一、实验目的

1) 掌握 String 类的常用方法；

2) 掌握使用 Scanner 类对象从字符串中解析程序所需要的数据；

3) 掌握 Date 类以及 Calendar 类的常用方法。

二、实验环境

Windows 7,Eclipse+jdk8 ,PC

三、实验内容与步骤

1．编写程序实现用户注册、登录、更改密码、注销和相关统计信息输出。包含 2 个类：User 类（UML 图如图 1 所示）和主类。

- 主类中有 3 个方法，判断 Email 地址合法性，提取字符串中相应信息和主方法。
- 在主类的方法 splitInformatin(String)中，从参数中提取账号（不妨就是学号）、姓名、出生日期、密码和 Email 地址，并调用方法判断提取出的 Email 地址的有效性，如果提取成功，创建 User 类对象，可以用对象数组实现。
- 在主类的主方法中，实现如下功能：

1) 从键盘读取多个（至少 3 个）字符串（用户注册信息），调用方法 splitInformatin(String)实现用户信息分离，然后调用 User 类中的方法 register 完成用户注册

2) 从键盘读取用户登录字符串（至少 2 个），调用 User 类中方法 login 完成登录功能

3) 从键盘读取用户更改密码字符串，调用 User 类的方法 updatePassword 完成用户修改密码功能。

4) 从键盘读取用户字符串，调用 User 类的方法 logout 完成用户注销功能。

5) 统计用户状态信息。

UserInformation
userNo：String **userName：String** **password：String** **birthday：Date** **age:short** **loginState:boolean** **email:String** **registerCount：Static int**
register(String,String,String,Date,String):void **login(String,String)):boolean** **logout(String):void** **updatePassword(String,String,String):boolean** **computeAge():void**

图 1　用户类

四、实验代码与实验结果及分析

实验代码：

文件 Demo.java

```
import java.util.*;
public class Demo {
    public static void main(String args[]) {
        UserInformation user[]=new UserInformation[10];
        Scanner reader=new Scanner(System.in);
        System.out.println("Enter the user registration string to end with End:");
        String information=reader.nextLine();
        while(true) {
            user[UserInformation.registerCount]=new UserInformation();
            splitInformatin(user[UserInformation.registerCount],information);
            information=reader.nextLine();
            if(information.equals("End")) break;
            }
        user[0].computeAge();
        System.out.println("Login account1:\nUser1 number:");
        String tempUser=reader.nextLine();
        System.out.println("password:");
        if(user[0].login(tempUser, reader.nextLine())) user[0].loginState=true;
        System.out.println("Login account2:\nUser2 number:");
        tempUser=reader.nextLine();
        System.out.println("password:");
        if(user[1].login(tempUser, reader.nextLine())) user[1].loginState=true;
        System.out.print("Update password for user1:");
        System.out.println("(Input as 17111205005,ABC,123456)");
        StringTokenizer analyze=new StringTokenizer(reader.nextLine(),",");
        tempUser=analyze.nextToken();
        String tempPassword1=analyze.nextToken();
        String tempPassword2=analyze.nextToken();
        if(user[0].updatePassword(tempUser, tempPassword1, tempPassword2))
user[0].password=tempPassword2;
        System.out.println("Input user number to logout account1:");
        user[0].logout(reader.nextLine());
        System.out.println("User status information table:");
        System.out.println("User\t\tName\tBirthday\tage\tEmail\t\tLogin state\t");
        for(int i=0;i<=UserInformation.registerCount;i++) {
            System.out.print(user[i].userNo+"\t");
            System.out.print(user[i].userName+"\t");
            System.out.print(String.format("%tY-%<tm-%<td",user[i].birthday)+"\t");
            System.out.print(user[i].age+"\t");
            System.out.print(user[i].email+"\t");
            if(user[i].loginState) System.out.println("Online");
            else System.out.println("Offline");
        }
}
static void splitInformatin(UserInformation users,String information) {
    StringTokenizer analyze=new StringTokenizer(information,",");
    String userNo=analyze.nextToken();
    String userName=analyze.nextToken();
    String birthday=analyze.nextToken();
    String password=analyze.nextToken();
    String email=analyze.nextToken();
    if(emailTest(email)) {
        Date date=null;
        String dateStr=birthday;
        String[] dateDivide=dateStr.split("-");
        if(dateDivide.length==3){
        int year = Integer.parseInt(dateDivide [0].trim());
        int month = Integer.parseInt(dateDivide [1].trim());
        int day = Integer.parseInt(dateDivide [2].trim());
        Calendar c = Calendar.getInstance();
        c.set(year, month-1, day);
        date = c.getTime();
        users.register(userNo, userName, password, date, email);
        UserInformation.registerCount++;
        }
    }
}
static boolean emailTest(String email) {
    int count=0;
    for (int i = 0; i < email.length(); i++) {
        char chat = email.charAt(i);
        if (chat=='@') {
            count++;
            }
        if (count>1) {
            return false;
            }
        }
    int pointIndex=email.lastIndexOf(".");
    int aindex=email.indexOf("@");
    if (pointIndex<aindex) {
        return false;
        }
    if (aindex+1>=pointIndex) {
        return false;
        }
    if (aindex==0) {
        return false;
        }
    return true;
    }
}
```

文件 UserInformation.java

```
import java.util.*;
public class UserInformation {
    String userNo;
    String userName;
    String password;
    Date birthday=null;
    short age;
    boolean loginState=false;
    String email;
    static int registerCount=0;
    void register(String userNo,String userName,String password,Date birthday,String email)
{
        this.userNo=userNo;
        this.userName=userName;
        this.password=password;
        this.birthday=birthday;
        this.email=email;
        computeAge();
    }
    boolean login(String userNo,String password) {
        if(this.userNo.equals(userNo)&&this.password.equals(password)) {
            System.out.println("Login succeed!");
            return true;
        }
        else{
            System.out.println("Login failed!");
            return false;
        }
    }
    void logout(String userNo) {
        if(this.userNo.equals(userNo)) {
            System.out.println("Logout succeed!");
            loginState=false;
        }
        else System.out.println("Logout failed!");
    }
    boolean updatePassword(String userNo,String passwordOld,String passwordNew) {
        if(this.userNo==userNo&&this.password==passwordOld) {
            System.out.println("Update succeed!");
            return true;
        }
        else {
            System.out.println("Update failed!");
            return false;
        }
    }
    void computeAge() {
        Calendar cal = Calendar.getInstance();
        Calendar ca2 = Calendar.getInstance();
        cal.setTime(new Date());
        ca2.setTime(birthday);
        age=(short)(cal.get(Calendar.YEAR)-ca2.get(Calendar.YEAR));
    }
}
```

实验结果及分析：

```
Enter the user registration string to end with End:
18111303011,Tom,2000-01-01,tom123,tom@qq.com↙
18111303012,Alice,2000-05-01,alice123,alice@fox.com↙
18111303013,Jack,2001-01-03,jack234,tom@live.com↙
End
Login account1:
User1 number:
18111303011↙
password:
tom123↙
Login succeed!
Login account2:
User2 number:
18111303012↙
password:
alice123↙
Login succeed!
Update password for user1:(Input as 17111205005,ABC,123456)
18111303011,tom223,tom345↙
Update failed!
Input user number to logout account1:
18111303011↙
Logout succeed!
```

```
User status information table:
User          Name      Birthday      age  Email             Login state
18111303011 Tom       2000-01-01    19   tom@qq.com        Offline
18111303012 Alice     2000-05-01    19   alice@fox.com     Online
18111303013 Jack      2001-01-03    18   tom@live.com      Offline
```

五、实验总结

实验要求的功能已实现，但在程序最后完成时报错

```
Exception in thread "main" java.lang.NullPointerException
	at Demo.main(Demo.java:35)
```

还有待改进