



Assignment 1

Simple ATM

Objectives

- Getting Started with JAVA Programming Language.
- Becoming familiar with Java IDE (e.g., eclipse, NetBeans, ...).
- Develop a simple GUI to input and display application data.
- Apply OOP principles to your GUI code.

Description

It is required to implement a simple ATM that supports the following:

- Withdrawal, Deposit and Balance Inquiry.
- History navigation through the last 5 transactions.
- First you need to validate card number against a pre-defined number in your system.
- The main menu contains:
 1. Five buttons: Withdrawal, Deposit, Balance Inquiry, Previous and Next (the last two for history navigation)
 2. One label: to show either your current balance when hitting **Balance Inquiry** button or history navigation when hitting **Previous** or **Next** buttons.
- When hitting **Withdraw** or **Deposit** buttons, user should be navigated to another window to enter the amount. You can use **buttons** representing numbers from 0 to 9 to take the input and **label** to show it.
- History should be updated after every successful transaction.
- Handle different special cases and show error dialogs when necessary (e.g. withdrawal amount is greater than the current balance, etc...).
- **[You should create at least 2 classes]** The best design is to create a class/classes and make sure all the logic is working, then create a new class/classes for the GUI.
- It is better to create an interface in the middle between logic class and GUI class/classes.
- **[Hint]** You can make a separate class called **Transaction** to hold the transaction type and value.



References

- Java Tutorial: [Link1](#), [Link2](#), [Link3](#).
- GUI Tutorial: [Link1](#), [Link2](#), [Link3](#), [Link4](#).
- GUI Swing basics: [Link1](#), [Link2](#).
- GUI SWT & JavaFX basics: [Link1](#), [Link2](#).
- Google is your best friend 😊

Proposed Interface

[Not mandatory] We shall propose this interface for you, you can use it if you want, you can make any changes you see fit to your design.

```
Public interface ATM {  
    //Return the current balance in string format  
    public String getCurrentBalance();  
    //withdraws from current balance and updates it  
    public void withdraw(String amount);  
    //adds to the current balance and updates it  
    public void deposit(String amount);  
    //returns the prev transaction in String format, or Null if no more history  
    public String prev();  
    //returns the next transaction in String format, or Null if no more history  
    public String next();  
}
```

And use it like this:

```
Public class MyATM implements ATM {  
    /* must override all the interface's functions */  
}
```



Deliverables

- Develop this assignment in Java programming language.
- You should submit your code online at the deadline, wait for the submission instructions on Piazza.
- You can use any graphics libraries (e.g. Swing, SWT, FX, ...), but text-mode user interface is not allowed.
- You should work individually.
- Late submission is accepted for only one week.
- **[Cheating Policy]** Delivering a copy will be severely penalized for both parties, so delivering nothing is so much better than delivering a copy.
- **[Late Policy]** Late delivery of the code will be allowed until **3 days** after the deadline, with a penalty of 30% of the final mark.

Good Luck 😊