到连接池对象
    }

    public void close() throws SQLException {
        pool.addLast(oldConnection);
    }

    public PreparedStatement prepareStatement(String sql) throws SQLException {
        return oldConnection.prepareStatement(sql);
    }

    public <T> T unwrap(Class<T> iface) throws SQLException {
        return oldConnection.unwrap(iface);
```

```java
    public void test1(){
        Connection conn = null;
        PreparedStatement ps = null;
        DataSource ds = new MyDataSource();
        try {
            conn = ds.getConnection();//从池中取出一个连接 MyConnection
            ps = conn.prepareStatement("..");
//          ...
        } catch (SQLException e) {
            e.printStackTrace();
        }finally{
            try {
                conn.close();//该关就关闭，是否真的关了取决于 conn对象中怎么来的
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
```