```java
package me.gacl.demo;

import java.io.InputStream;
import java.io.PrintWriter;
import java.lang.reflect.InvocationHandler;
import java.lang.reflect.Method;
import java.lang.reflect.Proxy;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.LinkedList;
import java.util.Properties;
import javax.sql.DataSource;

/**
* @ClassName: JdbcPool
* @Description:编写数据库连接池
* @author: 孤傲苍狼
* @date: 2014-9-30 下午11:07:23
*
*/
public class JdbcPool implements DataSource{

    /**
    * @Field: listConnections
    *         使用LinkedList集合来存放数据库链接，
    *        由于要频繁读写List集合，所以这里使用LinkedList存储数据库连接比较合适
    */
    private static LinkedList<Connection> listConnections = new LinkedList<Connection>();

    static{
        //在静态代码块中加载db.properties数据库配置文件
        InputStream in =
JdbcPool.class.getClassLoader().getResourceAsStream("db.properties");
        Properties prop = new Properties();
        try {
            prop.load(in);
            String driver = prop.getProperty("driver");
            String url = prop.getProperty("url");
            String username = prop.getProperty("username");
            String password = prop.getProperty("password");
            //数据库连接池的初始化连接数大小
            int jdbcPoolInitSize =Integer.parseInt(prop.getProperty("jdbcPoolInitSize"));
            //加载数据库驱动
            Class.forName(driver);
            for (int i = 0; i < jdbcPoolInitSize; i++) {
                Connection conn = DriverManager.getConnection(url, username, password);
                System.out.println("获取到了链接" + conn);
                //将获取到的数据库连接加入到listConnections集合中，listConnections集合此时就是一个存放了数据库连接的连接池
                listConnections.add(conn);
            }

        } catch (Exception e) {
            throw new ExceptionInInitializerError(e);
        }
    }

    @Override
    public PrintWriter getLogWriter() throws SQLException {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public void setLogWriter(PrintWriter out) throws SQLException {
        // TODO Auto-generated method stub

    }

    @Override
    public void setLoginTimeout(int seconds) throws SQLException {
        // TODO Auto-generated method stub

    }

    @Override
    public int getLoginTimeout() throws SQLException {
        // TODO Auto-generated method stub
        return 0;
    }

    @Override
    public <T> T unwrap(Class<T> iface) throws SQLException {
        // TODO Auto-generated method stub
        return null;
    }

    @Override
    public boolean isWrapperFor(Class<?> iface) throws SQLException {
        // TODO Auto-generated method stub
        return false;
    }

    /* 获取数据库连接
     * @see javax.sql.DataSource#getConnection()
     */
    @Override
    public Connection getConnection() throws SQLException {
        //如果数据库连接池中的连接对象的个数大于0
        if (listConnections.size()>0) {
            //从listConnections集合中获取一个数据库连接
            final Connection conn = listConnections.removeFirst();
            System.out.println("listConnections数据库连接池大小是" + listConnections.size());
            //返回Connection对象的代理对象
            return (Connection) Proxy.newProxyInstance(JdbcPool.class.getClassLoader(), conn.getClass().getInterfaces(), new InvocationHandler(){
                @Override
                public Object invoke(Object proxy, Method method, Object[] args)
                        throws Throwable {
                    if(!method.getName().equals("close")){
                        return method.invoke(conn, args);
                    }else{
                        //如果调用的是Connection对象的close方法，就把conn还给数据库连接池
                        listConnections.add(conn);
                        System.out.println(conn + "被还给listConnections数据库连接池了！！");
                        System.out.println("listConnections数据库连接池大小为" + listConnections.size());
                        return null;
                    }
                }
            });
        }else {
            throw new RuntimeException("对不起，数据库忙");
        }
    }
```

```java

    @Override
    public Connection getConnection(String username, String password)
            throws SQLException {
        return null;
    }
}
```