

Automated Teller Machine (ATM) Team #2

Team Member #1: Phuoc Le

Team Member #2: Sydney Graham

Team Member #3: Tim Morris

Team Member #4: Veasna Bun

AutomatedTellerMachine Class

For our team project, we will be creating an object class called AutomatedTellerMachine. Our program will allow customer to complete transaction without the aid of a bank representative. Those who have a debit or credit card will be able to call the object AutomatedTellerMachine from the AtmData class. This object program will be useful because it will speed up the transactions process between banks and their customers. The following is our take and thought process of implementing Automated Teller Machine (ATM) in the Java language program.

In the AutomatedTellerMachine class have four properties: String firstName, String lastName, double totalBalance, and integer pinNumber.

```
public String firstName;  
public String lastName;  
public double totalBalance;  
public int pinNumber;
```

The “AutomatedTellerMachine (String firstName, String lastName, int pinNumber)” take in three parameter and then store those three parameters in the follow properties. The totalBalance default at zero but can be set to any value by using the setBalance method. We will see this in the AtmData class.

```
this.firstName = firstName;  
this.lastName = lastName;  
this.pinNumber = pinNumber;
```

Since firstName and lastName are immutable our team did not have getter or setter method for these properties but instead had getter and setter methods for totalBalance and pinNumber properties.

```
public double getBalance(double newBalance) {  
    return totalBalance;  
}  
public double setBalance(double newBalance) {  
    totalBalance = newBalance;  
    return totalBalance;  
}  
public double setPinNumber(int newpinNumber) {  
    pinNumber = newpinNumber;  
    return pinNumber;  
}  
public double getPinNumber() {return pinNumber;}
```

In the “`public double accountDeposit(double totalDeposit)`” and “`public double accountWithdraw(double totalWithdraw)`” method takes in input from the parameter and will set new value to the `totalBalance` properties. These methods will either add value to the `totalBalance` or subtract value to the `totalBalance`. You can see these condition in the following line of code:

```
public double accountDeposit(double totalDeposit) {
    totalBalance = totalBalance + totalDeposit;
    return totalBalance;
}
public double accountWithdraw(double totalWithdraw) {
    totalBalance = totalBalance - totalWithdraw;
    return totalBalance;
}
```

Our team also wanted to have a method that will display the account detail with the user `firstName`, `lastName`, and `totalBalance`. When trying to display the `totalBalance` the output would sometime display cent with either one or more than two digits even after casing it to a double. We were able to represent cent with only 2 digits and was able to do this by using “`System.out.printf("%.2f", totalBalance);`”.

```
public void accountDetail() {
    System.out.print(firstName + " " + lastName + " your account " + " has a total balance of $");
    System.out.printf("%.2f", totalBalance);
    System.out.println(".");
}
```

Finally, in the `AutomatedTellerMachine` class, the method “`public void bonusProgram(int guess)`” is a bonus feature added to this class. This feature gives the user the chance of earning extra cash when withdraw money and guessing the correct number. Our team decided to add this feature to give a fun and unique feature to the `AutomatedTellerMachine`, something different.

```
public void bonusProgram(int guess) {
    if (guess == Math.ceil((Math.random() * 20))) {
        System.out.println("You won the bonus with 50 $!");
        totalBalance += 50;
    } else {
        System.out.println("Unlucky, you did not guess the correct number.");
    }
}
```

AtmData Class

In the `AtmData` class is where the object `AutomatedTellerMachine` is created. Each object that is created are store in an array. It is in an array of objects. The parameter taken in here are unique and is its own. The only way to access the object is to input the correct `firstName`, `lastName`, and `pinNumber`. The user `firstName` and `lastName` are not case sensitive but the `pinNumber` will only take in integers.

Note: The user `firstName`, `lastName`, and `pinNumber` is like a debit and credit card. You need to have these protocol to access the `AutomatedTellerMachine`. The size of the array is 11 making this array having a total of 11 user.

Here is the list of users in the array list that you can try out to demonstrate our program.

(firstName, lastName, pinNumber)

- Veasna, Bun, 1234
- Tim, Morris, 5678
- Phuoc, Le, 910
- Sydney, Graham, 1112
- Veasna, B, 1234
- Sydney, Graham, 131415
- P, Le, 161718
- Morris, Tim, 1920212223
- Jonny, Tsunami, 1999
- Travis Scott, 2991
- Ash, Ketchum, 1997;

For our AtmData class we also set every account to have its own unique totalBalance.

- bankOfA.setBalance(2478.23); (Veasna Bun)
- bankOfB.setBalance(); will have a default balance of zero dollars and zero cent. (Tim Morris)
- bankOfC.setBalance(118.95); (Phuoc Le)
- bankOfD.setBalance(1235); (Sydney Graham)
- bankOfE.setBalance(2.00); (Veasna B)
- bankOfF.setBalance(50); (Sydney Graham, 131415)
- bankOfG.setBalance(100.00051); (P, Le)
- bankOfH.setBalance(0.01); (Morris Time)
- bankOfI.setBalance(1065.00); (Jonny Tsunami)
- bankOfJ.setBalance(45000000); (Travis Scott)
- bankOfK.setBalance(999.99); (Ash Ketchum)

For the most part of this Atm Data class our team spend most of it debugging compiler errors and adding feature like one that you may find in the real-world bank machine. Some of those features let the user:

- Check their account Balance.
- Deposit funds into your account.
- Withdraw funds from your account.
- Change their account pin number.

The team also spend most of the time finding ways to loop each of these features in AtmData class. Making sure that each parameter that require a certain type of input would not crash. We also found that "hasNextDouble ()" was most efficient because will not take in String input and will take it integer input. Knowing this help us causes the compiler to not crash when inappropriate input was entered from the user.

Team Member #1 Feedback: Phuoc Le Feedback

Everyone in the team knows what to do and which part they were assigned. We use discord to contact with each other which is convenient. The bonus system in my head is more

formal with more effect. I was only able to give bonus out without any effect at all. Practical ideas would be having some boxes or circle moving around or switch screen when a customer gets a bonus.

Team Member #2 Feedback: Sydney Graham

I was very open to other ideas since I had not fully fleshed out my garden aid idea. I enjoyed the idea of the AutomatedTellerMachine since it had a lot of potential to apply many of the concepts learned this quarter. Considering everyone hectic schedules we all were able to communicate effectively and work together to meet the deadline. For future improvements creating a timeline and delegating tasks for organizational purposes.

Team Member #3 Feedback: Tim Morris

Whoa...Let me first get a big breath of fresh air. I will say coding is a very basic and simple idea but should not in any way be underestimated. Studying foreign languages and trying to rely on what had worked in the past had not worked with Java coding. Developing and further comprehending a language not spoken every day in my mind is a far-fetched idea when considering Java coding. This is something that you must work at day in and day out. I believe everyone has their strengths and weaknesses regarding coding in any language.

Our team worked very well with the hand we were dealt. It is tough to be collaborative when everyone is has different goals, ambitions, and daily time, yet we were resilient. I would most likely be mistaken to say that ATMs use code like ours. This made our project so intriguing. We could see the basis of how one would work. This brings to question what areas in such could need robust safety measures and how having a solid PIN is of the utmost importance in keeping your electronic profiles secure. I forgot which recording we had first gone over loops in, which developed a function to find all prime numbers in an int. What I do remember is that our computers could do these calculations quickly. It gives me a better understanding of why we all need to understand computers as best as possible.

Great end project! As years go by, I will be interested in improving my coding skills by looking back to this project and seeing what else I could write to make it more closely resemble the code that we all rely on to keep daily funds secure.

Team Member #4 Feedback: Veasna Bun

From the start of this project, I realize that everyone will probably be busy studying for their final, so I did not want to stick with my original idea of making an object representing the motion of the human hand. So, I am glad that we ended up changing our object idea to AutomatedTellerMachine. I felt like this idea was more manageable with the amount of time limit we had for this class. Our team was able to accomplish and communicate our idea this by connecting through discord and sending file through digit message. One way to improve on this project will probably finding a way to add excel to manage the workload and add to-do list for every team member. I think this will work best so team member is less likely to work on the same time.

