Exercise 1

Write a program that can serve as a simple calculator. This calculator keeps track of a single number (of type double) that is called result and that starts out as 0.0. Each cycle allows the user to repeatedly add, subtract, multiply, or divide by a second number. The result of one of these operations becomes the new value of result. The calculation ends when the user enters the letter R for “result” (either in upper- or lowercase). The user is allowed to do another calculation from the beginning as often as desired. The input format is shown in the following sample dialogue. If the user enters any operator symbol other than +, −, \*, or /, then an UnknownOperatorException is thrown and the user is asked to reenter that line of input. Defining the class UnknownOperatorException is part of this project.

**Sample output:**

Calculator is on.

result = 0.0

+5

result + 5.0 = 5.0

new result = 5.0

\* 2.2

result \* 2.2 = 11.0

updated result = 11.0

% 10

% is an unknown operation.

Reenter, your last line:

\* 0.1

result \* 0.1 = 1.1

updated result = 1.1

r

Final result = 1.1

Again? (y/n)

yes

result = 0.0

+10

result + 10.0 = 10.0

new result = 10.0

/2

result / 2.0 = 5.0

r

Final result = 5.0

Again? (y/n)

N

**End of Program**

import java.util.Scanner;

class Calculator{

    double number, result = 0.0;

    char operator;

    double getResult(double number, char operator){

*this*.number = number;

*this*.operator = operator;

*if* (operator == '+'){

            result += number;

            System.out.println("result + " + number + " = " + result);

*return* result;

        }  *else* *if* (operator == '\*'){

            result \*= number;

            System.out.println("result \* " + number + " = " + result);

*return* result;

        }  *else* *if* (operator == '-'){

            result -= number;

            System.out.println("result - " + number + " = " + result);

*return* result;

        }  *else* *if* (operator == '/'){

            result /= number;

            System.out.println("result / " + number + " = " + result);

*return* result;

        }  *else* {

*return* result;

        }

    }

}

public class CalculatorProgram{

    public static void main(String[] args) {

        Calculator calculator = *new* Calculator();

        Scanner input = *new* Scanner(System.in);

        double number;

        char operator;

        operator = input.next().charAt(0);

        number = input.nextDouble();

        System.out.println("new result = "+calculator.getResult(number, operator));

*while*(true){

            operator = input.next().charAt(0);

*if* (operator == 'r' || operator == 'R'){

                System.out.println("Final result = " + calculator.getResult(number, operator));

                System.out.println("Again? (y/n)");

                operator = input.next().charAt(0);

*if* (operator == 'y' || operator == 'y'){

                    calculator.result = 0.0;

                    System.out.println("result = " + calculator.result);

                    operator = input.next().charAt(0);

                    number = input.nextDouble();

                    System.out.println("new result = "+calculator.getResult(number, operator));

                }

*if* (operator == 'n' || operator == 'N'){

*break*;

                }

            } *else* {

                number = input.nextDouble();

*if* (operator == '+' || operator == '-' || operator == '\*' || operator == '/'){

                    System.out.println("updated result = "+calculator.getResult(number, operator));

                }  *else* {

                    System.out.println(operator + " is an unknown operation. \nReenter, your last line:");

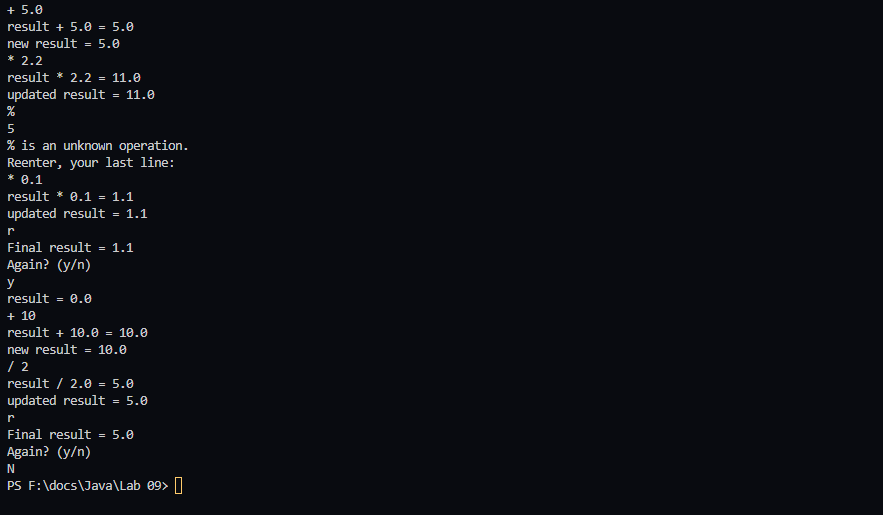
                }

            }

        }

    }

}



Exercise 2

Here is a snippet of code that inputs two integers and divides them:

Scanner scan = new Scanner(System.in);

int n1, n2;

double r;

n1 = scan.nextInt();

n2 = scan.nextInt();

r = ( double) n1 / n2;

Place this code into a try-catch block with multiple catches so that different error messages are printed if we attempt to divide by zero or if the user enters textual data instead of integers (java.util.InputMismatchException). If either of these conditions occurs, then the program should loop back and let the user enter new data.

import java.util.InputMismatchException;

import java.util.Scanner;

public class TryCatchProgram {

    public static void main(String[] args) {

        Scanner scan = *new* Scanner(System.in);

        int n1, n2;

        double r;

*try*{

            n1 = scan.nextInt();

            n2 = scan.nextInt();

            r = (double)(n1/n2);

        }

*catch*(ArithmeticException ae){

            System.out.println("Cannot be divided by 0");

*try*{

                n1 = scan.nextInt();

                n2 = scan.nextInt();

                r = (double)(n1/n2);

            }

*finally*{}

        }

*catch* (InputMismatchException ime){

            System.out.println("You can only input numbers.");

*try*{

                n1 = scan.nextInt();

                n2 = scan.nextInt();

                r = (double)(n1/n2);

            }

*finally*{}

        }

    }

}



Modify the previous exercise so that the snippet of code is placed inside a method. The method should be named ReturnRatio, read the input from the keyboard, and throw different exceptions if there is a division by zero or an input mismatch between text and an integer. Create your own exception class for the case of division by zero. Invoke ReturnRatio from your main method and catch the exceptions in main. The main method should invoke the ReturnRatio method again if any exception occurs.

import java.util.InputMismatchException;

import java.util.Scanner;

public class TryCatchExpend {

    static Scanner scan = *new* Scanner(System.in);

    static double returnRatio(int n1, int n2) {

        double r;

        r = (double) (n1 / n2);

*return* r;

    }

    public static void main(String[] args) {

        int n1, n2;

        double r;

*try* {

            n1 = scan.nextInt();

            n2 = scan.nextInt();

            System.out.println(returnRatio(n1 , n2));

        }

*// catch (Exception e){*

*// System.out.println(e);*

*// } // we can't do this if we do this will produce an error*

*catch* (ArithmeticException ae) {

            System.out.println("Cannot be divided by 0");

*try* {

                n1 = scan.nextInt();

                n2 = scan.nextInt();

                System.out.println(returnRatio(n1 , n2));

            } *finally* {

            }

        } *catch* (InputMismatchException ime) {

            System.out.println("You can only input numbers.");

*try* {

                n1 = scan.nextInt();

                n2 = scan.nextInt();

                System.out.println(returnRatio(n1 , n2));

            } *finally* {

            }

        } *catch* (Exception e) {

            System.out.println(e);

        }

    }

}

