



链滴

Java Io 实现增删改查

作者: [BenBenBen0000](#)

原文链接: <https://ld246.com/article/1607916087133>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



Entity

```
package studio.ben.trainssystem.entity;
```

```
/**  
 * @Author: Ben  
 * @Date: 2020/12/12 16:56  
 * User实体  
 */  
public class User {  
    Integer id;  
    String userName;  
    String password;  
  
    public User(String userName, String password) {  
        this.userName = userName;  
        this.password = password;  
    }  
  
    public User(Integer id, String userName, String password) {  
        this.id = id;  
        this.userName = userName;  
        this.password = password;  
    }  
  
    public Integer getId() {  
        return id;  
    }  
  
    public void setId(Integer id) {  
        this.id = id;  
    }  
}
```

```

public String getUsername() {
    return userName;
}

public void setUsername(String userName) {
    this.userName = userName;
}

public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}

@Override
public String toString() {
    return "User{" +
        "id='" + id + '\'' +
        ", userName='" + userName + '\'' +
        ", password='" + password + '\'' +
        '}';
}

public User() {
}
}

```

IUserDao

```

package studio.ben.trainsystem.dao;

import studio.ben.trainsystem.entity.User;

import java.util.List;

/**
 * @Author: Ben
 * @Date: 2020/12/12 17:00
 */
public interface IUserDao {

    /**
     * 增加用户IUserDao
     * @param user
     * @return boolean
     */
    boolean insertUserDao(User user);

    /**

```

```

* 根据用户id删除用户IUserDao
* @param id
* @return boolean
*/
boolean deleteUserDao(Integer id);

/**
* 根据用户id更改用户信息IUserDao
* @param user
* @param id
* @return boolean
*/
boolean updateUserDao(User user,Integer id) ;

/**
* 查询用户所有信息IUserDao
* @return List<User>
*/
List<User> selectUserDao() ;
}

```

insertUserDao

```

@Override
public boolean insertUserDao(User user) {
    try {
        BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter("TrainSystem\\User.txt",true));
        String returnString=String.format("%s,%s,%s",user.getId(),user.getUserName(),user.getPassword());
        bufferedWriter.write(returnString);
        bufferedWriter.newLine();
        bufferedWriter.close();
        return true;
    }catch (Exception e){
        System.out.println("文件增加失败");
        return false;
    }
}

```

deleteUserDao

```

@Override
public boolean deleteUserDao(Integer id) {
    try{
        BufferedReader bufferedReader = new BufferedReader(new FileReader("TrainSystem\\User.txt"));
        List<User> userList=new ArrayList<>();
        String string;
        while ((string=bufferedReader.readLine())!=null){
            String[] dates=string.split(",");
            User user=new User(Integer.parseInt(dates[0]),dates[1],dates[2]);
            userList.add(user);
        }
    }
}

```

```

        for (int i = 0; i <= userList.size()-1 ; i++) {
            User user=userList.get(i);
            if (user.getId().equals(id)){
                userList.remove(i);
            }
        }
        BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter("TrainSystem\\User.txt"));
        for (int i = 0; i <userList.size() ; i++) {
            User user=userList.get(i);
            String returnString=String.format("%s,%s,%s",user.getId(),user.getUserName(),user.getPassword());
            bufferedWriter.write(returnString);
            bufferedWriter.newLine();
        }
        bufferedReader.close();
        bufferedWriter.close();
        return true;
    }catch (Exception e){
        System.out.println("数据删除失败");
        return false;
    }
}

```

updateUserDao

@Override

```

    public boolean updateUserDao(User user, Integer id) {
        try{
            BufferedReader bufferedReader = new BufferedReader(new FileReader("TrainSystem\\User.txt"));
            List<User> userList=new ArrayList<>();
            String string;
            while ((string=bufferedReader.readLine())!=null){
                String[] dates=string.split(",");
                User returnUser=new User(Integer.parseInt(dates[0]),dates[1],dates[2]);
                System.out.println(returnUser);
                userList.add(returnUser);
            }
            for (int i = 0; i <=userList.size()-1 ; i++) {
                User returnUser=userList.get(i);
                if (returnUser.getId().equals(id)){
                    userList.set(i,user);
                }
            }
            BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter("TrainSystem\\User.txt"));
            for (int i = 0; i <userList.size() ; i++) {
                User returnUser=userList.get(i);
                String returnString=String.format("%s,%s,%s",returnUser.getId(),returnUser.getUserName(),returnUser.getPassword());
                bufferedWriter.write(returnString);
                bufferedWriter.newLine();
            }
        }
    }

```

```
        bufferedReader.close();
        bufferedWriter.close();
        return true;
    } catch (Exception e) {
        System.out.println("数据更新失败");
        return false;
    }
}
```

```
}
```

selectUserDao

@Override

```
public List<User> selectUserDao() {
    try {
        BufferedReader bufferedReader = new BufferedReader(new FileReader("TrainSystem\
User.txt"));
        String string;
        List<User> userList = new ArrayList<>();
        while ((string = bufferedReader.readLine()) != null) {
            String[] dates = string.split(",");
            User user = new User(Integer.parseInt(dates[0]), dates[1], dates[2]);
            userList.add(user);
        }
        bufferedReader.close();
        return userList;
    } catch (Exception e) {
        System.out.println("数据查询失败");
        return null;
    }
}
```