

```

import java.awt.Color;
import java.util.InputMismatchException;
import java.util.Random;
import java.util.Scanner;

public class IterationExercises {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        System.out.println("Pick an integer number between 1 and 20,
inclusive:");
        int countMatches = scan.nextInt();
        countMatches(countMatches);
        scan.nextLine();

        // Call count matches
        System.out.println("Enter a string and I will tell you now many
UPPERCASE letters are in it:\n ");
        String userStr = scan.nextLine();
        countMatches(userStr);

        // Call compute total
        System.out.println("Enter a positive integer and I will tell you the
sum from 1 to your number, inclusive: ");
        int userInt = -1;
        try {
            userInt = scan.nextInt();
        }
        catch (InputMismatchException inputEx) {
            System.out.println("An InputMismatchException occurred: " +
inputEx.getMessage());
        }
        computeTotal(userInt);
        scan.nextLine(); // Reads the newline

        // Call dart simulation
        dartSimulation();

        // Call credit card number checker
        System.out.println("Enter a credit card number and I will tell you if
it's valid: ");
        String userCC = scan.nextLine(); // nextLine() lets you enter a
string with spaces
        creditCardChecker(userCC); // Or use an int, you can decide

        // Call manipulating an image
        manipulatingAnImage();

        scan.close();
    }

    public static void countMatches(int guess) {
        Scanner scan = new Scanner(System.in);

        Random rand = new Random();
        int randomNumber = rand.nextInt(20) + 1;
        System.out.println("The game is to guess a number between 1 and 20.
Secret number picked.");
    }
}

```

```

int currentGuess = guess;

System.out.println("Your Guess: " + currentGuess);

if (currentGuess == randomNumber) {
    System.out.println("You guessed correctly!");
    System.out.println("Correct Number: " + randomNumber);
    return;
}
while (currentGuess != randomNumber) {

    System.out.println("You guessed incorrectly.");
    if (currentGuess > randomNumber) {
        System.out.println("Your guess is too high.\n");
    } else {
        System.out.println("Your guess is too low.\n");
    }
    System.out.print("Enter your next guess (1-20): ");

    if (scan.hasNextInt()) {
        currentGuess = scan.nextInt();
        scan.nextLine();
    } else {
        System.out.println("Invalid input. Please enter an integer.");
        scan.next();
        scan.nextLine();
        currentGuess = 0;
    }
}
System.out.println("Your Guess: " + currentGuess);
System.out.println("You guessed correctly!");
System.out.println("Correct Number: " + randomNumber);
}

/**
 * Count how many UPPERCASE letters are in a given string. Keep a counter, a
variable that is initialized to 0 and incremented whenever there is a match.
 * Display the final result back to the user.
 * @param str the given input String
 */
public static void countMatches(String str) {
    int count = 0;
    for (int i = 0; i < str.length(); i++) {
        char ch = str.charAt(i);
        if (Character.isUpperCase(ch)) {
            count++;
        }
    }
    System.out.println("The string has " + count + " uppercase letters");
}

/**
 * Compute the sum of the numbers from 1 to a given number. Keep a running
total: a variable to which you add each input value.
 * Display the final result back to the user.
 * @param number the given number
 */
public static void computeTotal(int number) {

```

```

        if (number < 1) {
            System.out.println("Only positive integers.");
            return;
        }
        int sum = 0;
        for (int i = 1; i <= number; i++) {
            sum = sum + i;
        }
        System.out.println("The sum from 1 to " + number + " is " + sum);
    }

    /**
     * Darts are thrown at random points onto the square corners (1, 1) and (-1,
-1).
     * If the dart lands inside the unit circle (that is, the circle with center
(0, 0) and radius 1), it is a hit.
     * Otherwise it is a miss. Run this simulation and use it to determine an
approximate value for PI.
     */
    public static void dartSimulation() {
        final int TOTAL_THROWS = 500000;
        int hits = 0;
        Random rand = new Random();

        for (int i = 0; i < TOTAL_THROWS; i++) {
            double x = (rand.nextDouble() * 2) - 1;
            double y = (rand.nextDouble() * 2) - 1;
            double distanceSquared = x * x + y * y;

            if (distanceSquared <= 1.0) {
                hits++;
            }
        }
        double piEstimate = 4.0 * hits / TOTAL_THROWS;

        System.out.println("Total Throws: " + TOTAL_THROWS);
        System.out.println("Hits in Circle: " + hits);
        System.out.printf("Estimated Value of PI: %.5f\n", piEstimate);
    }

    /**
     * Implement the algorithm to determine if a given credit card number is
valid.
     * @param ccNumber the given credit card number
     */
    public static void creditCardChecker(String ccNumber) {
        if (ccNumber.length() != 8 || !ccNumber.matches("\\d+")) {
            System.out.println("Error: Please enter a valid 8-digit
number.");
            return;
        }
        int totalSum = 0;
        int digit = 0;

        //second-to-last digit backwards
        for (int i = 6; i >= 0; i -= 2) {
            digit = Character.getNumericValue(ccNumber.charAt(i));
            int doubled = digit * 2;

```

```

        if (doubled > 9) {
            totalSum = totalSum + (doubled % 10) + (doubled / 10);
        } else {
            totalSum = totalSum + doubled;
        }
    }
    //last digit backwards
    for (int i = 7; i >= 0; i -= 2) {
        digit = Character.getNumericValue(ccNumber.charAt(i));
        totalSum += digit;
    }
    if (totalSum % 10 == 0) {
        System.out.println("The number is valid.");
    } else {
        System.out.println("The number is invalid.");
        int requiredLastDigit = 10 - (totalSum % 10);
        System.out.println("A valid number must have a total sum ending
in 0. The last digit of a valid number must be " + requiredLastDigit + ".");
    }
}

public static void manipulatingAnImage() {
    Picture picture = new Picture();
    picture.pick();

    if (picture.getWidth() == 0 || picture.getHeight() == 0) {
        System.out.println("No image was picked, cannot perform
manipulation.");
        return;
    }

    for (int x = 0; x < picture.getWidth(); x++) {
        for (int y = 0; y < picture.getHeight(); y++) {

            Color original = picture.getColorAt(x, y);

            int red = 255 - original.getRed();
            int green = 255 - original.getGreen();
            int blue = 255 - original.getBlue();

            Color negative = new Color(red, green, blue);

            picture.setColorAt(x, y, negative);
        }
    }
    System.out.println("Image converted to negative");
}
}

```