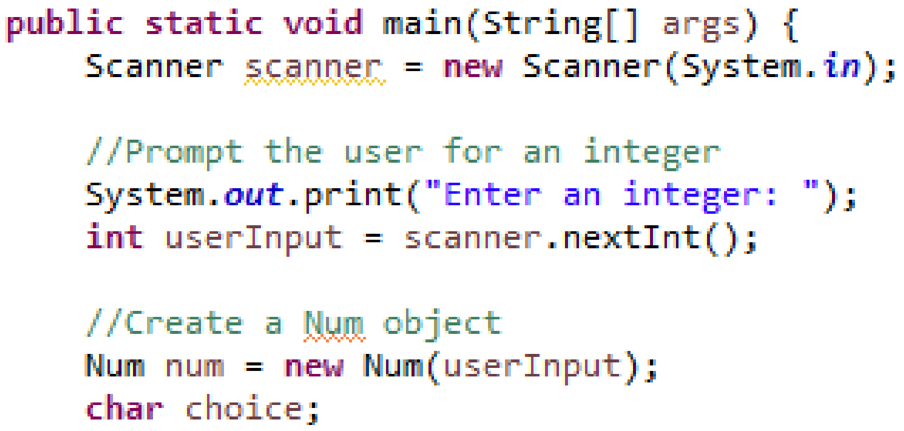
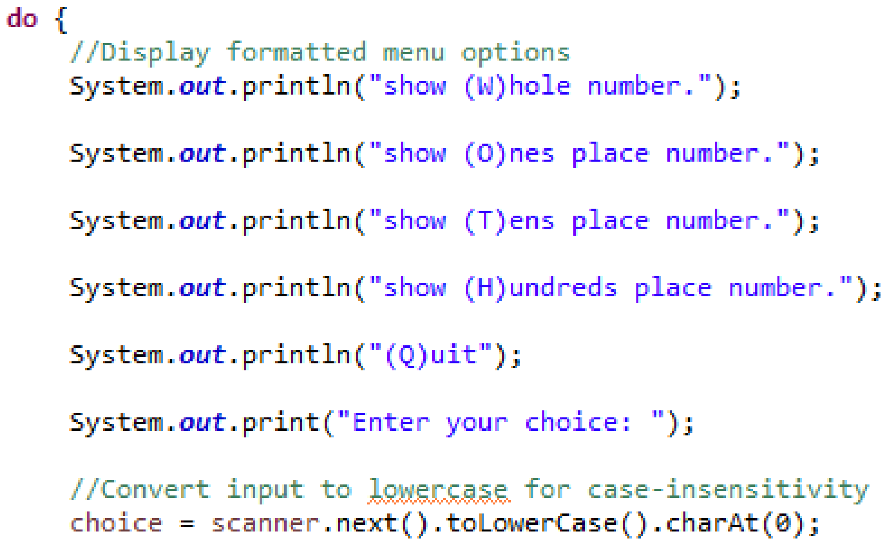
ReflectionLog DigitExtractor and Num



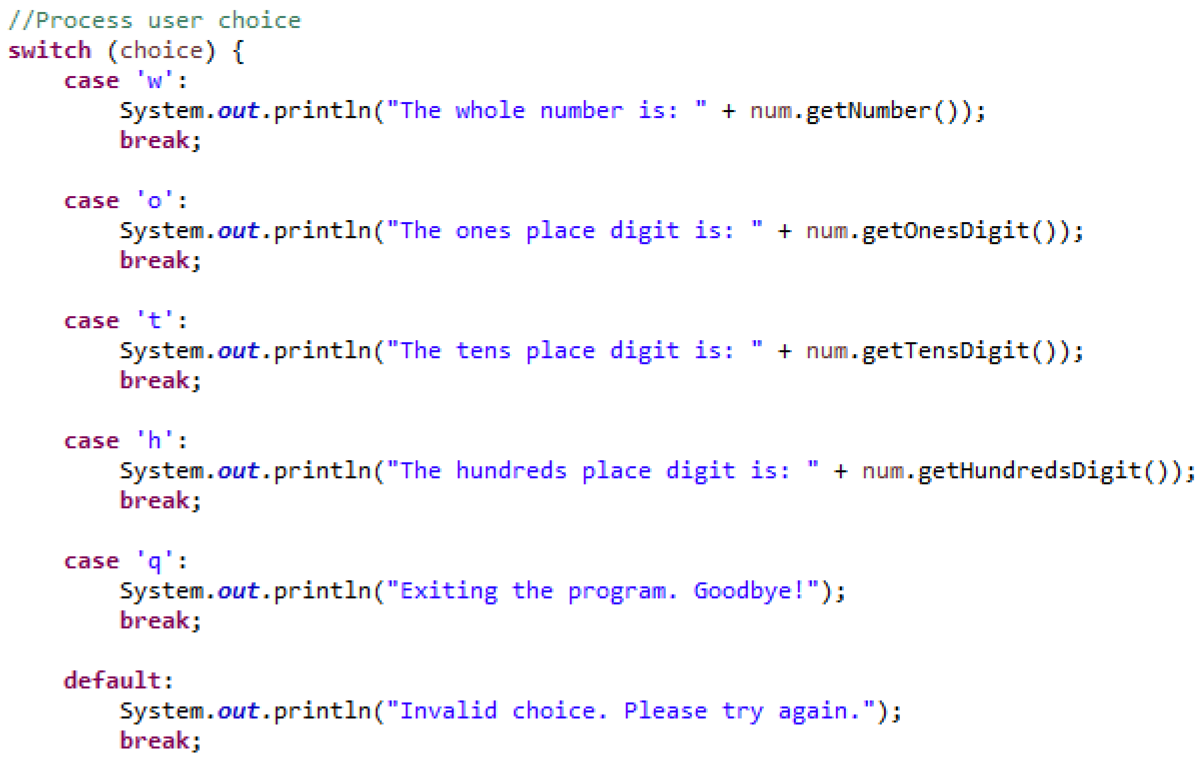
```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    //Prompt the user for an integer
    System.out.print("Enter an integer: ");
    int userInput = scanner.nextInt();

    //Create a Num object
    Num num = new Num(userInput);
    char choice;
```

This code snippet prompts the user to enter an integer, which is then stored in the variable userInput. A Num object is created with the integer as an argument to its constructor, allowing the program to work with the value. The choice variable is declared to store a character, likely for further user input, but it is not yet used in the provided code. The Scanner object is used to capture the user's input from the console.



```
do {
    //Display formatted menu options
    System.out.println("show (W)hole number.");

    System.out.println("show (O)nes place number.");

    System.out.println("show (T)ens place number.");

    System.out.println("show (H)undreds place number.");

    System.out.println("(Q)uit");

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

    //Convert input to lowercase for case-insensitivity
    choice = scanner.next().toLowerCase().charAt(0);
```

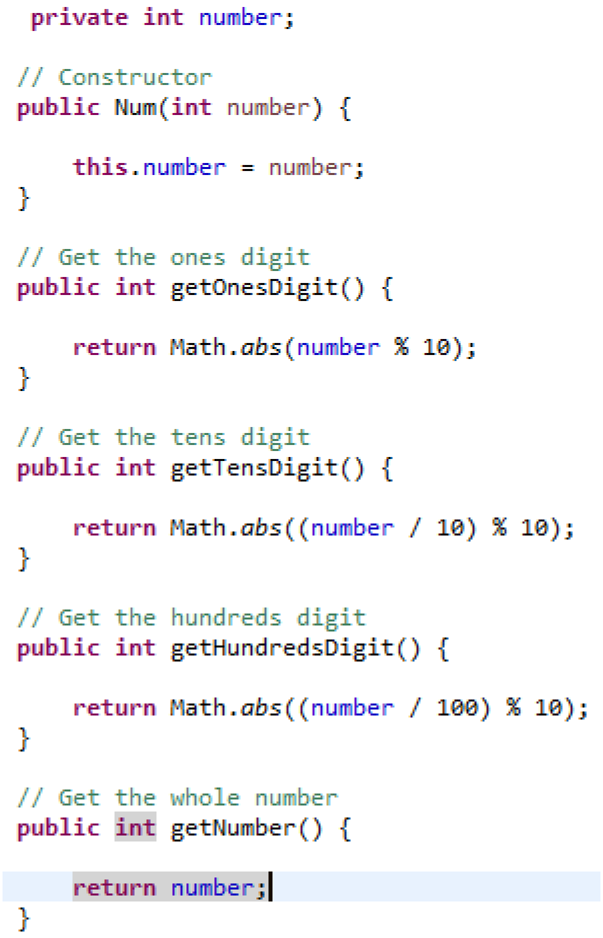
This code displays a menu of options to the user, allowing them to choose how they want to view the digits of the entered integer. The menu includes options to display the whole number, ones place, tens place, or hundreds place of the number, and also a quit option. The user’s choice is read as a string and then converted to lowercase for case-insensitivity, ensuring that both uppercase and lowercase input will be accepted. The choice variable stores the user's input, which is then used to determine the action to perform.



```
//Process user choice
switch (choice) {
    case 'w':
        System.out.println("The whole number is: " + num.getNumber());
        break;

    case 'o':
        System.out.println("The ones place digit is: " + num.getOnesDigit());
        break;

    case 't':
        System.out.println("The tens place digit is: " + num.getTensDigit());
        break;

    case 'h':
        System.out.println("The hundreds place digit is: " + num.getHundredsDigit());
        break;

    case 'q':
        System.out.println("Exiting the program. Goodbye!");
        break;

    default:
        System.out.println("Invalid choice. Please try again.");
        break;
```

This code processes the user's menu choice with a switch statement. Depending on the input (w, o, t, h, or q), it performs the corresponding action, such as displaying the whole number, ones place, tens place, or hundreds place digit. If the user selects q, the program exits with a goodbye message. An invalid choice results in a prompt to try again.

Num



```
 private int number;

// Constructor
public Num(int number) {

    this.number = number;
}

// Get the ones digit
public int getOnesDigit() {

    return Math.abs(number % 10);
}

// Get the tens digit
public int getTensDigit() {

    return Math.abs((number / 10) % 10);
}

// Get the hundreds digit
public int getHundredsDigit() {

    return Math.abs((number / 100) % 10);
}

// Get the whole number
public int getNumber() {

    return number;
}
```

The Num class is designed to store an integer and provides methods to retrieve specific digits of that number. The constructor initializes the number variable, which holds the user's input. The getOnesDigit() method returns the ones place digit, while getTensDigit() and getHundredsDigit() return the tens and hundreds place digits, respectively, by performing integer division and modulus operations. The getNumber() method simply returns the whole number. All methods use Math.abs() to ensure positive values are returned, regardless of the sign of the input number. These methods allow the program to access individual digits of the number as needed.