IC-2101 Programación Orientada a Objetos **Laboratorio 1**

Prof: MSc. Samanta Ramijan Carmiol **Introducción a Java**

En clase hemos estudiado los conceptos básicos de clases, objetos o instancias, atributos, métodos (incluído el método constructor), literales, modificadores de visibilidad y la referencia this en Java. Para poner en práctica la teoría estudiada, con ayuda de una herramienta para hacer diagramas y el IDE de su preferencia, realice lo siguiente.

1. Un diagrama UML e implemente en Java la clase Cuenta:
    a. Atributo privado de tipo entero llamado `id` (valor por defecto 0).
    b. Atributo privado de tipo double llamado `balance` (valor por defecto 0).
    c. Atributo privado de tipo double llamado `tasaDeInteresAnual` (valor por defecto 0). Note que las tasas de interés son porcentuales. Asuma que todas las cuentas tienen la misma tasa de interés.
    d. Atributo privado de tipo Date llamado `fechaDeCreacion` que almacena la fecha cuando se crea la cuenta.
    e. Un constructor sin argumentos que crea la cuenta por defecto.
    f. Un constructor que toma los argumentos de `id` y un `balance` para crear una cuenta con ellos.
    g. Los métodos get y set (accessor y mutator) para `id`, `balance` y `tasaDeInteresAnual`.
    h. El método get para acceder a la `fechaDeCreacion`.
    i. Un método llamado `obtenerTasaDeInteresMensual()` que retorne el interés mensual.
    j. Un método llamado `calcularInteresMensual()` que retorne el interés mensual, calculado con la fórmula:

        `balance * tasaDeInteresMensual`.
    k. Un método llamado `retirarDinero` que disminuye una cantidad específica de dinero de la cuenta.
    l. Un método llamado `depositarDinero` que aumenta una cantidad específica de dinero a la cuenta.

2. Escriba una clase Main.java que le probar su código:
    a. Cree un objeto de tipo Cuenta, con el id 1122 y un balance de 500.000CRC y un interés anual de 4.5%.
    b. Utilice el método `depositarDinero` para sumar 150.000CRC al balance de la cuenta.
    c. Utilice el método `retirarDinero` para restar 200.000CRC al balance de la cuenta.
    d. Imprima en consola el balance, el interés mensual y la fecha de cuando se creó la cuenta.
    e. Cree un segundo objeto de tipo cuenta con los datos de su preferencia.
    f. Imprima en consola el balance, el interés mensual y la fecha de creación de la segunda cuenta.
3. A partir de la clase Cuenta cree una nueva clase ATM.java para simular el funcionamiento de un cajero automático.
    a. Cree diez cuentas en un arreglo con los id 0, 1, ..., 9 y el balance inicial de 100.000CRC.
    b. A través de la consola el programa le solicita al usuario ingresar un id, si el id es incorrecto, se le solicita al usuario ingresar nuevamente el id.
    c. Una vez el id es aceptado, el menú principal se despliega como en el ejemplo de corrida. Se puede ingresar la opción 1 para ver el balance actual, la 2 para retirar dinero, la 3 para depositar dinero y la 4 para salir del menú. Una vez se haya salido, el programa le solicita al usuario nuevamente ingresar un id, es decir que una vez que se inicie el programa no se detiene su ejecución por si sola.

```
Ingrese su id:  4 [enter]

Menú Principal
1. Ver el balance actual
2. Retirar Dinero
3. Depositar Dinero
4. Salir
```

Ingrese su selección. **1 [enter]**

El balance es de 100000.0

Menú Principal
1. Ver el balance actual

2. Retirar Dinero

3. Depositar Dinero

4. Salir

Ingrese su selección. **2 [enter]**

Ingrese una cantidad para retirar: **50000 [enter]**

Menú Principal
1. Ver el balance actual

2. Retirar Dinero

3. Depositar Dinero

4. Salir

Ingrese su selección. **1 [enter]**

El balance es de 50000.0

Menú Principal
1. Ver el balance actual

2. Retirar Dinero

3. Depositar Dinero

4. Salir

Ingrese su selección. **3 [enter]**

Ingrese una cantidad para depositar: **5000 [enter]**

Menú Principal
1. Ver el balance actual

2. Retirar Dinero

3. Depositar Dinero

4. Salir

Ingrese su selección. **1 [enter]**

```
El balance es de 55000.0


Menú Principal
1. Ver el balance actual
2. Retirar Dinero
3. Depositar Dinero
4. Salir
Ingrese su selección. 4 [enter]
El balance es de 55000.0


Ingrese su id:
```

4. Realice un diagrama UML de la clase ATM.java

Aspectos Administrativos

1. Límite para la entrega de la asignación: Jueves 22 de agosto a las 3pm.
2. Plataforma de revisión: Repositorio de código git
3. Cada archivo debe estar debidamente documentado con la información personal del estudiante que lo escriba, además de explicar su código e indicar cualquier referencia a código de terceros
4. Se debe incluir un archivo README que contenga el enunciado de los ejercicios.