```
import java.util.InputMismatchException;
import java.util.Scanner;

public class FixedPointIteration {

    public static double g(double x) {
        return (Math.pow(x, 2) + 3) / 5;
    }

    public static void fixedPoint(double p0, double tol, int N) {
        try {
            double p = 0;

            for (int i = 1; i <= N; i++) {
                p = g(p0);
                System.out.printf("Iteration %2d: p = %.10f\n", i, p);

                if (Math.abs(p - p0) < tol) {
                    System.out.println("\nApproximate fixed point found: p = " + p);
                    return;
                }

                p0 = p;
            }

            System.out.println("\nMethod failed after " + N + " iterations.");
        } catch (Exception e) {
            System.err.println("Error during iteration: " + e.getMessage());
        }
    }

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

        try {
            System.out.print("Enter initial guess p0: ");
            double p0 = input.nextDouble();

            System.out.print("Enter tolerance: ");
            double tol = input.nextDouble();

            if (tol <= 0) {
                throw new IllegalArgumentException("Tolerance must be greater than 0.");
            }

            System.out.print("Enter maximum number of iterations: ");
            int N = input.nextInt();

            if (N <= 0) {
                throw new IllegalArgumentException("Number of iterations must be greater than
            }

            fixedPoint(p0, tol, N);
```

```
        } catch (InputMismatchException e) {
            System.err.println("Invalid input type. Please enter numbers only.");
        } catch (IllegalArgumentException e) {
            System.err.println("Input Error: " + e.getMessage());
        } catch (Exception e) {
            System.err.println("Unexpected Error: " + e.getMessage());
        } finally {
            input.close();
            System.out.println("Program finished.");
        }
    }
}
```