MySavings Mastery ReflectionLog

Foreword: Screenshots were taken on both my home and school computer, hence the difference between the package in some of the screenshots.

Below is the first rendition of the TestMySavings class:

```java
package skibidi;

import java.util.Scanner;

public class TestMySavings {

    public static int input() {
        @SuppressWarnings("resource")
        //Prepare for user input
        Scanner userInput = new Scanner(System.in);
        //Record user input
        int choice = userInput.nextInt();
        //Returns user input
        return choice;
    }

    public static void main(String[] args) {

        do {

            //Prompt user for input
            System.out.println("1. Show 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.println("Enter your choice: ");

            //Calls PiggyBank Object from MySavings Class with the user input
            MySavings.PiggyBank(input());

        } while (input() != 0);
        //Loops while user input != 0

    }

}
```

Within the class are the main and input(); objects. The input object serves the purpose of recording and returning user input as an int. The main object includes a do while loop that prompts for user input, then calls the PiggyBank(int input); object from the MySavings class,

while the input(); object doesn't return 0. The main class includes a logical error within the do while loop, resulting in user input required twice in order to complete a cycle.

Below is the first rendition of the MySavings class:

```java
package skibidi;

public class MySavings {

    //Creates
    private static double totalValue = 0;

    public static void PiggyBank(int input) {
        switch (input) {
        case 0:
            break;
        case 1:
            System.out.println("You have " + totalValue + " in your piggybank.");
            break;
        case 2:
            System.out.println("Case2");
            totalValue += 0.01;
            break;
        case 3:
            System.out.println("Case3");
            totalValue += 0.05;
            break;
        case 4:
            System.out.println("Case4");
            totalValue += 0.1;
            break;
        case 5:
            System.out.println("Case5");
            totalValue += 0.25;
            break;
        case 6:
            System.out.println("Case6");
            totalValue = 0;
            break;
        }
    }

}
```

The class contains the variable totalValue and an object PiggyBank(int input); The PiggyBank object includes a switch statement that depends on the input variable. Depending on the case the totalValue is incremented by 0.01, 0.05, 0.1, 0.25, initialized to 0, or prints its current value.

Below is the an example of the output of the first rendition:

```
1. Show total in bank.
2. Add a penny.
3. Add a nickel.
4. Add a dime.
5. Add a quarter.
6. Take money out of bank.
Enter 0 to quit.
Enter your choice:
2
Case2
2
1. Show total in bank.
2. Add a penny.
3. Add a nickel.
4. Add a dime.
5. Add a quarter.
6. Take money out of bank.
Enter 0 to quit.
Enter your choice:
1
You have 0.01 in your piggybank.
2
1. Show total in bank.
2. Add a penny.
3. Add a nickel.
4. Add a dime.
5. Add a quarter.
6. Take money out of bank.
Enter 0 to quit.
Enter your choice:
5
Case5
1
1. Show total in bank.
2. Add a penny.
3. Add a nickel.
4. Add a dime.
5. Add a quarter.
6. Take money out of bank.
Enter 0 to quit.
Enter your choice:
1
You have 0.26 in your piggybank.
0
```

Below is the second rendition of the TestMySavings Class:

```java
package skibidi;

import java.util.Scanner;

public class TestMySavings {

    //Object that returns user input
    public static int selection() {
        @SuppressWarnings("resource")
        //Prepare for user input
        Scanner userInput = new Scanner(System.in);
        //Record user input
        int choice = userInput.nextInt();
        //Returns user input
        return choice;
    }

    public static void main(String[] args) {

        //Declares original input not as 0, so that the loop can begin
        int input = 10;

        //Loops while input is not 0
        while (input != 0) {


            //Prompt user for input
            System.out.println("1. Show 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.println("Enter your choice: ");

            //Declares input variable as selection() object
            input = selection();

            //Calls PiggyBank Object from MySavings Class with the input variable
            MySavings.PiggyBank(input);

        }

    }
}
```

My first rendition of changes fixed the logical error by declaring a new input variable, and only calling the selection(); object. Alongside this I improved the readability, by updating comments, and by changing the name of the input(); object to selection();.

Below is the second rendition of the MySavings class:

```java
package skibidi;

public class MySavings {

    //Creates public variable which represents the total value in the piggybank
    private static double totalValue = 0;

    public static void PiggyBank(int input) {
        //Switch statement depending on user input
        switch (input) {
        //Breaks switch
        case 0:
            break;
        case 1:
            //Prints total value in piggybank
            System.out.println("You have " + totalValue + " in your piggybank.");
            break;
        case 2:
            //Adds 1 cent to the piggybank
            totalValue += 0.01;
            break;
        case 3:
            //Adds 5 cents to the piggybank
            totalValue += 0.05;
            break;
        case 4:
            //Adds 10 cents to the piggybank
            totalValue += 0.1;
            break;
        case 5:
            //Adds 25 cents to the piggybank
            totalValue += 0.25;
            break;
        case 6:
            //Removes all coins from the piggybank
            totalValue = 0;
            break;
        }
    }

}
```

I made no changes to this class beyond readability by updating the comments.

Below is the an example of the output of the second rendition:

```
1. Show total in bank.
2. Add a penny.
3. Add a nickel.
4. Add a dime.
5. Add a quarter.
6. Take money out of bank.
Enter 0 to quit.
Enter your choice:
2
1. Show total in bank.
2. Add a penny.
3. Add a nickel.
4. Add a dime.
5. Add a quarter.
6. Take money out of bank.
Enter 0 to quit.
Enter your choice:
1
You have 0.01 in your piggybank.
1. Show total in bank.
2. Add a penny.
3. Add a nickel.
4. Add a dime.
5. Add a quarter.
6. Take money out of bank.
Enter 0 to quit.
Enter your choice:
6
1. Show total in bank.
2. Add a penny.
3. Add a nickel.
4. Add a dime.
5. Add a quarter.
6. Take money out of bank.
Enter 0 to quit.
Enter your choice:
1
You have 0.0 in your piggybank.
1. Show total in bank.
2. Add a penny.
3. Add a nickel.
4. Add a dime.
5. Add a quarter.
6. Take money out of bank.
Enter 0 to quit.
Enter your choice:
0
```

The error seen in the first rendition is no longer present.

Below is the final rendition of the TestMySavings class:

```
package Mastery;

import java.util.Scanner;

public class TestMySavings {

    //Object that returns user input
    public static int selection() {
        @SuppressWarnings("resource")
        //Prepare for user input
        Scanner userInput = new Scanner(System.in);
        //Record user input
        int choice = userInput.nextInt();
        //Returns user input
        return choice;
    }

    public static void main(String[] args) {

        //Declares input variable
        int input;

        //Loops while input is not 0
         do {
            //Prompt user for input
            System.out.println("1. Show 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.println("Enter your choice: ");

            //Declares input variable as selection() object
            input = selection();

            //Calls PiggyBank Object from MySavings Class with the input variable
            MySavings.PiggyBank(input);

        } while (input != 0);

    }

}
```

I improved the readability, and simplified the program by changing the while loop to a do while loop, this way I no longer need to pre-declare a value for input, and instead the condition is checked after the loop is run.

Below is the final rendition of the MySavings class:

```
package Mastery;

import java.text.DecimalFormat;

public class MySavings {

    //Decimal format, to round to 2 decimals
    private static final DecimalFormat df = new DecimalFormat("0.00");

    //Creates public variable which represents the total value in the piggybank
    private static double totalValue = 0;

    public static void PiggyBank(int input) {
        //Switch statement depending on user input
        switch (input) {
        //Breaks switch
        case 0:
            break;
        case 1:
            //Prints total value in piggybank
            System.out.println("You have " + df.format(totalValue) + " in your piggybank.");
            break;
        case 2:
            //Adds 1 cent to the piggybank
            totalValue += 0.01;
            break;
        case 3:
            //Adds 5 cents to the piggybank
            totalValue += 0.05;
            break;
        case 4:
            //Adds 10 cents to the piggybank
            totalValue += 0.1;
            break;
        case 5:
            //Adds 25 cents to the piggybank
            totalValue += 0.25;
            break;
        case 6:
            //Removes all coins from the piggybank
            totalValue = 0;
            break;
        }
    }

}
```

I added a decimal format object so that any output would be rounded to 2 decimal places, this prevents any unreasonably long outputs due to floating point math.

Below is the an example of the output of the final rendition:

```
1. Show total in bank.
2. Add a penny.
3. Add a nickel.
4. Add a dime.
5. Add a quarter.
6. Take money out of bank.
Enter 0 to quit.
Enter your choice:
2
1. Show total in bank.
2. Add a penny.
3. Add a nickel.
4. Add a dime.
5. Add a quarter.
6. Take money out of bank.
Enter 0 to quit.
Enter your choice:
1
You have 0.01 in your piggybank.
1. Show total in bank.
2. Add a penny.
3. Add a nickel.
4. Add a dime.
5. Add a quarter.
6. Take money out of bank.
Enter 0 to quit.
Enter your choice:
6
1. Show total in bank.
2. Add a penny.
3. Add a nickel.
4. Add a dime.
5. Add a quarter.
6. Take money out of bank.
Enter 0 to quit.
Enter your choice:
1
You have 0.00 in your piggybank.
1. Show total in bank.
2. Add a penny.
3. Add a nickel.
4. Add a dime.
5. Add a quarter.
6. Take money out of bank.
Enter 0 to quit.
Enter your choice:
0
```

No errors present.