

1、定义一个用户类(User),用户名(username)和密码(password)是这个类的属性。

(1) 实例化两个用户,分别有不同的用户名和密码。

(2) 设计一个方法, 显示密码

示例:

```
A = User("liua", 12345)
```

```
B = User("hwh", 15389)
```

```
A.showPassword()
```

```
B.showPassword()
```

2、定义一个用户类(User),用户名(username)和密码(password)是这个类的属性。

- (1) 实例化两个用户,分别有不同的用户名和密码。
- (2) 设计一个方法, 显示密码
- (2) 设计一个方法, 修改密码

示例:

```
A = User("liua", 12345)
```

```
B = User("hwh", 15389)
```

```
A.changePassword(6666)
```

```
A.showPassword()
```

```
B.changePassword(9999)
```

```
B.showPassword()
```

3、编写一个代表矩形的类，其具备长、宽属性，求面积功能

(1) 实例化两个矩形,分别计算面积。

示例:

```
A = Circle(5,6)
```

```
B = Circle (1,7)
```

```
A.calArea()
```

```
A.showArea ()
```

输出: 30

(2678)4、老人的数目

- 给你一个下标从 0 开始的字符串 details 。details 中每个元素都是一位乘客的信息，信息用长度为 15 的字符串表示，表示方式如下：
 - 前十个字符是乘客的手机号码。
 - 接下来的一个字符是乘客的性别。
 - 接下来两个字符是乘客的年龄。
 - 最后两个字符是乘客的座位号。
 - 请你返回乘客中年龄 严格大于 60 岁 的人数。

示例 1:

输入： details = ["7868190130M7522","5303914400F9211","9273338290F4010"]

输出： 2

解释： 下标为 0 , 1 和 2 的乘客年龄分别为 75 , 92 和 40 。所以有 2 人年龄大于 60 岁。

示例 2:

输入： details = ["1313579440F2036","2921522980M5644"]

输出： 0

解释： 没有乘客的年龄大于 60 岁。

5、高龄员工统计

- 某一个公司里共有 n 名员工，按从 0 到 $n-1$ 编号。每个员工 i 的年龄为 $Ages[i]$ 岁；
- 公司要求你来统计年龄大于 $target$ 岁的员工数量，同时统计最大年龄；
- 给你一个下标从 0 开始、长度为 n 的非负整数数组 $Ages$ 和一个非负整数 $target$ 。
- 请你返回年龄大于 $target$ 的员工数量，最大年龄值

示例输入 1

ages = [35, 42, 58, 60, 29, 75]

target = 50

示例输入 2

ages = [35, 42, 58, 55, 68, 98, 88, 29, 75]

target = 66
