Concept Guide: Methods – code modularity and reusability

Course: CS 200

Introduction to Java Programming, Comprehensive Version, 11th ed. Text

Chapter: 6 Methods

|  |  |  |
| --- | --- | --- |
| Concept/Topic: | Text Notes: | Lecture Notes: |
| Methodsdefined |  |  |
| Benefit to using Methods |  |  |
| Key Components to a method |  |  |
| modifier |  |  |
| return types |  |  |
| MethodIdentifier |  |  |
| Parameters |  |  |
| Algorithm for creating and using method |  |  |
| MethodPlacement |  |  |
| Variable Scope |  |  |
| Pass By ValueVs.Pass By Reference |  |  |
| 3ThingsTo Remember: |  |  |
| OverloadingMethods(see text only) |  |  |

Lab Time! Copy and Paste into jGrasp and finish

/\*----------------------------------------------------------------------------------------

Finish the Super Calculator

1. Allow user to run/repeat this application as often as they want.

Hint: need a repetition structure in main method

2. Complete the getValue method below, then replace the appropriate current code

with a call to the geValue method instead

Hint: Can you copy & paste anything to get this one done quick & easy?

3. Create the methods:

“uFaD” Output a user friendly application description. What does this program do?

"3 - Easy Subtract: valA - valB\n"

+"4 - Easy multiply: valA \* valB\n"

+"5 - Area of a rectangle: length \* width\n");

and call them in the main method per the menu method listing

Hint: Can you re-use any of these methods? i.e. one method calls another Try it!

4. Add at least 5, but no more than 7 menu option additional features to your

very own personal Super Calculator! Create the methods & modify the main method as needed.

Go Ahead - get creative & have some fun! (but don't borrow ideas from your neighbor)

Note: These methods must be flowcharted FIRST & approved before they can be implemented.

Note: All output results should be to the "Run I/O" frame, but user prompts should be dialog boxes

-----------------------------------------------------------------------------------------\*/

import java.util.Scanner;

import javax.swing.JOptionPane;

public class Asst07SuperCalculator

{

public static void main(String args[ ])

 {

 Scanner keyboard = new Scanner(System.in);

 double value;

 int choice;

 String inputMain;

 double val1;

 double val2;

 double results;

 uFaD(); //User Friendly Application Description

 choice = menu();

 switch(choice)

 {

 case 1:

 inputMain = JOptionPane.showInputDialog("Enter Value 1:");

 val1=Double.parseDouble(inputMain);

 inputMain = JOptionPane.showInputDialog("Enter Value 2:");

 val2=Double.parseDouble(inputMain);

 results = easyAdd(val1,val2);

 System.out.println(val1 + " + " + val2 + " = " + results);

 break;

 case 2:

 bigAdd();

 break;

 case 3:

 JOptionPane.showMessageDialog(null, "Method under construction!");

 break;

 case 4:

 JOptionPane.showMessageDialog(null, "Method under construction!");

 break;

 case 5:

 JOptionPane.showMessageDialog(null, "Method under construction!");

 break;

 }//end of switch

 } //closing main method

Public static void uFaD()

{

}

public static int menu()

{

 String input;

 int option;

 boolean inRange = false;

 int minInput=1;

 int maxInput=5;

 do

 {

 input = JOptionPane.showInputDialog("Choose a task for the Super Calculator:\n"

 +"1 - Easy Add: valA + valB\n"

 +"2 - Big Add: n values summed\n"

 +"3 - Easy Subtract: valA - valB\n"

 +"4 - Easy multiply: valA \* valB\n"

 +"5 - Area of a rectangle: length \* width\n");

 option = Integer.parseInt(input);

 inRange = rangeCheck(option,minInput,maxInput);

 if (inRange != true)

 JOptionPane.showMessageDialog(null, "Error! Please enter a value between "+ minInput + " - " + maxInput);

 }while(inRange != true);

 return option;

}//end of menu

public static boolean rangeCheck(int x, int min, int max)

{

 boolean valid = false;

 if(x>=min && x<=max)

 valid = true;

 return valid;

}//end of rangeCheck

public static double easyAdd(double a, double b)

{

double resultsEA = a+b;

return resultsEA;

}//end of easyAdd

public static void bigAdd()

{

double resultsBA=0;

String inputBA;

double x;

inputBA = JOptionPane.showInputDialog("Please enter 1st value: ");

 x =Double.parseDouble(inputBA);

 System.out.print(x);

 resultsBA = easyAdd(resultsBA,x);

do

{

 inputBA = JOptionPane.showInputDialog("Please enter another value: ");

 x =Double.parseDouble(inputBA);

 System.out.print(" + " + x);

 resultsBA = easyAdd(resultsBA,x);

 inputBA = JOptionPane.showInputDialog("Do you have another value? Y or N: ");

}while(inputBA.charAt(0)=='y' || inputBA.charAt(0) =='Y');

 System.out.print(" = " + resultsBA);

}//end of big add

public static double getValue()

{

//This method should prompt the user for a value using a dialog box

//Then it should convert the string returned by the dialog box to a double PDT

//Finally, it should return the double value to the calling method.

//Replace all possible user prompts in this code for double data values with a call to this method instead

return 0.0;

}

} //closing class header