Report: Programming Project 2

Nur Arafat

P100096845

# Code

## MathFunctions.Java

**package** edu.ohio.ise.ise6900.math;

**import** java.math.BigInteger;

**import** java.text.DecimalFormat;

**import** java.text.DecimalFormatSymbols;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**import** java.util.Locale;

/\*

\* ISE6900 Object Oriented Application in Industrial Engineering

\* Programming Project 02

\*

\* @author Nur Shomik Arafat

\* Date: 2017-01-20

\* @version 1.0

\*

\*/

**import** java.util.Scanner;

/\*\*

\* Class ...

\*

\* **@author** Nur Arafat

\*

\*/

**public** **class** MathFunctions {

**protected** **static** **final** **int** ***SIN*** = 1,

***COS*** = 2,

***TAN*** = 3,

***ASIN*** = 4,

***ACOS*** = 5,

***ATAN*** = 6,

***LN*** = 7,

***SQRT*** = 8,

***POW*** = 9,

***FACT*** = 10;

**private** **final** Integer[] optSet = {***SIN***, ***COS***, ***TAN***, ***ASIN***, ***ACOS***, ***ATAN***, ***LN***, ***SQRT***, ***POW***, ***FACT***};

**protected** **final** ArrayList<Integer> options = **new** ArrayList<Integer>(Arrays.*asList*(optSet));

**private** String errMsg, outputMsg;

**private** **boolean** doubleOutput;

**private** **double** outputValue=0;

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

MathFunctions mf = **new** MathFunctions();

mf.runConsoleApp();

}

**public** **void** runConsoleApp(){

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

String option = **null**, inputStr = **null**;

Integer choice = 0;

**double** input = 0;

**while** (**true**) {

**this**.promptUser();

option = scan.nextLine();

**if** (option != **null** && (option.equalsIgnoreCase("exit") || option.equalsIgnoreCase("quit"))) {

System.***out***.println("\nThank you for using MathFunctions console application!");

**break**;

}

**try** {

choice = Integer.*parseInt*(option);

**if**(**this**.options.contains(choice)){

System.***out***.print(" Enter input:");

inputStr = scan.nextLine();

**try**{

input = Double.*parseDouble*(inputStr);

} **catch** (NumberFormatException doubEx) {

**this**.errMsg = "You have entered \"" + inputStr + "\" for calculation !!\n !! Only numbers are allowed";

**continue**;

}

}

**else**{

**throw** **new** NumberFormatException();

}

} **catch** (NumberFormatException nfe) {

// nfe.printStackTrace();

**this**.errMsg = "Your selected option \"" + option + "\" is not in menu !!\n !! Please choose a number from the menu and enter";

**continue**;

}

**this**.doubleOutput = **true**;

**switch** (choice) {

**case** MathFunctions.***SIN***:

**this**.outputMsg = "sin(" + input + ") = ";

**this**.outputValue = Math.*sin*(Math.*toRadians*(input));

**break**;

**case** MathFunctions.***COS***:

**this**.outputMsg = "cos(" + input + ") = ";

**this**.outputValue = Math.*cos*(Math.*toRadians*(input));

**break**;

**case** MathFunctions.***TAN***:

**this**.outputMsg = "tan(" + input + ") = ";

**this**.outputValue = Math.*tan*(Math.*toRadians*(input));

**break**;

**case** MathFunctions.***ASIN***:

**if** (**this**.isProperFraction(input)) {

**this**.outputMsg = "asin(" + input + ") = (degrees) ";

**this**.outputValue = Math.*toDegrees*((Math.*asin*(input)));

} **else** {

**this**.errMsg = "Please enter a number between -1.0 and 1.0 for arcsin";

}

**break**;

**case** MathFunctions.***ACOS***:

**if** (**this**.isProperFraction(input)) {

**this**.outputMsg = "acos(" + input + ") = (degrees) ";

**this**.outputValue = Math.*toDegrees*((Math.*acos*(input)));

} **else** {

**this**.errMsg = "Please enter a number between -1.0 and 1.0 for arccos";

}

**break**;

**case** MathFunctions.***ATAN***:

**this**.outputMsg = "atan(" + input + ") = (degrees) ";

**this**.outputValue = Math.*toDegrees*((Math.*atan*(input)));

**break**;

**case** MathFunctions.***LN***:

**if** (**this**.isPositiveRealNumber(input)) {

**this**.outputMsg = "ln(" + input + ") = ";

**this**.outputValue = Math.*log*(input);

} **else** {

**this**.errMsg = "Please enter a positive number";

}

**break**;

**case** MathFunctions.***SQRT***:

**if** (**this**.isNonNegativeRealNumber(input)) {

**this**.outputMsg = "sqrt(" + input + ") = ";

**this**.outputValue = Math.*sqrt*(input);

} **else** {

**this**.errMsg = "Please enter a positive number or zero";

}

**break**;

**case** MathFunctions.***POW***:

System.***out***.print(" Enter power:");

**double** power = Double.*parseDouble*(scan.nextLine());

**this**.outputMsg = "pow(" + input + ", "+ power+") = ";

**this**.outputValue = Math.*pow*(input, power);

**break**;

**case** MathFunctions.***FACT***:

**if** (**this**.isNonNegativeInteger(input)) {

String out = "factorial(" + input + ") = ";

**if** (input < 13) {

out += *factorial*((**int**) input);

} **else** **if** (input < 21) {

out += *factorial*((**long**) input);

} **else** {

out += *factorial*(BigInteger.*valueOf*((**long**) input)).toString();

}

**this**.outputMsg = out;

**this**.doubleOutput = **false**;

} **else** {

**this**.errMsg = "Please enter a positive integer or zero";

}

**break**;

**default**:

**this**.errMsg = "Please choose a number from the menu";

**break**;

}

}

scan.close();

}

**public** **boolean** isProperFraction(**double** input) {

**if**(input>=-1 && input<=1){

**return** **true**;

}

**return** **false**;

}

**public** **boolean** isPositiveRealNumber(**double** input) {

**if**(input > 0){

**return** **true**;

}

**return** **false**;

}

**public** **boolean** isNonNegativeRealNumber(**double** input) {

**if**(input >= 0){

**return** **true**;

}

**return** **false**;

}

**public** **boolean** isNonNegativeInteger(**double** input) {

**if**(input < 0){

**return** **false**;

}

**if**(input>Math.*floor*(input)){

**return** **false**;

}

**return** **true**;

}

**public** **static** **int** factorial(**int** input) {

**if**(input==0 || input==1){

**return** 1;

}

**return** *factorial*(input-1)\*input;

}

**public** **static** **long** factorial(**long** input) {

**if**(input==0 || input==1){

**return** 1;

}

**return** *factorial*(input-1) \* input;

}

**public** **static** BigInteger factorial(BigInteger input){

**if**(input.equals(BigInteger.***ZERO***) || input.equals(BigInteger.***ONE***) ){

**return** BigInteger.***ONE***;

}

**return** (*factorial*(input.subtract(BigInteger.***ONE***))).multiply(input);

}

**private** **void** promptUser(){

System.***out***.println(

"\n ----------------------------------------------"

+ "\n (Enter 'exit' or 'quit' to exit the program)\n"

+ "\n Enter any of the following numbers:"

+ "\n ->sin : " + MathFunctions.***SIN***

+ "\n ->cos : " + MathFunctions.***COS***

+ "\n ->tan : " + MathFunctions.***TAN***

+ "\n ->asin : " + MathFunctions.***ASIN***

+ "\n ->acos : " + MathFunctions.***ACOS***

+ "\n ->atan : " + MathFunctions.***ATAN***

+ "\n ->ln : " + MathFunctions.***LN***

+ "\n ->sqrt : " + MathFunctions.***SQRT***

+ "\n ->pow : " + MathFunctions.***POW***

+ "\n ->factorial: " + MathFunctions.***FACT***

+ "\n");

**if**(**this**.errMsg != **null**){

System.***out***.flush();

System.***err***.flush();

**try** {

Thread.*sleep*(100);

} **catch** (InterruptedException e) {

e.printStackTrace();

}

System.***err***.println(" !! " + **this**.errMsg + " !!\n");

System.***out***.flush();

System.***err***.flush();

**this**.errMsg = **null**;

**try** {

Thread.*sleep*(100);

} **catch** (InterruptedException e) {

e.printStackTrace();

}

}

**if**(**this**.outputMsg != **null**){

**if**(**this**.doubleOutput){

DecimalFormat df = **new** DecimalFormat("0", DecimalFormatSymbols.*getInstance*(Locale.***ENGLISH***));

df.setMaximumFractionDigits(340);//DecimalFormat.DOUBLE\_FRACTION\_DIGITS

String outv = df.format(**this**.outputValue);

System.***out***.println(" LAST RESULT:\n ##> " + outputMsg + outv + "\n");

}

**else**{

System.***out***.println(" LAST RESULT:\n ##> " + outputMsg + "\n");

}

}

System.***out***.print(" $$ select ->");

}

/\* (non-Javadoc)

\* @see java.lang.Object#hashCode()

\*/

@Override

**public** **int** hashCode() {

**final** **int** prime = 31;

**int** result = 1;

result = prime \* result + Arrays.*hashCode*(optSet);

result = prime \* result + ((options == **null**) ? 0 : options.hashCode());

**return** result;

}

/\* (non-Javadoc)

\* @see java.lang.Object#equals(java.lang.Object)

\*/

@Override

**public** **boolean** equals(Object obj) {

**if** (**this** == obj)

**return** **true**;

**if** (obj == **null**)

**return** **false**;

**if** (!(obj **instanceof** MathFunctions))

**return** **false**;

MathFunctions other = (MathFunctions) obj;

**if** (!Arrays.*equals*(optSet, other.optSet))

**return** **false**;

**if** (options == **null**) {

**if** (other.options != **null**)

**return** **false**;

} **else** **if** (!options.equals(other.options))

**return** **false**;

**return** **true**;

}

/\* (non-Javadoc)

\* @see java.lang.Object#toString()

\*/

@Override

**public** String toString() {

**return** "MathFunctions [optSet=" + Arrays.*toString*(optSet) + ", options=" + options + "]";

}

}

# Screen-shots











