Credit Name: CSE 3130 Object-Oriented Programming 2
Assignment Name: Account, PersonalAcct, BusinessAcct

By: Hielan Lee-Tremblay

Reflection Log

Understanding the Problem

To understand this project, I read the instructions carefully and identified that I needed an abstract parent class (Account) and two subclasses (PersonalAcct and BusinessAcct). I also noted the rules for minimum balances and associated fees ($2 for personal accounts below $100, $10 for business accounts below $500). I figured the assignment likely wanted to allow the user to interact with accounts via a menu, similar to previous projects.

Planning the Solution

Before coding, I made a plan: Create the abstract parent class Account with a private balance variable and an abstract withdraw() method. Create PersonalAcct and BusinessAcct subclasses that override withdraw() with their own rules. Build a menu driven main class (MainAccount) to allow the user to choose account type, enter a balance, make withdrawals, and repeat until quitting. Implement a helper method processWithdrawal() to demonstrate polymorphism, similar to payEmployee() in the UEmployee project.

Implementation

I wrote the code in small steps. First, I created the Account class with private balance and abstract withdraw(). Then I implemented PersonalAcct and BusinessAcct, adding the logic for minimum balance fees. Finally, I built the MainAccount class with a repeating menu and the helper method to process withdrawals.

Overcoming Challenges

The most challenging part was actual experimenting with this access modifier called protected (between private and public), because I could actual get my code to fully work with it which was interesting to me, but I didn’t want to deviate from the private classes we had used in our other mastery’s so it was more experimental then anything.

Learning

I learned more about how the protected access modifier works and how it doesn’t allow other classes to access it like public but it does allow subclasses in different or the same package to access it which I can see being very useful in the future.

Some resources I used

https://www.youtube.com/watch?v=HvPlEJ3LHgE

https://www.w3schools.com/java/java_polymorphism.asp

https://www.geeksforgeeks.org/java/overriding-in-java/