Project Details

This project extends the previous one and is designed for you to practice inheritance and abstract classes.

Same as the last project, you will write a program that allows a user to repeatedly create and manage express card accounts. However, we will add a type for each express card account. There are two types of express card accounts: a student express account or a faculty express account. All express account have fields `accountNumber`, `accountBalance`, `numberOfMeals`, `pricePerMeal`, `baseAmtForBonus` and `accountTypeName`.

Some changes for an express account are listed below:

- The criteria of receiving bonus for these two types of express accounts are different:
  - The `baseAmtForBonus` is set to $500.0 for a student express account.
  - The `baseAmtForBonus` is set to $0.0 for a faculty express account in the corresponding constructor.
- `rewardLevel` and `rewardAmt` are two variables only for a student express account for bonus receiving bonus. Same as in the previous project, for every deposit that is greater than the `baseAmtForBonus`, for each `rewardLevel`, a student account receives `rewardAmt` amount of money. For example, if `rewardLevel` is set to 200.0 and `rewardAmt` is set to 2.0, then a deposit of $500 to a $0 balance student express account will result $4.0 bonus and a new balance of $504.0. However, a deposit of $499 to a $100 balance student express account will not obtain any bonus and final balance is $599.0.

  The variable `rewardLevel` should be set to $200.0 and `rewardAmt` to $2.0. While we won't change the value of them in this project, they must be variables.
- A faculty express account has only one extra variable for bonus: `rewardPct`, the percentage of deposit that will become the bonus. In this project, it is set to 0.01. For example, since the `baseAmtForBonus` is $0.0 for a faculty, a $50 deposit to a $0 balance faculty express account will result $0.5 bonus and a new balance of $50.5.
- The `pricePerMeal` is $8.0 for a faculty account and $10.0 for a student.

With the above changes, when creating a new account, your program should ask the use to choose which type of the new account s/he wants to create. If the user chooses a student express account, then all the headers displayed in the sub-menu should be of the form "STUDENT EXPRESS ACCOUNT #0, BALANCE: $0.0, NUMBER OF MEALS: 0". Otherwise, the headers displayed in the sub-menu should be of the form "FACULTY EXPRESS ACCOUNT #0, BALANCE: $0.0, NUMBER OF MEALS: 0".

Implementation Hints

- Start from the `ExpressAccount` class that you have written from the previous project, remove `rewardLevel` and `rewardAmt` fields and corresponding methods from this class.
- Implement classes for the student and faculty express accounts that extend the `ExpressAccount` and implement the appropriate behavior for each type of account. These subclasses should reuse as much as possible from the base `ExpressAccount` class. If functionality is shared between

both account types, then it should be included in the `ExpressAccount` class (i.e., don't implement the same thing three times in the subclasses, implement it once in the `ExpressAccount` class). You may add constructors and protected setters to your `ExpressAccount` class; however, you should be able to reuse the methods `purchaseMeal` and `haveMeal` implemented in your `ExpressAccount` class.

- Same as before, you have two levels of menu: main menu and sub-menus. For each option in the main menu, there is a sub-menu for this option. However, when the user chooses to create a new account, your program should print out a sub-menu that asks the user to choose a type of the account.
- After you are done with the implementation, make sure you clean up your code. Remove unnecessary testing code and comments.
- All source files should include a description header, be properly indented and commented.

Code Skeleton

Your final java source code should include the following classes:

- `ExpAcct.java` : This is where your `main` method will be. It contains the whole logic of the application.
- `ExpressAccount.java` : This is the super class that contains the common properties and behaviors of the classes `FacultyExpressAccount` and `StudentExpressAccount`.
- `FacultyExpressAccount.java`
- `StudentExpressAccount.java`

Part of the code skeleton is shown as follows:

```
public class ExpAcct {
  ......
  public static void main(String[] args) {
    ......
  }
}

public abstract class ExpressAccount {
  ......
  public ExpressAccount(int accNumber) {
    ......
  }

  public int getAccountNumber() {
    ......
  }

  public double getAccountBalance() {
    ......
  }

  public double getBaseAmtForBonus() {
```

```
    ......
  }

  public double getPricePerMeal() {
    ......
  }

  public int getNumOfMeals() {
    ......
  }

  public String toString() {
    ......
  }

  ......
}

public class FacultyExpressAccount extends ExpressAccount {
  ......
}

public class StudentExpressAccount extends ExpressAccount {
  ......
}
```

What to turn in:

JAR your *.java files into a file called Project5.jar. Upload the jar file to Canvas under category Project5.