

Java Foundations Practices - Section 6

Problem 1: Validate a Bank PIN

Develop a java program to validate bank PIN of a customer. Use a while loop to repeat code until a valid PIN is entered.

Task:

1. Declare a valid integer PIN.
2. Prompt the user to enter the PIN.
3. In a while loop, perform the following steps:
 - Compare the user-entered PIN with the already declared PIN
 - If the entered PIN is not the same, prompt the user to enter the PIN again
 - Repeat the loop until the correct PIN is entered
4. Print a message confirming that the correct PIN has been entered and that the user now has access to their account. The ValidatePin.java file is available to help you get started.

Code:

```
import java.util.Scanner;

public class ValidatePin {

    public static void main(String[] args) {

        int validPin = 1234;

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter your PIN: ");

        int enteredPin = scanner.nextInt();

        while (enteredPin != validPin) {

            System.out.print("Invalid PIN. Try again: ");

            enteredPin = scanner.nextInt();

        }

        System.out.println("Correct PIN entered. You now have access to your account.");

    }

}
```

Output:

```
Main.java
1 import java.util.Scanner;
2 public class ValidatePin {
3     public static void main(String[] args) {
4         int validPin = 1234;
5         Scanner scanner = new Scanner(System.in);
6         System.out.print("Enter your PIN: ");
7         int enteredPin = scanner.nextInt();
8         while (enteredPin != validPin) {
9             System.out.print("Invalid PIN. Try again: ");
10            enteredPin = scanner.nextInt();
11        }
12        System.out.println("Correct PIN entered. You now have access to your account.");
13    }
14 }
```

```
Output
java -cp /tmp/Y1eDJwcpP0/ValidatePin
Enter your PIN: 1111
Invalid PIN. Try again: 1221
Invalid PIN. Try again: 1234
Correct PIN entered. You now have access to your account.

=== Code Execution Successful ===
```

Problem 2: Displaying Multiples of a Number

Develop a java program to calculate the multiples of a given number using a for loop.

Task:

Have the user enter a number, and then use a for loop to display all the multiples of that number from 1 to 12.

Code:

```
import java.util.Scanner;

public class MultiplesCalculator {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");

        int number = scanner.nextInt();

        System.out.println("Multiples of " + number + ":");

        for (int i = 1; i <= 12; i++) {

            int multiple = number * i;

            System.out.println(number + " x " + i + " = " + multiple);

        }

    }

}
```

Output:

Problem 3: Programmatic ASCII Art

Using text to create a picture is known as ASCII art. In section 2, we made an ASCII art cat. This required us to type every character in the art we wanted to create. In this practice, you'll find a way to draw basic shapes programmatically in customizable sizes.

Task:

Task Complete the following two methods in LoopShape.java:

- createRectangle(): This method accepts two arguments for width and height which should be used to print a rectangle
- createTriangle(): This method accepts one argument for the size of a leg, which should be used to print an isosceles right triangle

Try changing the value of the arguments you're supplying these two methods from the main method. Make sure your program can successfully draw each shape to a custom size. Additionally, your program must:

- Refuse to draw shapes with any dimension less than 1
- Be able to draw shapes with any dimension equal to 1 (a 1x1 shape should print just a single character)

Code:

```
public class LoopShape {  
  
    public static void createRectangle(int width, int height) {  
  
        if (width < 1 || height < 1) {  
  
            System.out.println("Cannot draw a shape with dimensions less than 1.");  
  
            return;  
  
        }  
  
        for (int i = 0; i < width; i++) {  
  
            System.out.print("#");
```

```

    }
    System.out.println();
    for (int i = 0; i < height - 2; i++) {
        System.out.print("#");
        for (int j = 0; j < width - 2; j++) {
            System.out.print(" ");
        }
        System.out.println("#");
    }
    if (height > 1) {
        for (int i = 0; i < width; i++) {
            System.out.print("#");
        }
        System.out.println();
    }
}

public static void createTriangle(int size) {
    if (size < 1) {
        System.out.println("Cannot draw a shape with dimensions less than 1.");
        return;
    }
    for (int i = 0; i < size; i++) {
        for (int j = 0; j <= i; j++) {
            System.out.print("#");
        }
        System.out.println();
    }
}

public static void main(String[] args) {
    createRectangle(5, 4);
}

```

```

        System.out.println();

        createTriangle(5);

    }

}

```

Output:

The screenshot shows a Java IDE with a file named `Main.java`. The code defines a `LoopShape` class with a `createRectangle` method. This method checks if the width and height are greater than 1. If not, it prints an error message. If they are, it prints a rectangle of hash symbols (`#`) with the specified width and height. The `main` method calls `createRectangle(5, 5)` and `createTriangle(5)`. The output window shows the result of the execution, which is a 5x5 grid of hash symbols followed by a blank line and a triangle of 5 hash symbols. The execution was successful.

```

Main.java
1 public class LoopShape {
2     public static void createRectangle(int width, int height) {
3         if (width < 1 || height < 1) {
4             System.out.println("Cannot draw a shape with dimensions less than 1");
5             return;
6         }
7         for (int i = 0; i < width; i++) {
8             System.out.print("#");
9         }
10        System.out.println();
11        for (int i = 0; i < height - 2; i++) {
12            System.out.print("#");
13            for (int j = 0; j < width - 2; j++) {
14                System.out.print(" ");
15            }
16            System.out.println("#");
17        }
18        if (height > 1) {
19            for (int i = 0; i < width; i++) {
20                System.out.print("#");
21            }
22            System.out.println();
23        }
24    }
}

Output
java -cp /tmp/I1SMBazJld/LoopShape
#####
# #
# #
#####
#
###
###
####
#####

=== Code Execution Successful ===

```