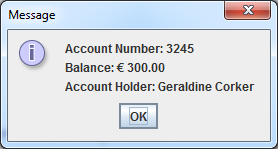
OOP Lab 2\_Supplemental

Bank Application

1. Write the class definition for a *BankAccount.* The *BankAccount* should have the following three **attributes**: accnum(String), balance(double) and customer. The data type of the customer attribute is Person, a class that has a single attribute called name, with appropriate accessor, mutator and toString methods (You do not need to write the code for the Person class). The *Bank Account* class should have **mutator** and **accessor** methods defined for all attributes as well as a **toString** method that uses the String format method to format its output. The class should have a **no-argument constructor** that sets up the attributes indirectly via a call to a second **3-argument constructor**. The 3 argument constructor accepts a person object as an argument. An example output of the **toString** method is given below.



1. Two additional methods are required to allow money be lodged and withdrawn from a Bank account object. Write appropriate code for these two methods.

1. Draw a **UML** class diagram (using Visio) for the **modified** *Bank Account* class.
2. Write a minimal **driver** class that creates an **array** of *Bank Account* objects. The driver should cater for adding multiple bank accounts, terminating on the user’s instruction.

Film Application

1. Write the class definition for a *Film.* The *Film* should have the following three **attributes**: title:String, director:String, duration:int. The class should have **mutator** and **accessor** methods defined for all attributes as well as a **toString** method that uses the **String format method** to prepare its output. The class should have a **no-argument constructor** that sets up the attributes indirectly via a call to a second **3-argument constructor**.
2. An additional **class** **variable** is required for the *Film* class that **tracks the number of Film objects created** using the class. The variable should have an **appropriate accessor method**. Indicate where appropriate modifications are required to your code written for a) and write the code.
3. Write a minimal **driver** class called **MyFlicks**, including any import statements, that creates and an **array** of Films called *catalog*. The values for each attribute of a Film should be requested from the user using **input dialog boxes**. The application should cater for multiple Films, terminating at the user’s request.
4. Write an additional class method called **displayFilms** for inclusion in the driver class written for part d) that displays the number of Films and the array of Films. The statement calling this method from the main method is **displayFilms();**. The output is displayed in a message dialog box containing a **JTextArea**. An example output is shown below.

