Trabajo Práctico Nro. 4

Titular: Lic. Claudio O. Biale

1) Diseñe una clase llamada Persona y sus dos subclases llamadas Estudiante y Empleado. Las clases Docente y NoDocente son subclases de Empleado. Una persona tiene un nombre, dirección, número de teléfono y dirección de correo electrónico. Un estudiante tiene una situación de clase *(de primer año, segundo año, tercer año, cuarto año o quinto año)*. Un empleado tiene un salario y una fecha de contrato. Un Docente tiene un cargo *(ayudante de primera, jefe de trabajos prácticos o adjunto)*, un no docente tiene un título. Se debe reemplazar el método *toString* en cada clase para mostrar el nombre de la clase y el nombre de la persona. No deben permitirse instancias de Persona y Empleado.

Dibuje el diagrama de clases y escriba un programa que cree un estudiante, un docente y un no docente e invoque a sus métodos *toString()*.

2) Implemente una clase pila que almacene una lista de objetos en un atributo *lista*. Debe implementar los siguientes métodos:

- *vacia(): boolean* Retorna verdadero si la pila esta vacía.
- *largo(): int* Retorna la cantidad de elementos que se encuentran en la pila.
- *cima(): Object* Retorna el elemento superior de la pila sin removerlo.
- *desapilar(): Object* Retorna y remueve el elemento superior de la pila.
- *apilar(o: Object): void* Agrega un nuevo elemento en la parte superior de la pila.
- *toString() : String* devuelve la siguiente cadena: *"Pila: <elementos de la pila>"*

Dibuje el diagrama de clases y escriba un programa que ingrese 5 enteros a la pila, muestre el elemento superior, invoque al método *toString()*, desapile un elemento, lo muestre e invoque al método *toString()*.

3) Tomando como base la clase Cuenta definida en el TP 2 – Punto 3, modifique la clase Cuenta:

- Agregue un nuevo atributo denominado *nombre* de tipo String para almacenar el nombre del cliente.
- Agregue un nuevo constructor que construya una cuenta con un determinado *id*, *nombre* y *saldo*.
- Agregue un nuevo atributo denominado *transacciones* cuyo tipo es ArrayList que almacena las transacciones de la cuenta. Cada transacción es una instancia de la clase Transaccion.
- La clase Transaccion se define con los atributos *fecha* de tipo Date, *tipo* de tipo char *(D = Deposito, R = Retiro)*, *cantidad* de tipo double. Tiene un constructor que recibe un valor para fecha, tipo y cantidad.
- Agregue los métodos accesores y modificadores en ambas clases.
- Modifique los métodos *retirar* y *depositar* para agregar una transacción a la lista de transacciones de la cuenta.
- Todas las demás propiedades y métodos son los mismos que en el ejercicio original.

Escriba un programa de prueba que cree una cuenta con tasa de interés anual del 5,00%, saldo de $ 1.000 e id 1234 a nombre de Florencia. Deposite $ 30, $ 40 y $ 50 en la cuenta y retire $ 5, $ 4 y $ 2 de la cuenta. Imprima un resumen de cuenta que muestre el nombre del titular de la cuenta, la tasa de interés, el saldo y todas las transacciones.

4) Realice una captura de pantalla de la ejecución desde la linea de comandos del punto 1 del TP 3.

5) Realice una captura de pantalla de la ejecución desde la linea de comandos del punto 2 del TP 3.

Opcionales:

6) Tomando como base la clase Cuenta definida en el TP 2 – Punto 3, cree dos subclases para Caja de Ahorro y Cuenta Corriente, una cuenta corriente tiene un importe que permite girar en descubierto, en el caso de la caja de ahorro no se tiene ese importe. Defina a la clase Cuenta como abstracta.

Dibuje el diagrama de clases y escriba un programa que cree una cuenta corriente y una caja de ahorro e invoque a sus métodos *toString()*.

7) Basándose en el punto dos, implemente una clase pila usando herencia, debe extender de ArrayList.

Dibuje el diagrama de clases y escriba un programa que ingrese 5 Strings a la pila, muestre el elemento superior, invoque al método *toString()*, desapile un elemento, lo muestre e invoque al método *toString()*.