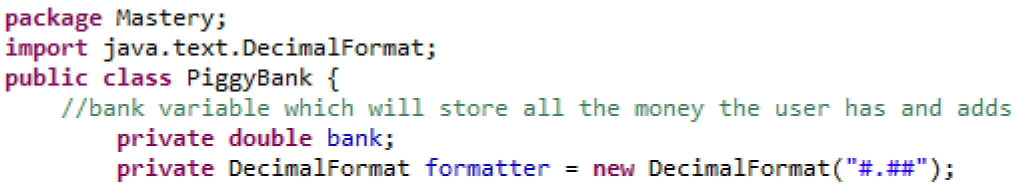
ReflectionLog: MySavings

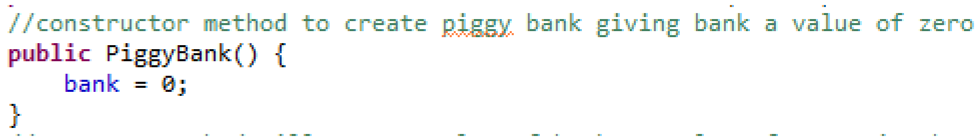
Youdis

PiggyBank class



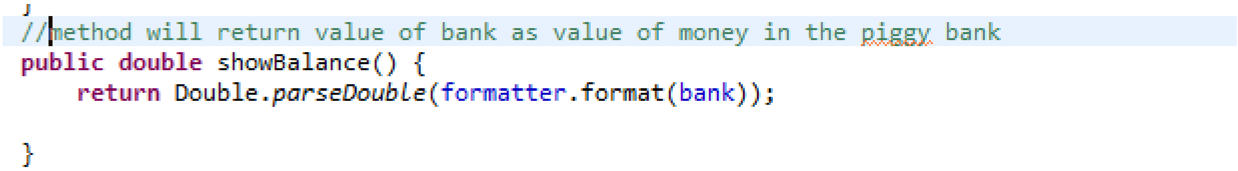
```java
package Mastery;
import java.text.DecimalFormat;
public class PiggyBank {
    //bank variable which will store all the money the user has and adds
        private double bank;
        private DecimalFormat formatter = new DecimalFormat("#.##");
```

Making a private bank variable to store the amount of money the user has. Make decimal format object to round, avoiding long decimal numbers.



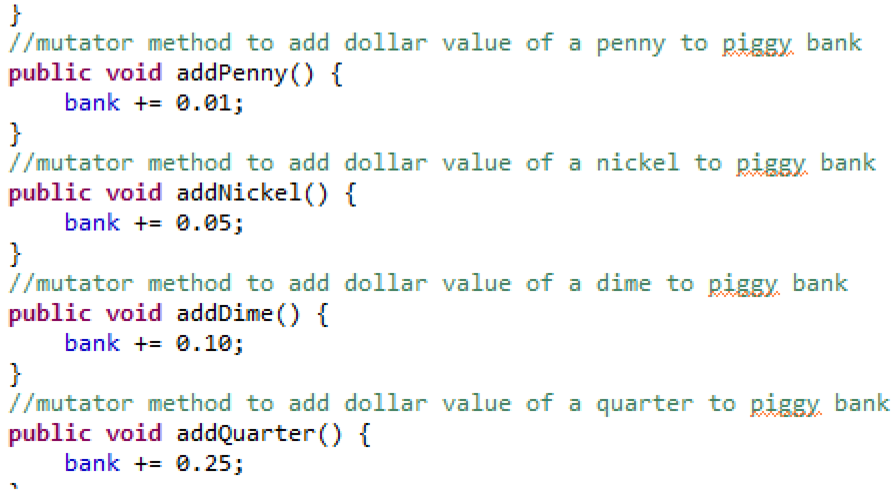
```java
  //constructor method to create piggy bank giving bank a value of zero
  public PiggyBank() {
      bank = 0;
  }
```

constructor method which will create a PiggyBank object which stores the variable bank and initializes it at zero to start.



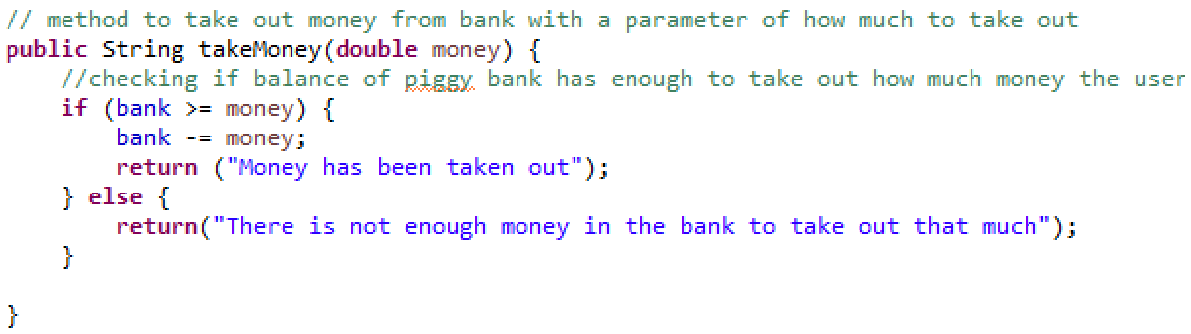
```java
 //method will return value of bank as value of money in the piggy bank
 public double showBalance() {
     return Double.parseDouble(formatter.format(bank));

 }
```

Accessor/get method that returns the value of the bank variable of the PiggyBank object which is the balance of the PiggyBank to where it was called.



```java
 }
 //mutator method to add dollar value of a penny to piggy bank
 public void addPenny() {
     bank += 0.01;
 }
 //mutator method to add dollar value of a nickel to piggy bank
 public void addNickel() {
     bank += 0.05;
 }
 //mutator method to add dollar value of a dime to piggy bank
 public void addDime() {
     bank += 0.10;
 }
 //mutator method to add dollar value of a quarter to piggy bank
 public void addQuarter() {
     bank += 0.25;
```

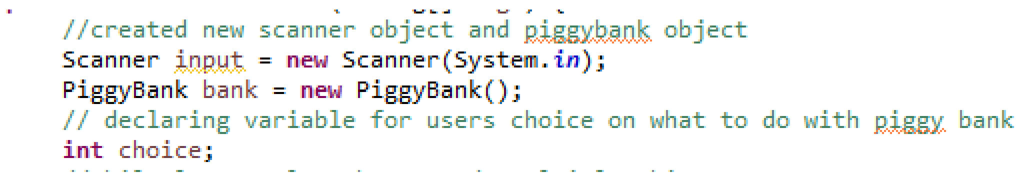
4 mutator/set methods which will add either a penny, nickel, dime, or quarter, depending on which one is called , to the PiggyBank object's bank variable.



```
// method to take out money from bank with a parameter of how much to take out
public String takeMoney(double money) {
    //checking if balance of piggy bank has enough to take out how much money the user
    if (bank >= money) {
        bank -= money;
        return ("Money has been taken out");
    } else {
        return("There is not enough money in the bank to take out that much");
    }

}
```

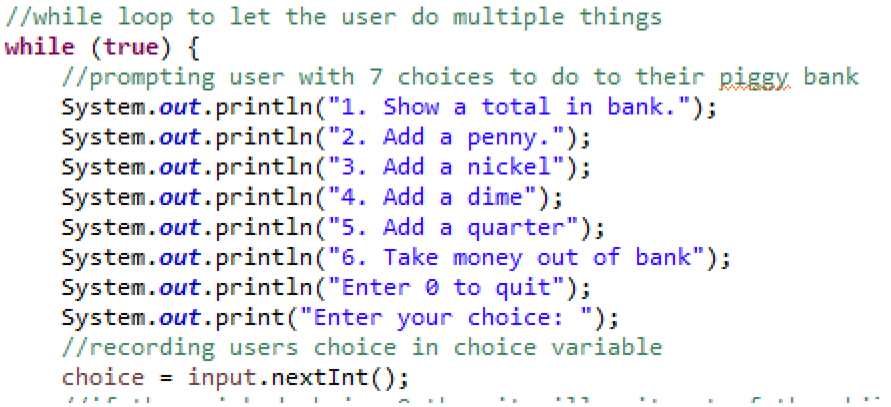
mutator/set method which will take money out of the bank if you have enough money, if not it will return a message saying you don’t have enough money in your piggy bank. It checks by seeing if the value given to the parameter is smaller than or equal to the bank variable.

MySavings



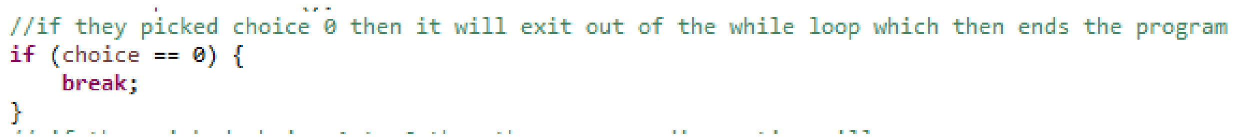
```
//created new scanner object and piggybank object
Scanner input = new Scanner(System.in);
PiggyBank bank = new PiggyBank();
// declaring variable for users choice on what to do with piggy bank
int choice;
```

Created Scanner and PiggyBank object to record inputs and to hold user’s money. Also declared choice variable to record user’s decisions on what to do with their bank.



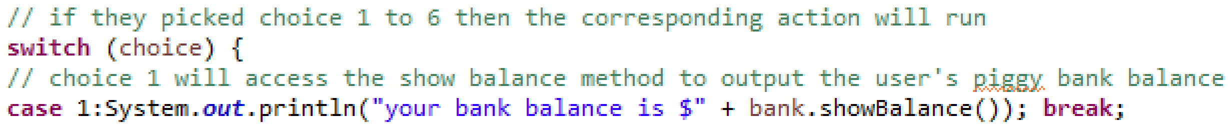
```
//while loop to let the user do multiple things
while (true) {
    //prompting user with 7 choices to do to their piggy bank
    System.out.println("1. Show a total in bank.");
    System.out.println("2. Add a penny.");
    System.out.println("3. Add a nickel");
    System.out.println("4. Add a dime");
    System.out.println("5. Add a quarter");
    System.out.println("6. Take money out of bank");
    System.out.println("Enter 0 to quit");
    System.out.print("Enter your choice: ");
    //recording users choice in choice variable
    choice = input.nextInt();
```

Created a while loop to let users do multiple actions to save. Prompts users with the 7 options to what they can do to their piggy bank. Which is show bank total, add penny, add nickel, add dime, add quarter, take money out, or quit. Then will initialize the user’s choice on what of the 7 choices to do with their bank.



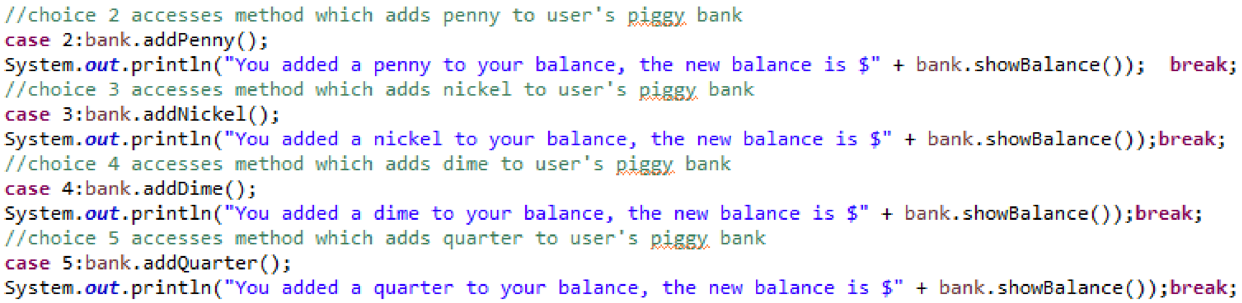
```java
//if they picked choice 0 then it will exit out of the while loop which then ends the program
if (choice == 0) {
    break;
}
```

If they picked the seventh choice by entering 0 the while loop will break, therefore ending the program.



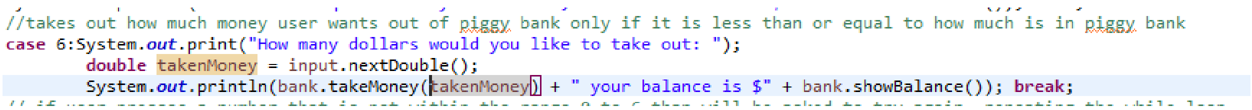
```java
// if they picked choice 1 to 6 then the corresponding action will run
switch (choice) {
// choice 1 will access the show balance method to output the user's piggy bank balance
case 1:System.out.println("your bank balance is $" + bank.showBalance()); break;
```

Then enter a switch case for the remaining 6 choices. Choice 1 will print the return value of the show balance method which will output the user's bank. The while loop will repeat after this going back to showcasing the 7 options.



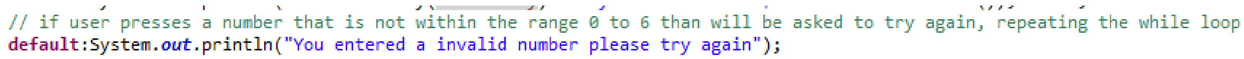
```java
//choice 2 accesses method which adds penny to user's piggy bank
case 2:bank.addPenny();
System.out.println("You added a penny to your balance, the new balance is $" + bank.showBalance());  break;
//choice 3 accesses method which adds nickel to user's piggy bank
case 3:bank.addNickel();
System.out.println("You added a nickel to your balance, the new balance is $" + bank.showBalance());break;
//choice 4 accesses method which adds dime to user's piggy bank
case 4:bank.addDime();
System.out.println("You added a dime to your balance, the new balance is $" + bank.showBalance());break;
//choice 5 accesses method which adds quarter to user's piggy bank
case 5:bank.addQuarter();
System.out.println("You added a quarter to your balance, the new balance is $" + bank.showBalance());break;
```

Choices 2 to 5 will call their respective method, add penny, nickel, dime, or quarter and then show the user’s bank balance by outputting the return value of showBalance method. The while loop will repeat after this going back to showcasing the 7 options.



```java
//takes out how much money user wants out of piggy bank only if it is less than or equal to how much is in piggy bank
case 6:System.out.print("How many dollars would you like to take out: ");
        double takenMoney = input.nextDouble();
        System.out.println(bank.takeMoney(takenMoney) + " your balance is $" + bank.showBalance()); break;
```

Choice 6 will ask the user to withdraw money and enter that value as the parameter for the takeMoney method. If the user has enough money in their bank account then will output money will be taken out. If not it will output not enough money in your bank balance. The while loop will repeat after this going back to showcasing the 7 options.



```java
// if user presses a number that is not within the range 0 to 6 than will be asked to try again, repeating the while loop
default:System.out.println("You entered a invalid number please try again");
```

If the user didn’t press a number 1-6 in the pick a choice option an error message will appear telling them to retry. Then the while loop will repeat going back to the showcasing of the 7 options.