

# Exercise 04

## 9.7 (账户类Account)

设计一个名为Account的类，它包括：

- 一个名为id的int类型私有数据域（默认值都为0）。
- 一个名为balance的double类型私有数据域（默认值都为0）。
- 一个名为annualInterestRate的double类型私有数据域存储当前利率（默认值都为0）。假设所有的账户都有相同的利率。
- 一个名为dateCreated的Date类型的私有数据域，存储账户的开户日期。
- 一个用于创建默认账户类型的无参构造方法。
- 一个用于创建带特定id和初始余额的账户的构造方法。
- id、balance和annualInterestRate的访问器和修改器。
- dateCreated的访问器。
- 一个名为getMonthlyInterestRate()的方法，返回月利率。
- 一个名为getMonthlyInterest()的方法，返回月利息。
- 一个名为withDraw的方法，从账户提取特定数额。
- 一个名为deposit的方向账户存储特定数额。

画出该类的UML图并实现这个类。（提示：方法getMonthlyInterest()用于返回月利息，而不是利率。月利息是 $\text{balance} * \text{monthlyInterestRate}$ 。 $\text{monthlyInterestRate}$ 是 $\text{annualInterestRate} / 12$ 。注意， $\text{annualInterestRate}$ 是一个百分数，如4.5%，你需要将其除以100。）

编写一个程序，创建一个账户ID为1122、余额为20000美元、年利率为4.5%。使用withdraw方法从账户取出2500美元，用deposit方法存3000美元，打印余额、月利息以及这个账户的开户日期。

# Exercise 04

## 9.13 (位置类Location)

设计一个名为Location的类，画出该类的UML图，定位二维数组中的最大值及其位置。这个类包括公共的数据域row、column和maxValue，二维数组中的最大值及其下标用int型的row和column以及double型的maxValue存储。

编写下面的方法，返回一个二维数组中最大值的位置。

```
public static Location locationLargest(double [][] a)
```

返回值是一个Location的实例。编写一个测试程序，提示用户输入一个二维数组，然后显示这个数组中最大元素的位置。下面是一个运行示例：

Enter the number of rows and columns in the array: 3 4

Enter the array:

23.5 35 2 10

4.5 3 45 3.5

35 44 5.5 9.6

The location of the largest element is 45 at (1, 2)