消息记录提供接口 **Message**，实例类 **MessageManager**，
提供 **Message**、**MessagewithOne**、**ChattingRecord** 类中字段的 **get** 方法

接口 **Message**

```
public interface Message{
    /**
     * 获取当前用户的全部消息记录
     * @param Userid      --用户id
     * @return 一个 ChattingRecord 类
     * @throws Exception 数据库连接异常
     */
    public ChattingRecord AllMessage(int Userid) throws Exception;
    /**
     * 获取当前用户与某一好友的全部消息记录
     * @param Userid      --用户id
     * @param Friendid    --好友id
     * @return 一个 MessagewithOne 类
     * @throws Exception 数据库连接异常
     */
    public MessagewithOne AllMessagewithOne(int Userid, int Friendid) throws Exception;
    /**
     * 获取当前用户与某一好友的未读消息记录
     * @param Userid      --用户id
     * @param Friendid    --好友id
     * @return 一个 MessagewithOne 类
     * @throws Exception 数据库连接异常
     */
    public MessagewithOne UnreadMessagewithOne(int Userid, int Friendid) throws Exception;
```

```
/**
 * 获取当前用户与某一好友的最近三条消息
 * @param Userid      --用户id
 * @param Friendid    --好友id
 * @return 含有最多三条消息的 MessagewithOne 类
 * @throws Exception 数据库连接异常
 */
public MessagewithOne LatestThreeMessagewithOne(int Userid, int Friendid) throws Exception;
/**
 * 删除当前用户的全部消息记录
 * @param Userid      --用户id
 * @return 删除操作是否成功
 * @throws Exception 数据库连接异常
 */
public boolean DeleteAllMessage(int Userid) throws Exception;
/**
 * 删除当前用户与某一好友的全部消息记录
 * @param Userid      --用户id
 * @param Friendid    --好友id
 * @return 删除操作是否成功
 * @throws Exception 数据库连接异常
 */
public boolean DeleteAllMessagewithOne(int Userid, int Friendid) throws Exception;
```

```
/**
 * 删除当前用户与某一好友消息记录中 id 为 Messageid 的消息
 * @param Userid      --用户id
 * @param Friendid    --好友id
 * @return 删除操作是否成功(若该记录先前已被删除仍返回 true,
 *         只有当该 Messageid 不存在时返回 false)
 * @throws Exception 数据库连接异常
 */
public boolean DeleteOneMessage(int Userid, int Friendid, int Messageid) throws Exception;
```

实例类 MessageManager

```java
/**
 * Message 接口的实例化
 * @author Harris
 * @see    Message
 */
public class MessageManager implements Message{
    @Override
    public ChattingRecord AllMessage(int Userid) throws Exception{…

    public MessagewithOne AllMessagewithOne(int Userid, int Friendid) throws Exception{…

    public MessagewithOne UnreadMessagewithOne(int Userid, int Friendid) throws Exception{…

    public MessagewithOne LatestThreeMessagewithOne(int Userid, int Friendid) throws Exception{…

    public boolean DeleteAllMessage(int Userid) throws Exception{…

    public boolean DeleteAllMessagewithOne(int Userid, int Friendid) throws Exception{…

    public boolean DeleteOneMessage(int Userid, int Friendid, int Messageid) throws Exception{…
```

Message、MessagewithOne、ChattingRecord 类

```java
/**
 * 消息存储与管理，包含 Message、MessagewithOne、ChattingRecord 类
 * 对成员变量提供 get 方法
 * @author Harris
 */
public class Message{
    public String MessageText;
    public String MessageTime;
    public boolean hasread;
    public boolean valid;
    public int id;
    /**…
    public String getMessageText(){…
    /**…
    public String getMessageTime(){…
    /**…
    public int getMessageid(){…
    /**…
    public boolean has_read(){…
    /**…
    public boolean is_valid(){…
}
```

```java
public class MessagewithOne{
    public Message[] ChatRecordwithOne;
    public String FriendName;
    public int FriendID;
    public int MessageNumberwithOne;
    /**…
    public Message[] getChatRecordwithone(){…
    /**…
    public String getFriendName(){…
    /**…
    public int getFriendID(){…
    /**…
    public int getMessageNumberwithOne(){…
}

public class ChattingRecord{
    public MessagewithOne[] ChatRecord;
    public String UserName;
    public int UserID;
    public int MessageNumberwithAll;
    /**…
    public MessagewithOne[] getChatRecord(){…
    /**…
    public String getUserName(){…
    /**…
    public int getUserID(){…
    /**…
    public int getMessageNumberwithAll(){…
}
```