**Gas Flow Rate, Productivity Index, Skin Factor and Flow Rate Prediction:**

The Java Programming Language and its platform was invented by James Gosling at Oracle. It has been widely recognized as one of the most versatile programming languages in the world. Java Applications are written in group of related instructions called by various names. Sometimes, they are called “Classes” and other times, they are known as “Interfaces”.

Languages like Java and Python are called higher level languages. As useful as they are in passing instructions and workflow across to the computer, the machine does not really understand them. In order then for the series of instructions written in these programming languages to be understood by the computer (so that one could run it as an application), it must be converted to computer’s own language (machine language). What helps in this conversion is known as Compiler. The Java compiler converts Java language to the machine language so that the programs can execute.

**Java Computation Package**

Java platform comes with several classes out of the box. There is no need to download any special libraries (series of java programs written by other programmers or professionals to do certain common things) to perform simple to fairly complex Mathematical and Engineering Calculations. Java computes very fast and integrate seamlessly with external engineering toolboxes such as MATLAB for example.

The computations for logarithms, exponentiation and other operations were performed in this project using the package “java.math.*\*”.*

To make the application development easier, a scripting language – groovy targeting the JVM has been used. This has made the development to be less verbose and focus on getting things done rather than worrying on the language. Groovy however, uses java libraries and compiles back to java classes which are still executed by the Java Virtual Machine

Graphical User Interfaces

The common design pattern in application is to develop an interface that the user will interact with. This interface is usually graphical and include set of controls such as text field for getting user input, check box for getting user’s choices, progress bar for monitoring the rate of task execution, etc. Having this at heart, java comes out of the box with Graphical User Interface components that user can interact with to make effective use of the application.

The package used for GUI in this project is the “java.swing.\*”.

**Project Implementation**

The Java Application was implemented with four traits and one main java class viz:

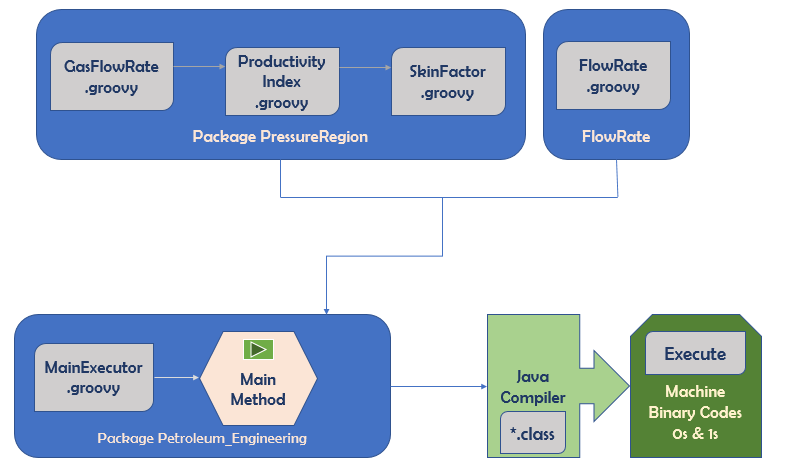
Traits - GasFlowRate, ProductivityIndex, SkinFactor, FlowRate

Class – ExecHelper, MainExecutor.

The MainExecutor class contains the main method which is from where the application flow is to start. When a class implements an interface, it automatically inherits all the trait’s method, hence all the methods in the four traits are accessible to the Main Executor class using the “implements” keyword. Hence, the easy flow of behavior invocation.

**Application Flow**

The Sequence diagram below shows the program flow of execution:



All computations are coordinated from the MainExecutor class which glues the Graphical User Interface together with the implementation logic. The interface methods are invoked when the value is to be obtained from a computation.

The following values were catered for:

* Gas Flow Rate at General Point Region
* Gas Flow Rate at High Point Region
* Gas Flow Rate at Low Point Region
* Alternative Computation for Gas Flow Rate at General Point Region
* Alternative Computation for Gas Flow Rate at Low Point Region
* Productivity Index at General Point Region
* Productivity Index at High Point Region
* Productivity Index at Low Point Region
* Alternative Computation for Productivity Index at General Point Region
* Alternative Computation for Productivity Index Low Point Region
* Universal Skin Factor
* Skin Factor at General Point Region
* Skin Factor at High Point Region
* Skin Factor at Low Point Region
* Skin Factor at Normal Point Region
* Normal/Universal Flow Rate
* Present Flow Rate
* Future Flow Rate

The program was developed and executed on Windows 10 HP computer with 4gig RAM and 232GB hard drive. It runs fairly fast and effectively as seen from the snapshot below.

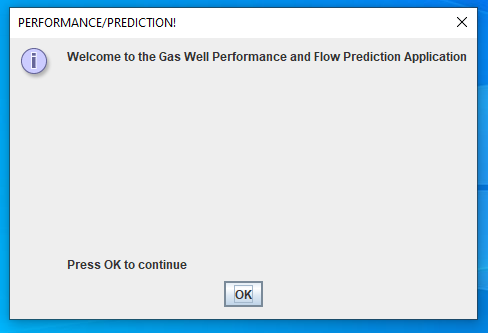


Fig 1.0: Application’s first landing page

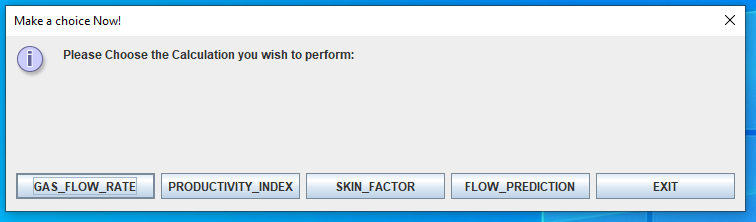


Fig 1.2: Getting User’s Choice from the main categories

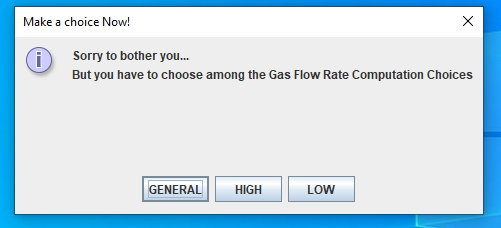


Fig 1.3: Getting the User’s Choice from the Gas Flow Rate Category

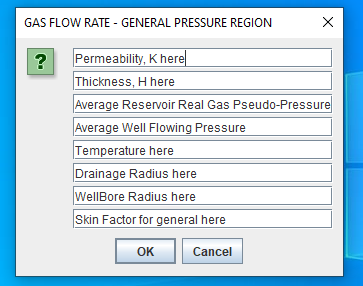


Fig 1.4: Accepting gas parameter inputs from the User.

**Simulated Gas Well Data and Computation**

**General additional well information:**

k = 65 md

h = 15’

T= 600°R

re = 1000’

rw = 0.25’

s = -0.4, 0, 0.4

**For High:**

Using pr of 4000 and pwf of 3200

Average is 3600psi so we use the well values at p = 3600 from the table drawn above.

**For Low:**

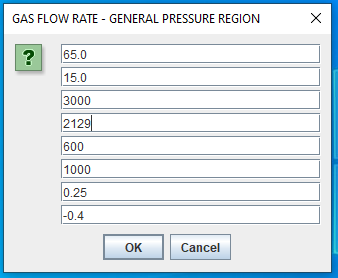
Using pr of 2000 and pwf of 1058

Average is 1649 psi so we use the well values at p = 1649 from the table

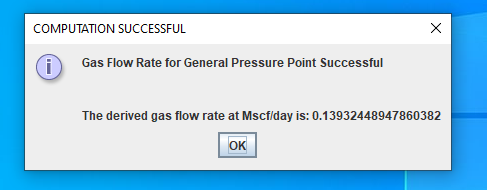
ug is 0.017 and z is 0.791 from interpolation

**Solution:**

* **GAS FLOW RATE - GENERAL PRESSURE REGION**

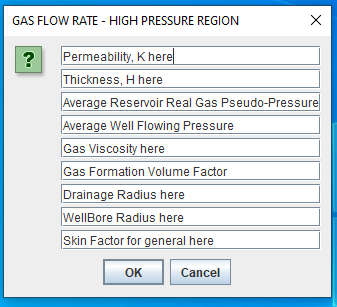


**Fig. 1.5: Gas parameter input for General Pressure Region**

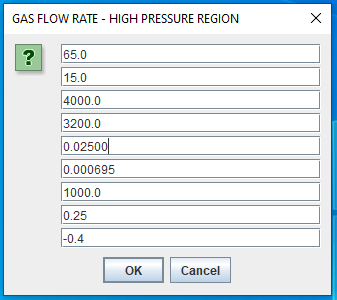


**Fig. 1.6: Computed Gas Flow Rate value for General Pressure Region**

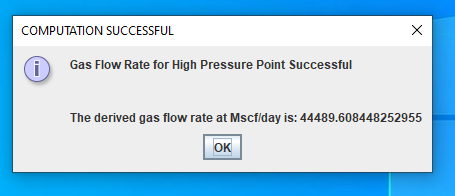
* **GAS FLOW RATE – HIGH PRESSURE REGION**



**Fig. 1.7: Gas parameter input layout for High Pressure Region**

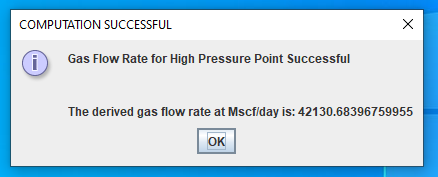


**Fig. 1.8: Gas parameter input values for High Pressure Region**



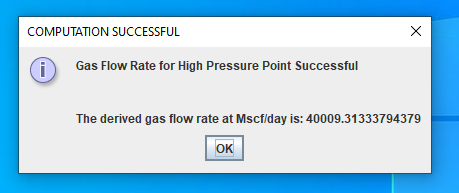
**Fig. 1.9: Computed Gas Flow Rate value for High Pressure Region at**

**(Skin Factor = -0.4)**



**Fig. 2.0: Computed Gas Flow Rate value for High Pressure Region at**

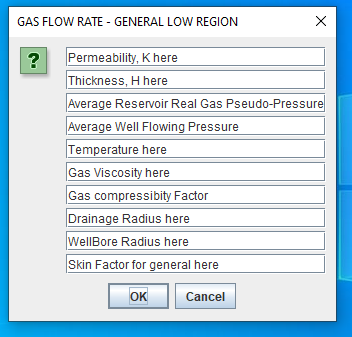
**(Skin Factor = 0)**



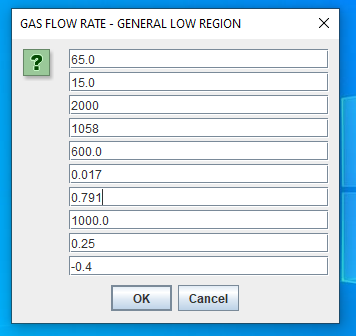
**Fig. 2.0: Computed Gas Flow Rate value for High Pressure Region at**

**(Skin Factor = 0.4)**

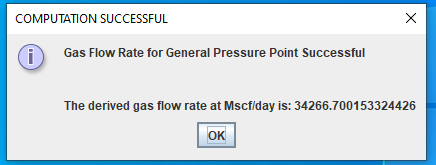
* **GAS FLOW RATE – LOW PRESSURE REGION**



**Fig. 2.1: Gas parameter input layout for Low Pressure Region**

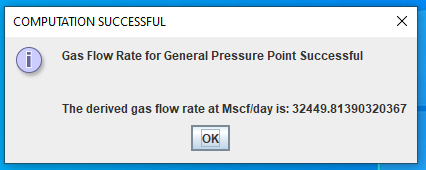


**Fig. 2.2: Gas parameter input values for Low Pressure Region**



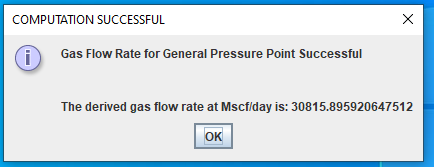
**Fig. 2.3: Computed Gas Flow Rate value for Low Pressure Region at**

**(Skin Factor = -0.4)**



**Fig. 2.4: Computed Gas Flow Rate value for High Pressure Region at**

**(Skin Factor = 0)**



**Fig. 2.5: Computed Gas Flow Rate value for High Pressure Region at**

**(Skin Factor = 0.4)**

**TABULATED RESULTS: showing Flow Rate results at various Skin Factor values**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Skin Factor (-0.4)** | **Skin Factor (0)** | **Skin Factor (0.4 )** |
| **Flow Rate (General)** | **0.13932448947860382** | **------------** | **------------** |
| **Flow Rate (High)** | **44489.608448252955** | **42130.68396759955** | **40009.31333794379** |
| **Flow Rate (Low)** | **34266.700153324426** | **32449.81390320367** | **30815.895920647512** |

**Glossary**

**Source Codes**

**Package Gas\_Engineering**

**MainExecutor.groovy**

**package Gas\_Engineering**

**import static javax.swing.JOptionPane.\***

**class MainExecutor {**

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

**//Execution starts here...**

**showMessageDialog(null, '''Welcome to the Gas Well Performance and Flow Prediction Application**

**\n\n\n\n\n\n\n\n\n\n\nPress OK to continue''', "PERFORMANCE/PREDICTION!", INFORMATION\_MESSAGE)**

**String[] choices = ["GAS\_FLOW\_RATE", "PRODUCTIVITY\_INDEX", "SKIN\_FACTOR", "FLOW\_PREDICTION", "EXIT"]**

**int chosen = showOptionDialog(null, "Please Choose the Calculation you wish to perform:\n\n\n\n\n\n\n",**

**"Make a choice Now!", DEFAULT\_OPTION, INFORMATION\_MESSAGE, null, choices, choices[0])**

**//Instantiate the ExecHelper class:**

**ExecHelper execHelper = new ExecHelper()**

**if (chosen == 0) {**

**//compute Gas Flow Rate**

**execHelper.gasFlowRate()**

**} else if (chosen == 1) {**

**//compute Productivity Index**

**execHelper.prodIndex()**

**} else if (chosen == 2) {**

**//compute Skin Factor**

**execHelper.skinFactor()**

**} else if (chosen == 3) {**

**//compute Flow Prediction**

**execHelper.flowPrediction()**

**} else {**

**showMessageDialog(null, "Application about to exit. Thanks for using the Application", "GOODBYE!", WARNING\_MESSAGE)**

**System.exit(0)**

**}**

**}**

**}**

**Package Gas\_Engineering:**

**ExecHelper.groovy**

**package Gas\_Engineering**

**import PressureRegion.\***

**import FlowRate.\***

**import javax.swing.JTextField**

**import static javax.swing.JOptionPane.\***

**/\*import static javax.swing.JOptionPane.DEFAULT\_OPTION**

**import static javax.swing.JOptionPane.INFORMATION\_MESSAGE**

**import static javax.swing.JOptionPane.OK\_CANCEL\_OPTION**

**import static javax.swing.JOptionPane.OK\_OPTION**

**import static javax.swing.JOptionPane.WARNING\_MESSAGE**

**import static javax.swing.JOptionPane.YES\_NO\_OPTION**

**import static javax.swing.JOptionPane.showConfirmDialog**

**import static javax.swing.JOptionPane.showMessageDialog**

**import static javax.swing.JOptionPane.showOptionDialog\*/**

**class ExecHelper implements GasFlowRate, ProductivityIndex, SkinFactor, FlowRate{**

**def gasFlowRate = {**

**showMessageDialog(null, "You have Chosen the Gas Flow Rate Calculation!", "GAS FLOW RATE SELECTED", INFORMATION\_MESSAGE)**

**String[] GFRchoices = ["GENERAL", "HIGH", "LOW"]**

**int GFRchosen = showOptionDialog(null, "Sorry to bother you...\nBut you have to choose among the Gas Flow Rate Computation Choices\n\n\n\n\n\n",**

**"Make a choice Now!", DEFAULT\_OPTION, INFORMATION\_MESSAGE, null, GFRchoices, GFRchoices[0])**

**switch (GFRchosen) {**

**case 0://general perm, thick, avrPseudo, avrWellFactor, temp, drainRad, wellBoreRad, skinFac**

**//Prompt the user to enter appropriate parameters:**

**JTextField perm = new JTextField("Permeability, K here")**

**JTextField thick = new JTextField("Thickness, H here")**

**JTextField avrPseudo = new JTextField("Average Reservoir Real Gas Pseudo-Pressure")**

**JTextField avrWellFactor = new JTextField("Average Well Flowing Pressure")**

**JTextField temp = new JTextField("Temperature here")**

**JTextField drainRad = new JTextField("Drainage Radius here")**

**JTextField wellBoreRad = new JTextField("WellBore Radius here")**

**JTextField skinFac = new JTextField("Skin Factor for general here")**

**Object[] options = [perm, thick, avrPseudo, avrWellFactor, temp, drainRad, wellBoreRad, skinFac]**

**showMessageDialog(null, "INSTRUCTION:\nSupply all integer whole number values as decimal i.e 2 as 2.0 in the parameters")**

**int computeCommand = showConfirmDialog(null, options, "GAS FLOW RATE - GENERAL PRESSURE REGION", OK\_CANCEL\_OPTION)**

**showMessageDialog(null, "The supplied gas parameters inputed previously will be used to compute \nthe Gas Flow Rate at General Pressure Point\n\n")**

**if (computeCommand == OK\_OPTION) {**

**//Begin the computation:**

**try {**

**def GFRgeneral = this.GFRgeneral(perm.text as Double, thick.text as Double, avrPseudo.text as Double, avrWellFactor.text as Double, temp.text as Double, drainRad.text as Double, wellBoreRad.text as Double, skinFac.text as Double)**

**showMessageDialog(null, "Gas Flow Rate for General Pressure Point Successful \n\n\nThe derived gas flow rate at Mscf/day is: ${GFRgeneral}", "COMPUTATION SUCCESSFUL", INFORMATION\_MESSAGE)**

**int yes\_no = showConfirmDialog(null, "Do you want to continue to the alternative computation of Gas Flow Rate at General Pressure Point?\n\n", "HEY!, WANNA CONTINUE?", YES\_NO\_OPTION)**

**if (yes\_no == YES\_OPTION) {**

**JTextField genProdIndex = new JTextField("Production Index for General Pressure Point ")**

**JTextField avrPseudoAlt = new JTextField("Average Reservoir Real Gas Pseudo-Pressure")**

**JTextField avrWellFactorAlt = new JTextField("Average Well Flowing Pressure")**

**Object[] optionsAlt = [genProdIndex, avrPseudoAlt, avrWellFactorAlt]**

**showMessageDialog(null, "Welcome to the alternative computation of Gas Flow Rate at General Pressure Point.")**

**showMessageDialog(null, "INSTRUCTION:\nSupply all integer whole number values as decimal i.e 2 as 2.0 in the parameters")**

**int chooseAlt = showConfirmDialog(null, optionsAlt, "GAS FLOW RATE ALTERNATIVE - GENERAL PRESSURE REGION", YES\_NO\_OPTION)**

**showMessageDialog(null, "\nThe supplied gas parameters obtained previously will be used to compute\nthe flow rate at general pressure point")**

**if (chooseAlt == YES\_OPTION) {**

**def GFRgeneralAlt = this.GFRgeneralAlt(genProdIndex.text as Double, avrPseudoAlt.text as Double, avrWellFactorAlt.text as Double)**

**showMessageDialog(null, "Gas Flow Rate for General Pressure Point Successful \n\n\nThe derived gas flow rate at Mscf/day is: ${GFRgeneralAlt}", "COMPUTATION SUCCESSFUL", INFORMATION\_MESSAGE)**

**showMessageDialog(null, "Computation Performed!,\n ...Exiting Now..Thanks for using the application", "APP TO EXIT", INFORMATION\_MESSAGE)**

**System.exit(0)**

**} else {**

**showMessageDialog(null, "Apparently, you dont want to continue...." +**

**"\n\n Exiting the System Now..Thanks for using the Application", "APP TO EXIT!", INFORMATION\_MESSAGE)**

**System.exit(0)**

**}**

**} else {**

**showMessageDialog(null, "Apparently, you dont want to continue...." +**

**"\n\n Exiting the System Now..Thanks for using the Application", "APP TO EXIT!", INFORMATION\_MESSAGE)**

**System.exit(0)**

**}**

**} catch (Exception ex) {**

**ex.printStackTrace()**

**showMessageDialog(null, "Error occurred in Computation!", "Flow Rate Computation Failure", WARNING\_MESSAGE)**

**//showConfirmDialog(null, "Do you want to continue to the alternative computation of General Gas Flow?")**

**}**

**} else {**

**showMessageDialog(null, "Application Cancelled...Goodbye")**

**System.exit(0)**

**}**

**break**

**case 1://high perm, thick, avrPressure, avrPressureWellFactor, gasViscosity, formVolFac, drainRad, wellBoreRad, skinFac**

**//Prompt the user to enter appropriate parameters:**

**JTextField perm = new JTextField("Permeability, K here")**

**JTextField thick = new JTextField("Thickness, H here")**

**JTextField avrPressure = new JTextField("Average Reservoir Real Gas Pseudo-Pressure")**

**JTextField avrPressureWellFactor = new JTextField("Average Well Flowing Pressure")**

**JTextField gasViscosity = new JTextField("Gas Viscosity here")**

**JTextField formVolFac = new JTextField("Gas Formation Volume Factor ")**

**JTextField drainRad = new JTextField("Drainage Radius here")**

**JTextField wellBoreRad = new JTextField("WellBore Radius here")**

**JTextField skinFac = new JTextField("Skin Factor for general here")**

**Object[] options = [perm, thick, avrPressure, avrPressureWellFactor, gasViscosity, formVolFac, drainRad, wellBoreRad, skinFac]**

**showMessageDialog(null, "INSTRUCTION: \n\n Supply all integer values as decimal i.e 2 as 2.0")**

**int computeCommand = showConfirmDialog(null, options, "GAS FLOW RATE - HIGH PRESSURE REGION", OK\_CANCEL\_OPTION)**

**showMessageDialog(null, "The supplied gas parameters above will be used to compute \nthe Gas Flow Rate at High Pressure Point")**

**if (computeCommand == OK\_OPTION) {**

**//Begin the computation:**

**try {**

**def GFRhigh = this.GFRhigh(perm.text as Double, thick.text as Double, avrPressure.text as Double, avrPressureWellFactor.text as Double, gasViscosity.text as Double,**

**formVolFac.text as Double, drainRad.text as Double, wellBoreRad.text as Double, skinFac.text as Double)**

**showMessageDialog(null, "Gas Flow Rate for High Pressure Point Successful \n\n\nThe derived gas flow rate at Mscf/day is: ${GFRhigh}",**

**"COMPUTATION SUCCESSFUL", INFORMATION\_MESSAGE)**

**int yes\_no = showConfirmDialog(null, "Do you want to continue to the alternative computation of Gas Flow Rate at High Pressure Point?\n\n", "HEY!, WANNA CONTINUE?", YES\_OPTION)**

**if (yes\_no) {**

**showMessageDialog(null, "Sorry, but there is no alternative Gas Flow Rate computation for High Pressure Point" +**

**"\n ..System about to exit", "NO ALTERNATIVE!", WARNING\_MESSAGE)**

**System.exit(0)**

**}**

**} catch (Exception ex) {**

**ex.printStackTrace()**

**showMessageDialog(null, "Error occurred in Computation!", "Flow Rate Computation Failure", WARNING\_MESSAGE)**

**System.exit(0)**

**}**

**} else {**

**showMessageDialog(null, "Application Cancelled...Goodbye")**

**System.exit(0)**

**}**

**break**

**case 2://low perm, thick, avrPressure, avrPressureWellFactor, temp, gasViscosity, gasCompressFac, drainRad, wellBoreRad, skinFac**

**//Prompt the user to enter appropriate parameters:**

**JTextField perm = new JTextField("Permeability, K here")**

**JTextField thick = new JTextField("Thickness, H here")**

**JTextField avrPressure = new JTextField("Average Reservoir Real Gas Pseudo-Pressure")**

**JTextField avrPressureWellFactor = new JTextField("Average Well Flowing Pressure")**

**JTextField temp = new JTextField("Temperature here")**

**JTextField gasViscosity = new JTextField("Gas Viscosity here")**

**JTextField gasCompressFac = new JTextField("Gas compressibity Factor")**

**JTextField drainRad = new JTextField("Drainage Radius here")**

**JTextField wellBoreRad = new JTextField("WellBore Radius here")**

**JTextField skinFac = new JTextField("Skin Factor for general here")**

**Object[] options = [perm, thick, avrPressure, avrPressureWellFactor, temp, gasViscosity, gasCompressFac, drainRad, wellBoreRad, skinFac]**

**showMessageDialog( null, "INSTRUCTION: \nSupply all integer values as decimal i.e 2 as 2.0")**

**int computeCommand = showConfirmDialog(null, options, "GAS FLOW RATE - GENERAL LOW REGION", OK\_CANCEL\_OPTION)**

**showMessageDialog(null, "The supplied gas parameters above obtained previously will be used to compute \nthe flow rate at low pressure point")**

**if (computeCommand == OK\_OPTION) {**

**//Begin the computation:**

**try {**

**def GFRlow = this.GFRlow(perm.text as Double, thick.text as Double, avrPressure.text as Double, avrPressureWellFactor.text as Double, temp.text as Double,**

**gasViscosity.text as Double, gasCompressFac.text as Double, drainRad.text as Double, wellBoreRad.text as Double, skinFac.text as Double)**

**showMessageDialog(null, "Gas Flow Rate for General Pressure Point Successful \n\n\nThe derived gas flow rate at Mscf/day is: ${GFRlow}",**

**"COMPUTATION SUCCESSFUL", INFORMATION\_MESSAGE)**

**int yes\_no = showConfirmDialog(null, "Do you want to continue to the alternative computation of Gas Flow Rate at General Pressure Point?\n\n", "HEY!, WANNA CONTINUE?", YES\_NO\_OPTION)**

**if (yes\_no == YES\_OPTION) {//lowProIndex, avrPressure, avrPressureWellFactor**

**JTextField lowProdIndex = new JTextField("Production Index for General Pressure Point ")**

**JTextField avrPressureAlt = new JTextField("Average Reservoir Real Gas Pseudo-Pressure")**

**JTextField avrPressureWellFactorAlt = new JTextField("Average Well Flowing Pressure")**

**Object[] optionsAlt = [lowProdIndex, avrPressureAlt, avrPressureWellFactorAlt]**

**showMessageDialog(null, "Welcome to the alternative computation of Gas Flow Rate at Low Pressure Point.")**

**int chooseAlt = showConfirmDialog(null, optionsAlt, "GAS FLOW RATE ALTERNATIVE - LOW PRESSURE REGION", YES\_NO\_OPTION)**

**showMessageDialog(null, "\n The supplied gas parameters previously obtained will be used to compute\nthe Flow Rate at Low Pressure Point")**

**if (chooseAlt == YES\_OPTION) {**

**def GFRlowAlt = this.GFRlowAlt(lowProdIndex.text as Double, avrPressureAlt.text as Double, avrPressureWellFactorAlt.text as Double)**

**showMessageDialog(null, "Gas Flow Rate for Low Pressure Point Successful \n\n\nThe derived gas flow rate at Mscf/day is: ${GFRlowAlt}", "COMPUTATION SUCCESSFUL", INFORMATION\_MESSAGE)**

**showMessageDialog(null, "Computation Performed!,\n ...Exiting Now..Thanks for using the application", "APP TO EXIT", INFORMATION\_MESSAGE)**

**System.exit(0)**

**} else {**

**showMessageDialog(null, "Apparently, you don't want to continue...." +**

**"\n\n Exiting the System Now..Thanks for using the Application", "APP TO EXIT!", INFORMATION\_MESSAGE)**

**System.exit(0)**

**}**

**} else {**

**showMessageDialog(null, "Apparently, you don't want to continue...." +**

**"\n\n Exiting the System Now..Thanks for using the Application", "APP TO EXIT!", INFORMATION\_MESSAGE)**

**System.exit(0)**

**}**

**} catch (Exception ex) {**

**ex.printStackTrace()**

**showMessageDialog(null, "Error occurred in Computation!", "Flow Rate Computation Failure", WARNING\_MESSAGE)**

**//showConfirmDialog(null, "Do you want to continue to the alternative computation of General Gas Flow?")**

**}**

**} else {**

**showMessageDialog(null, "Application Cancelled...Goodbye")**

**System.exit(0)**

**}**

**break**

**}**

**}**

**def prodIndex = {**

**showMessageDialog(null, "You have Chosen the Productivity Index Calculation!", "PRODUCTIVITY INDEX SELECTED", INFORMATION\_MESSAGE)**

**String[] PIchoices = ["GENERAL", "HIGH", "LOW"]**

**int PIchosen = showOptionDialog(null, "Sorry to bother you...\nBut you have to choose among the Gas Flow Rate Computation Choices\n\n\n\n\n\n",**

**"Make a choice Now!", DEFAULT\_OPTION, INFORMATION\_MESSAGE, null, PIchoices, PIchoices[0])**

**switch (PIchosen) {**

**case 0://general perm, thick, avrPseudo, avrWellFactor, temp, drainRad, wellBoreRad, skinFac**

**//Prompt the user to enter appropriate parameters:**

**JTextField perm = new JTextField("Permeability, K here")**

**JTextField thick = new JTextField("Thickness, H here")**

**JTextField avrPseudo = new JTextField("Average Reservoir Real Gas Pseudo-Pressure")**

**JTextField avrWellFactor = new JTextField("Average Well Flowing Pressure")**

**JTextField temp = new JTextField("Temperature here")**

**JTextField drainRad = new JTextField("Drainage Radius here")**

**JTextField wellBoreRad = new JTextField("WellBore Radius here")**

**JTextField skinFac = new JTextField("Skin Factor for general here")**

**Object[] options = [perm, thick, avrPseudo, avrWellFactor, temp, drainRad, wellBoreRad, skinFac]**

**showMessageDialog( null, "INSTRUCTION:\nSupply all integer values as decimal i.e 2 as 2.0")**

**int computeCommand = showConfirmDialog(null, options, "PRODUCTIVITY INDEX - GENERAL PRESSURE REGION", OK\_CANCEL\_OPTION)**

**showMessageDialog(null, "The supplied gas parameters above will be used to compute \nthe Productivity Index at General Pressure Point")**

**if (computeCommand == OK\_OPTION) {**

**//Begin the computation:**

**try {**

**def PIgeneral = this.PIgeneral(perm.text as Double, thick.text as Double, avrPseudo.text as Double, avrWellFactor.text as Double,**

**temp.text as Double, drainRad.text as Double, wellBoreRad.text as Double, skinFac.text as Double)**

**showMessageDialog(null, "Productivity Index for General Pressure Point Successful \n\n\nThe derived Productivity Index is: ${PIgeneral}",**

**"COMPUTATION SUCCESSFUL", INFORMATION\_MESSAGE)**

**int yes\_no = showConfirmDialog(null, "Do you want to continue to the alternative computation of Productivity Index at General Pressure Point?\n\n", "HEY!, WANNA CONTINUE?", YES\_NO\_OPTION)**

**if (yes\_no == YES\_OPTION) {//GFRgeneral, avrPseudo, avrWellFactor**

**JTextField genGFR = new JTextField("Gas Flow Rate for General Pressure Point ")**

**JTextField avrPseudoAlt = new JTextField("Average Reservoir Real Gas Pseudo-Pressure")**

**JTextField avrWellFactorAlt = new JTextField("Average Well Flowing Pressure")**

**Object[] optionsAlt = [genGFR, avrPseudoAlt, avrWellFactorAlt]**

**showMessageDialog(null, "Welcome to the alternative computation of Productivity Index at General Pressure Point.")**

**showMessageDialog( null, "INSTRUCTION:\nSupply all integer values as decimal i.e 2 as 2.0")**

**int chooseAlt = showConfirmDialog(null, optionsAlt, "PRODUCTIVITY INDEX ALTERNATIVE - GENERAL PRESSURE REGION", YES\_NO\_OPTION)**

**showMessageDialog(null, "\n\n The supplied gas parameters above will be used to compute\n" +**

**"the productivity Index at General Pressure Point")**

**if (chooseAlt == YES\_OPTION) {**

**def PIgenAlt = this.PIgenAlt(genGFR.text as Double, avrPseudoAlt.text as Double, avrWellFactorAlt.text as Double)**

**showMessageDialog(null, "Productivity Index for General Pressure Point Successful \n\n\nThe derived Productivity Index is: ${PIgenAlt}", "COMPUTATION SUCCESSFUL", INFORMATION\_MESSAGE)**

**showMessageDialog(null, "Computation Performed!,\n ...Exiting Now..Thanks for using the application", "APP TO EXIT", INFORMATION\_MESSAGE)**

**System.exit(0)**

**} else {**

**showMessageDialog(null, "Apparently, you dont want to continue...." +"\n\n Exiting the System Now..Thanks for using the Application", "APP TO EXIT!", INFORMATION\_MESSAGE)**

**System.exit(0)**

**}**

**} else {**

**showMessageDialog(null, "Apparently, you dont want to continue...." +**

**"\n\n Exiting the System Now..Thanks for using the Application", "APP TO EXIT!", INFORMATION\_MESSAGE)**

**System.exit(0)**

**}**

**} catch (Exception ex) {**

**ex.printStackTrace()**

**showMessageDialog(null, "Error occurred in Computation!", "Productivity Index Computation Failure", WARNING\_MESSAGE)**

**}**

**} else {**

**showMessageDialog(null, "Application Cancelled...Goodbye")**

**System.exit(0)**

**}**

**break**

**case 1://high**

**showMessageDialog(null, "OOps! No computation exist for Productivity Index at High Pressure Point!", "NO\_COMPUTATION\_EXIST", WARNING\_MESSAGE)**

**System.exit(0)**

**break**

**case 2://low perm, thick, temp, avrPressure, avrPressureWellFactor, gasViscosity, gasCompressFac, drainRad, wellBoreRad, skinFac**

**//Prompt the user to enter appropriate parameters:**

**JTextField perm = new JTextField("Permeability, K here")**

**JTextField thick = new JTextField("Thickness, H here")**

**JTextField avrPressure = new JTextField("Average Reservoir Real Gas Pseudo-Pressure")**

**JTextField avrPressureWellFactor = new JTextField("Average Well Flowing Pressure")**

**JTextField temp = new JTextField("Temperature here")**

**JTextField gasViscosity = new JTextField("Gas Viscosity here")**

**JTextField gasCompressFac = new JTextField("Gas compressibity Factor")**

**JTextField drainRad = new JTextField("Drainage Radius here")**

**JTextField wellBoreRad = new JTextField("WellBore Radius here")**

**JTextField skinFac = new JTextField("Skin Factor for general here")**

**Object[] options = [perm, thick, avrPressure, avrPressureWellFactor, temp, gasViscosity, gasCompressFac, drainRad, wellBoreRad, skinFac]**

**showMessageDialog(null, "INSTRUCTION: Supply all integer values as decimal i.e 2 as 2.0")**

**int computeCommand = showConfirmDialog(null, options, "PRODUCTIVITY INDEX - LOW PRESSURE REGION", OK\_CANCEL\_OPTION)**

**showMessageDialog(null, "The supplied gas parameters previously obtained will be used to compute \nthe productivity Index at low pressure point")**

**if (computeCommand == OK\_OPTION) {**

**//Begin the computation:**

**try {**

**def PIlow = this.PIlow(perm.text as Double, thick.text as Double, avrPressure.text as Double, avrPressureWellFactor.text as Double, temp.text as Double,**

**gasViscosity.text as Double, gasCompressFac.text as Double, drainRad.text as Double, wellBoreRad.text as Double, skinFac.text as Double)**

**showMessageDialog(null, "Productivity Index for Low Pressure Point Successful \n\n\nThe derived Productivity Index is: ${PIlow}",**

**"COMPUTATION SUCCESSFUL", INFORMATION\_MESSAGE)**

**int yes\_no = showConfirmDialog(null, "Do you want to continue to the alternative computation of Productivity Index at Low Pressure Point?\n\n", "HEY!, WANNA CONTINUE?", YES\_NO\_OPTION)**

**if (yes\_no == YES\_OPTION) {//GFRlow, avrPressure, avrPressureWellFactor**

**JTextField GFRlow = new JTextField("Gas Flow Rate for General Pressure Point")**

**JTextField avrPressureAlt = new JTextField("Average Reservoir Real Gas Pseudo-Pressure")**

**JTextField avrPressureWellFactorAlt = new JTextField("Average Well Flowing Pressure")**

**Object[] optionsAlt = [GFRlow, avrPressureAlt, avrPressureWellFactorAlt]**

**showMessageDialog(null, "Welcome to the alternative computation of Productivity Index at Low Pressure Point.")**

**showMessageDialog(null, "INSTRUCTION: Supply all integer values as decimal i.e 2 as 2.0")**

**int chooseAlt = showConfirmDialog(null, optionsAlt, "PRODUCTIVITY INDEX ALTERNATIVE - LOW PRESSURE REGION", YES\_NO\_OPTION)**

**showMessageDialog(null, "\nThe supplied gas parameters above will be used to compute\nthe Productivity Index at Low Pressure Point")**

**if (chooseAlt == YES\_OPTION) {**

**def PIlowAlt = this.PIlowAlt(GFRlow.text as Double, avrPressureAlt.text as Double, avrPressureWellFactorAlt.text as Double)**

**showMessageDialog(null, "Productivity Index for Low Pressure Point Successful \n\n\nThe derived productivity Index is: ${PIlowAlt}", "COMPUTATION SUCCESSFUL", INFORMATION\_MESSAGE)**

**showMessageDialog(null, "Computation Performed!,\n ...Exiting Now..Thanks for using the application", "APP TO EXIT", INFORMATION\_MESSAGE)**

**System.exit(0)**

**} else {**

**showMessageDialog(null, "Apparently, you don't want to continue...." +"\n\n Exiting the System Now..Thanks for using the Application", "APP TO EXIT!", INFORMATION\_MESSAGE)**

**System.exit(0)**

**}**

**} else {**

**showMessageDialog(null, "Apparently, you don't want to continue...." +**

**"\n\n Exiting the System Now..Thanks for using the Application", "APP TO EXIT!", INFORMATION\_MESSAGE)**

**System.exit(0)**

**}**

**} catch (Exception ex) {**

**ex.printStackTrace()**

**showMessageDialog(null, "Error occurred in Computation!", "Productivity Index Computation Failure", WARNING\_MESSAGE)**

**//showConfirmDialog(null, "Do you want to continue to the alternative computation of ?")**

**showMessageDialog(null, "\n\n Exiting the System Now..Thanks for using the Application",**

**"APP TO EXIT!", INFORMATION\_MESSAGE)**

**System.exit(0)**

**}**

**} else {**

**showMessageDialog(null, "Application Cancelled...Goodbye")**

**System.exit(0)**

**}**

**break**

**}**

**def skinFactor = {**

**showMessageDialog(null, "You have Chosen the Skin Factor Calculation!", "SKIN FACTOR SELECTED", INFORMATION\_MESSAGE)**

**String[] SFchoices = ["UNIVERSAL", "NORMAL", "GENERAL", "HIGH", "LOW"]**

**int SFchosen = showOptionDialog(null, "Sorry to bother you...\nBut you have to choose among the Gas Flow Rate Computation Choices\n\n\n\n\n\n",**

**"Make a choice Now!", DEFAULT\_OPTION, INFORMATION\_MESSAGE, null, SFchoices, SFchoices[0])**

**switch (SFchosen) {**

**case 0://Universal perm, permDueToSkin, damageRad, wellBoreRad**

**//Prompt the user to enter appropriate parameters:**

**JTextField perm = new JTextField("Permeability, K here")**

**JTextField permDueToSkin = new JTextField("Permeability due to skin here")**

**JTextField drainRad = new JTextField("Drainage Radius here")**

**JTextField wellBoreRad = new JTextField("WellBore Radius here")**

**Object[] options = [perm, permDueToSkin, drainRad, wellBoreRad]**

**showMessageDialog(null, "NB: Supply all integer values as decimal i.e 2 as 2.0")**

**int computeCommand = showConfirmDialog(null, options, "SKIN FACTOR - UNIVERSAL PRESSURE REGION", OK\_CANCEL\_OPTION)**

**showMessageDialog(null, "The supplied gas parameters above will be used to compute \nthe Skin Factor at Universal pressure point")**

**if (computeCommand == OK\_OPTION) {**

**//Begin the computation:**

**try {**

**def SFuniversal = this.SkinFactorUniversal(perm.text as Double, permDueToSkin.text as Double, drainRad.text as Double, wellBoreRad.text as Double) as Map**

**showMessageDialog(null, "Skin Factor for Universal Pressure Region Successful" +**

**"\n\n\nThe derived Universal Pressure Point is: ${SFuniversal["SFgenValue"]}" + "\n\n The state of the region is : ${SFuniversal["SFgenState"]}",**

**"COMPUTATION SUCCESSFUL", INFORMATION\_MESSAGE)**

**showMessageDialog(null, "Thanks for using the Application", "ABOUT TO EXIT", INFORMATION\_MESSAGE)**

**System.exit(0)**

**} catch(Exception ex) {**

**ex.printStackTrace()**

**showMessageDialog(null, "Error occurred in Computation!", "Flow Rate Computation Failure", WARNING\_MESSAGE)**

**System.exit(0)**

**}**

**} else {**

**showMessageDialog(null, "Application Cancelled...Goodbye")**

**System.exit(0)**

**}**

**break**

**//"UNIVERSAL","NORMAL", "GENERAL", "HIGH", "LOW"**

**case 1://normal: NormalFlowRate, gasPerm, thick, avrPressure, avrPressureWellFactor, gasViscosity, formVolFac, drainRad, wellBoreRad**

**//Prompt the user to enter appropriate parameters:**

**JTextField NormalFlowRate = new JTextField("Normal Flow Rate here")**

**JTextField gasPerm = new JTextField("Permeability, K here")**

**JTextField thick = new JTextField("Thickness, H here")**

**JTextField avrPressure = new JTextField("Average Reservoir Real Gas Pseudo-Pressure")**

**JTextField avrPressureWellFactor = new JTextField("Average Well Flowing Pressure")**

**JTextField gasViscosity = new JTextField("Gas Viscosity here")**

**JTextField formVolFac = new JTextField("Gas Formation Volume Factor ")**

**JTextField drainRad = new JTextField("Drainage Radius here")**

**JTextField wellBoreRad = new JTextField("WellBore Radius here")**

**Object[] options = [NormalFlowRate, gasPerm, thick, avrPressure, avrPressureWellFactor, gasViscosity, formVolFac, drainRad, wellBoreRad]**

**showMessageDialog(null, "INSTRUCTION: Supply all integer values as decimal i.e 2 as 2.0")**

**int computeCommand = showConfirmDialog(null, options, "SKIN FACTOR - NORMAL PRESSURE REGION", OK\_CANCEL\_OPTION)**

**showMessageDialog(null, "The supplied gas parameters previously obtained will be used to compute \nthe Skin Factor at Normal Pressure Region")**

**if (computeCommand == OK\_OPTION) {**

**//Begin the computation:**

**try {**

**def SFnormal = this.SkinFactorFromFlowPrediction(NormalFlowRate.text as Double, gasPerm.text as Double, thick.text as Double, avrPressure.text as Double, avrPressureWellFactor.text as Double, gasViscosity.text as Double, formVolFac.text as Double, drainRad.text as Double, wellBoreRad.text as Double) as Map**

**showMessageDialog(null, "Skin Factor for Normal Pressure Point Successful" +**

**"\n\n\nThe derived Skin Factor for Normal Flow Rate is: ${SFnormal["SFgenValue"]}" +**

**"\n\n The state of the region is : ${SFnormal["SFgenState"]}", "COMPUTATION SUCCESSFUL", INFORMATION\_MESSAGE)**

**showMessageDialog(null, "Thanks for Using this Application, About to exit now", "EXIT", INFORMATION\_MESSAGE)**

**System.exit(0)**

**} catch (Exception ex) {**

**ex.printStackTrace()**

**showMessageDialog(null, "Error occurred in Computation!", "Flow Rate Computation Failure", WARNING\_MESSAGE)**

**System.exit(0)**

**}**

**} else {**

**showMessageDialog(null, "Application Cancelled...Goodbye")**

**System.exit(0)**

**}**

**break**

**case 2://general GFRgeneral, perm, thick, avrPseudo, avrWellFactor, temp, drainRad, wellBoreRad**

**//Prompt the user to enter appropriate parameters:**

**JTextField GFRgeneral = new JTextField("Gas Flow Rate at general point here")**

**JTextField perm = new JTextField("Permeability, K here")**

**JTextField thick = new JTextField("Thickness, H here")**

**JTextField avrPseudo = new JTextField("Average Reservoir Real Gas Pseudo-Pressure")**

**JTextField avrWellFactor = new JTextField("Average Well Flowing Pressure")**

**JTextField temp = new JTextField("Temperature here")**

**JTextField drainRad = new JTextField("Drainage Radius here")**

**JTextField wellBoreRad = new JTextField("WellBore Radius here")**

**Object[] options = [GFRgeneral, perm, thick, avrPseudo, avrWellFactor, temp, drainRad, wellBoreRad]**

**showMessageDialog(null,"INSTRUCTION: Supply all integer values as decimal i.e 2 as 2.0")**

**int computeCommand = showConfirmDialog(null, options, "SKIN FACTOR - GENERAL PRESSURE REGION", OK\_CANCEL\_OPTION)**

**showMessageDialog(null, "\nThe supplied gas parameters previously obtained will be used to compute \nthe Skin Factor at general pressure Region")**

**if (computeCommand == OK\_OPTION) {**

**//Begin the computation:**

**try {**

**def SFgeneral = this.SkinFactorGeneral(GFRgeneral.text as Double, perm.text as Double, thick.text as Double, avrPseudo.text as Double,**

**avrWellFactor.text as Double, temp.text as Double, drainRad.text as Double, wellBoreRad.text as Double) as Map**

**showMessageDialog(null, "Skin Factor for General Pressure Point Successful \n\n\nThe derived Skin Factor is: ${SFgeneral["SFgenValue"]}" + "\n\n The state of the region is : ${SFgeneral["SFgenState"]}",**

**"COMPUTATION SUCCESSFUL", INFORMATION\_MESSAGE)**

**showMessageDialog(null, "Apparently, you don't want to continue...." +**

**"\n\n Exiting the System Now..Thanks for using the Application", "APP TO EXIT!", INFORMATION\_MESSAGE)**

**System.exit(0)**

**showMessageDialog(null, "Apparently, you don't want to continue...." +**

**"\n\n Exiting the System Now..Thanks for using the Application", "APP TO EXIT!", INFORMATION\_MESSAGE)**

**System.exit(0)**

**} catch (Exception ex) {**

**ex.printStackTrace()**

**showMessageDialog(null, "Error occurred in Computation!", "Skin Computation Error", WARNING\_MESSAGE)**

**//showConfirmDialog(null, "Do you want to continue to the alternative computation of General Gas Flow?")**

**}**

**} else {**

**showMessageDialog(null, "Application Cancelled...Goodbye")**

**System.exit(0)**

**}**

**break**

**case 3://high GFRhigh, perm, thick, avrPressure, avrPressureWellFactor, gasViscosity, formVolFac, drainRad, wellBoreRad**

**//Prompt the user to enter appropriate parameters:**

**JTextField GFRhigh = new JTextField("Gas Flow Rate at High Pressure Region")**

**JTextField perm = new JTextField("Permeability, K here")**

**JTextField thick = new JTextField("Thickness, H here")**

**JTextField avrPressure = new JTextField("Average Reservoir Real Gas Pseudo-Pressure")**

**JTextField avrWellFactor = new JTextField("Average Well Flowing Pressure")**

**JTextField gasViscosity = new JTextField("Gas Viscosity here")**

**JTextField formVolFac = new JTextField("Formation Volume Factor here")**

**JTextField drainRad = new JTextField("Drainage Radius here")**

**JTextField wellBoreRad = new JTextField("WellBore Radius here")**

**Object[] options = [GFRhigh, perm, thick, avrPressure, avrWellFactor, gasViscosity, formVolFac, drainRad, wellBoreRad]**

**showMessageDialog(null, "NB: Supply all integer values as decimal i.e 2 as 2.0")**

**int computeCommand = showConfirmDialog(null, options, "SKIN FACTOR - HIGH PRESSURE REGION", OK\_CANCEL\_OPTION)**

**showMessageDialog(null, "The supplied gas parameters previously obtained will be used to compute \nthe Skin Factor at high pressure Region")**

**if (computeCommand == OK\_OPTION) {**

**//Begin the computation:**

**try {**

**def SFhigh = this.SkinFactorHigh(GFRhigh.text as Double, perm.text as Double, thick.text as Double, avrPressure.text as Double,**

**avrWellFactor.text as Double, gasViscosity.text as Double, formVolFac.text as Double,**

**drainRad.text as Double, wellBoreRad.text as Double) as Map**

**showMessageDialog(null, "Skin Factor for High Pressure Point Successful \n\n\nThe derived Skin Factor is: ${SFhigh["SFgenValue"]}" + "\n\n The state of the region is : ${SFhigh["SFgenState"]}",**

**"COMPUTATION SUCCESSFUL", INFORMATION\_MESSAGE)**

**showMessageDialog(null, "\n\n Exiting the System Now..Thanks for using the Application", "APP TO EXIT!", INFORMATION\_MESSAGE)**

**System.exit(0)**

**showMessageDialog(null, "\n\n Exiting the System Now..Thanks for using the Application", "APP TO EXIT!", INFORMATION\_MESSAGE)**

**System.exit(0)**

**} catch (Exception ex) {**

**ex.printStackTrace()**

**showMessageDialog(null, "Error occurred in Computation!", "Skin Computation Error", WARNING\_MESSAGE)**

**System.exit(0)**

**}**

**} else {**

**showMessageDialog(null, "Application Cancelled...Goodbye")**

**System.exit(0)**

**}**

**break**

**case 4: //low:GFRlow, perm, thick, avrPressure, avrPressureWellFactor, temp, gasViscosity, gasCompressFac, drainRad, wellBoreRad, skinFac**

**//Prompt the user to enter appropriate parameters:**

**JTextField GFRlow = new JTextField("Gas Flow Rate at High Pressure Region")**

**JTextField perm = new JTextField("Permeability, K here")**

**JTextField thick = new JTextField("Thickness, H here")**

**JTextField avrPressure = new JTextField("Average Reservoir Real Gas Pseudo-Pressure")**

**JTextField avrPressureWellFactor = new JTextField("Average Well Flowing Pressure")**

**JTextField gasViscosity = new JTextField("Gas Viscosity here")**

**JTextField formVolFac = new JTextField("Formation Volume Factor here")**

**JTextField drainRad = new JTextField("Drainage Radius here")**

**JTextField wellBoreRad = new JTextField("WellBore Radius here")**

**Object[] options = [GFRlow, perm, thick, avrPressure, avrPressureWellFactor, gasViscosity, formVolFac, drainRad, wellBoreRad]**

**showMessageDialog(null, "INSTRUCTION: Supply all integer values as decimal i.e 2 as 2.0")**

**int computeCommand = showConfirmDialog(null, options, "SKIN FACTOR - LOW PRESSURE REGION", OK\_CANCEL\_OPTION)**

**showMessageDialog(null, "The supplied gas parameters previously obtained will be used to compute \nthe Skin Factor at low pressure Region")**

**if (computeCommand == OK\_OPTION) {**

**//Begin the computation:**

**try {**

**def SFlow = this.SkinFactorHigh(GFRlow.text as Double, perm.text as Double, thick.text as Double, avrPressure.text as Double,**

**avrPressureWellFactor.text as Double, gasViscosity.text as Double,**

**formVolFac.text as Double, drainRad.text as Double, wellBoreRad.text as Double) as Map**

**showMessageDialog(null, "Skin Factor for High Pressure Point Successful \n\n\nThe derived Skin Factor is: ${SFlow["SFgenValue"]}" + "\n\n The state of the region is : ${SFlow["SFgenState"]}",**

**"COMPUTATION SUCCESSFUL", INFORMATION\_MESSAGE)**

**showMessageDialog(null, "\n\n Exiting the System Now..Thanks for using the Application", "APP TO EXIT!", INFORMATION\_MESSAGE)**

**System.exit(0)**

**} catch (Exception ex) {**

**ex.printStackTrace()**

**showMessageDialog(null, "Error occurred in Computation!", "Skin Computation Error", WARNING\_MESSAGE)**

**System.exit(0)**

**}**

**} else {**

**showMessageDialog(null, "Application Cancelled...Goodbye")**

**System.exit(0)**

**}**

**break**

**}**

**}**

**def flowPrediction = {**

**showMessageDialog(null, "You have Chosen the FlowRate Prediction Calculation", "FLOWRATE PREDICTION SELECTED", INFORMATION\_MESSAGE)**

**String[] FPchoices = ["PAST", "PRESENT", "FUTURE"]**

**int FPchosen = showOptionDialog(null, "Sorry to bother you...\nBut you have to choose among the Gas Flow Rate Computation Choices\n\n\n\n\n\n",**

**"Make a choice Now!", DEFAULT\_OPTION, INFORMATION\_MESSAGE, null, FPchoices, FPchoices[0])**

**switch (FPchosen) {**

**case 0://Normal gasPerm, thick, avrPressure, avrPressureWellFactor, gasViscosity, formVolFac, drainRad, wellBoreRad, scaleFactor**

**//Prompt the user to enter appropriate parameters:**

**JTextField gasPerm = new JTextField("Permeability, K here")**

**JTextField thick = new JTextField("Thickness, H here")**

**JTextField avrPressure = new JTextField("Average Reservoir Real Gas Pseudo-Pressure")**

**JTextField avrPressureWellFactor = new JTextField("Average Well Flowing Pressure")**

**JTextField gasViscosity = new JTextField("Gas Viscosity here")**

**JTextField formVolFac = new JTextField("Gas Formation Volume Factor")**

**JTextField drainRad = new JTextField("Drainage Radius here")**

**JTextField wellBoreRad = new JTextField("WellBore Radius here")**

**JTextField scaleFactor = new JTextField("Scale Factor here")**

**Object[] options = [gasPerm, thick, avrPressure, avrPressureWellFactor, gasViscosity, formVolFac, drainRad, wellBoreRad, scaleFactor]**

**showMessageDialog(null, "Supply all integer values as decimal i.e 2 as 2.0")**

**int computeCommand = showConfirmDialog(null, options, "NORMAL FLOW RATE", OK\_CANCEL\_OPTION)**

**showMessageDialog(null, "The supplied gas parameters above will be used \nto compute the Universal/Normal FlowRate")**

**if (computeCommand == OK\_OPTION) {**

**//Begin the computation:**

**try {**

**def FRnormal = this.NormalFlowRate(gasPerm.text as Double, thick.text as Double, avrPressure.text as Double,**

**avrPressureWellFactor.text as Double, gasViscosity.text as Double,**

**formVolFac.text as Double, drainRad.text as Double, wellBoreRad.text as Double, scaleFactor.text as Double)**

**showMessageDialog(null, "Skin Factor for Universal Pressure Region Successful \n\n\nThe derived Universal Pressure Point is: ${FRnormal}", "COMPUTATION SUCCESSFUL", INFORMATION\_MESSAGE)**

**showMessageDialog(null, "Thanks for using the Application", "ABOUT TO EXIT", INFORMATION\_MESSAGE)**

**System.exit(0)**

**} catch (Exception ex) {**

**ex.printStackTrace()**

**showMessageDialog(null, "Error occurred in Computation!", "Flow Rate Computation Failure", WARNING\_MESSAGE)**

**System.exit(0)**

**}**

**} else {**

**showMessageDialog(null, "Application Cancelled...Goodbye")**

**System.exit(0)**

**}**

**break**

**case 1://present: GasPermPresent, perm, thick, avrPressurePresent, gasViscosity, formVolFac**

**//Prompt the user to enter appropriate parameters:**

**JTextField gasPermPresent = new JTextField("Permeability, K for Present Flow here")**

**JTextField gasPermNormal = new JTextField("Universal Permeability, K here")**

**JTextField thick = new JTextField("Thickness, H here")**

**JTextField avrPressurePresent = new JTextField("Average Reservoir Present Real Gas Pressure")**

**JTextField gasViscosity = new JTextField("Gas Viscosity here")**

**JTextField formVolFac = new JTextField("Gas Formation Volume Factor ")**

**Object[] options = [gasPermPresent, gasPermNormal, thick, avrPressurePresent, gasViscosity, formVolFac]**

**showMessageDialog(null, "INSTRUCTION: \nSupply all integer values as decimal i.e 2 as 2.0")**

**int computeCommand = showConfirmDialog(null, options, "PRESENT FLOW RATE", OK\_CANCEL\_OPTION)**

**showMessageDialog(null, "The supplied gas parameters previously obtained will be used to compute" +**

**"\nthe Flow Rate at Present Pressure Region")**

**if (computeCommand == OK\_OPTION) {**

**//Begin the computation:**

**try {**

**def FRpresent = this.PresentFlowRate(gasPermPresent.text as Double, gasPermNormal.text as Double, thick.text as Double, avrPressurePresent.text as Double,**

**gasViscosity.text as Double, formVolFac.text as Double)**

**showMessageDialog(null, "Skin Factor for Normal Pressure Point Successful" +**

**"\n\n\nThe derived Skin Factor for Normal Flow Rate is: ${FRpresent}", "COMPUTATION SUCCESSFUL", INFORMATION\_MESSAGE)**

**showMessageDialog(null, "Thanks for Using this Application, About to exit now", "EXIT", INFORMATION\_MESSAGE)**

**System.exit(0)**

**} catch (Exception ex) {**

**ex.printStackTrace()**

**showMessageDialog(null, "Error occurred in Computation!", "Flow Rate Computation Failure", WARNING\_MESSAGE)**

**System.exit(0)**

**}**

**} else {**

**showMessageDialog(null, "Application Cancelled...Goodbye")**

**System.exit(0)**

**}**

**break**

**case 2://future GasPermFuture, perm, thick, avrPressureFuture, gasViscosity, formVolFac**

**//Prompt the user to enter appropriate parameters:**

**JTextField gasPermPresent = new JTextField("Permeability, K for Future Flow here")**

**JTextField gasPermNormal = new JTextField("Universal Permeability, K here")**

**JTextField thick = new JTextField("Thickness, H here")**

**JTextField avrPressurePresent = new JTextField("Average Reservoir Present Real Gas Pressure")**

**JTextField gasViscosity = new JTextField("Gas Viscosity here")**

**JTextField formVolFac = new JTextField("Gas Formation Volume Factor ")**

**Object[] options = [gasPermPresent, gasPermNormal, thick, avrPressurePresent, gasViscosity, formVolFac]**

**showMessageDialog(null, "INSTRUCTION: \nSupply all integer values as decimal i.e 2 as 2.0")**

**int computeCommand = showConfirmDialog(null, options, "FUTURE FLOW RATE", OK\_CANCEL\_OPTION)**

**showMessageDialog(null, "The supplied gas parameters obtained previously will be used to compute" +**

**"\nthe Flow Rate at Future Pressure Region")**

**if (computeCommand == OK\_OPTION) {**

**//Begin the computation:**

**try {**

**def FRfuture = this.FutureFlowRate(gasPermPresent.text as Double, gasPermNormal.text as Double,**

**thick.text as Double, avrPressurePresent.text as Double, gasViscosity.text as Double, formVolFac.text as Double)**

**showMessageDialog(null, "Flow Rate for Future Pressure Point Successful" +**

**"\n\n\nThe derived Skin Factor for Normal Flow Rate is: ${FRfuture}", "COMPUTATION SUCCESSFUL", INFORMATION\_MESSAGE)**

**showMessageDialog(null, "Thanks for Using this Application, About to exit now", "EXIT", INFORMATION\_MESSAGE)**

**System.exit(0)**

**} catch (Exception ex) {**

**ex.printStackTrace()**

**showMessageDialog(null, "Error occurred in Computation!", "Flow Rate Computation Failure", WARNING\_MESSAGE)**

**System.exit(0)**

**}**

**} else {**

**showMessageDialog(null, "Application Cancelled...Goodbye")**

**System.exit(0)**

**}**

**break**

**}**

**}**

**}**

**Package PressureRegion**

**GasFlowRate.groovy**

**package PressureRegion;**

**import java.lang.Math**

**import static javax.swing.JOptionPane.\***

**public trait GasFlowRate {**

**def logCalc (drainRad, wellBoreRad) {**

**def logCalc = Math.log((drainRad/wellBoreRad) as Double)**

**return logCalc**

**}**

**def commonDenumCalc(drainRad, wellBoreRad, skinFac){**

**def partialDenum = this.logCalc(drainRad, wellBoreRad) - 0.75 + skinFac**

**return partialDenum**

**}**

**//GasFlowRate for various Pressure Regions:**

**def GFRgeneral(perm, thick, avrPseudo, avrWellFactor, temp, drainRad, wellBoreRad, skinFac){**

**def EqnNum = perm \* thick \* (avrPseudo - avrWellFactor)**

**def EqnDenum = 1422 \* temp \* (this.commonDenumCalc(drainRad, wellBoreRad, skinFac))**

**def GFRgen = EqnNum / EqnDenum**

**return GFRgen**

**}**

**def GFRgeneralAlt (genProdIndex, avrPseudo, avrWellFactor){**

**def GFRgenAlt = genProdIndex \* (avrPseudo \* avrWellFactor)**

**return GFRgenAlt**

**}**

**def GFRhigh(perm, thick, avrPressure, avrPressureWellFactor, gasViscosity, formVolFac, drainRad, wellBoreRad, skinFac){**

**def EqnNum = 7.08 \* (1/Math.pow(10.0, 6.0)) \* perm \* thick \* (avrPressure - avrPressureWellFactor)**

**def EqnDenum = gasViscosity \* formVolFac \* (this.commonDenumCalc(drainRad, wellBoreRad, skinFac))**

**def GFRhigh = EqnNum/EqnDenum**

**return GFRhigh**

**}**

**def GFRhighAlt = {**

**return null;**

**}**

**def GFRlow(perm, thick, avrPressure, avrPressureWellFactor, temp, gasViscosity, gasCompressFac, drainRad, wellBoreRad, skinFac){**

**def EqnNum = perm \* thick \* ((Math.pow(avrPressure, 2.0)) - (Math.pow(avrPressureWellFactor, 2.0)))**

**def EqnDenum = 1422 \* temp \* (gasViscosity \* gasCompressFac) \* (this.commonDenumCalc(drainRad, wellBoreRad, skinFac))**

**def GFRlow = EqnNum/EqnDenum**

**return GFRlow**

**}**

**def GFRlowAlt (lowProIndex, avrPressure, avrPressureWellFactor){**

**def GFRlowAlt = lowProIndex \* ((Math.pow((double)avrPressure, 2.0)) - (Math.pow((double)avrPressureWellFactor, 2.0)))**

**return GFRlowAlt**

**}**

**}**

**Package PressureRegion**

**ProductivityIndex.groovy**

**package PressureRegion;**

**trait ProductivityIndex extends GasFlowRate {**

**def PIgeneral (perm, thick, avrPseudo, avrWellFactor, temp, drainRad, wellBoreRad, skinFac){**

**def EqnNum = perm \* thick**

**def EqnDenum = 1422 \* temp \* (commonDenumCalc(drainRad, wellBoreRad, skinFac))**

**def prodIndex = EqnNum/EqnDenum**

**return prodIndex**

**}**

**def PIgenAlt (GFRgeneral, avrPseudo, avrWellFactor){**

**def PIgenAlt = GFRgeneral/(avrPseudo - avrWellFactor)**

**return PIgenAlt**

**}**

**def PIhigh(){**

**return null**

**}**

**def PIhighAlt = {**

**return null**

**}**

**def PIlow(perm, thick, temp, avrPressure, avrPressureWellFactor, gasViscosity, gasCompressFac, drainRad, wellBoreRad, skinFac){**

**def EqnNum = perm \* thick**

**def EqnDenum = 1422 \* temp \* (gasViscosity \* gasCompressFac) \* (commonDenumCalc(drainRad, wellBoreRad, skinFac))**

**def prodIndex = EqnNum/EqnDenum**

**return prodIndex**

**}**

**def PIlowAlt (GFRlow, avrPressure, avrPressureWellFactor){**

**def PIlowAlt = GFRlow/(avrPressure - avrPressureWellFactor)**

**return PIlowAlt**

**}**

**}**

**Package ProductivityIndex**

**SkinFactor.groovy**

**package PressureRegion**

**trait SkinFactor {**

**def SkinFactorUniversal(perm, permDueToSkin, damageRad, wellBoreRad){**

**def SFgenAlt = ((perm/permDueToSkin) - 1) \* Math.log((damageRad/wellBoreRad) as Double)**

**if (SFgenAlt == 0) {**

**return ["SFgenValue": SFgenAlt, "SFgenState": "NORMAL STATE"]**

**} else if(SFgenAlt < 0){**

**return ["SFgenValue": SFgenAlt, "SFgenState": "STIMULATION STATE"]**

**}else if(SFgenAlt > 0){**

**return ["SFgenValue": SFgenAlt, "SFgenState": "DAMAGED STATE"]**

**}**

**}**

**//SkinFactor to be obtained from Flow predictions:**

**def SkinFactorFromFlowPrediction(NormalFlowRate, gasPerm, thick, avrPressure, avrPressureWellFactor, gasViscosity, formVolFac, drainRad, wellBoreRad){**

**def CalcFirstPartNum = gasPerm \* thick \* (avrPressure - avrPressureWellFactor)**

**def CalcFirstPartDenum = 141.2 \* (gasViscosity \* formVolFac) \* NormalFlowRate**

**def SkinFactor = (CalcFirstPartNum/CalcFirstPartDenum) - Math.log((drainRad/wellBoreRad) as Double)**

**//return SkinFactor**

**if (SkinFactor == 0) {**

**return ["SFgenValue": SkinFactor, "SFgenState": "NORMAL STATE"]**

**} else if(SFgen < 0){**

**return ["SFgenValue": SkinFactor, "SFgenState": "STIMULATION STATE"]**

**}else if(SFgen > 0){**

**return ["SFgenValue": SkinFactor, "SFgenState": "DAMAGED STATE"]**

**}**

**}**

**def SkinFactorGeneral (GFRgeneral, perm, thick, avrPseudo, avrWellFactor, temp, drainRad, wellBoreRad){**

**def firstPartCalc = (perm \* thick \* (avrPseudo \* avrWellFactor))/(1422 \* temp \* GFRgeneral)**

**def secondPartCalc = 0.75 - Math.log((drainRad/wellBoreRad) as Double)**

**def SFgen = firstPartCalc + secondPartCalc**

**//return {**

**if (SFgen == 0) {**

**return ["SFgenValue": SFgen, "SFgenState": "NORMAL STATE"]**

**} else if(SFgen < 0){**

**return ["SFgenValue": SFgen, "SFgenState": "STIMULATION STATE"]**

**}else if(SFgen > 0){**

**return ["SFgenValue": SFgen, "SFgenState": "DAMAGED STATE"]**

**}**

**//}**

**}**

**def SkinFactorHigh (GFRhigh, perm, thick, avrPressure, avrPressureWellFactor, gasViscosity, formVolFac, drainRad, wellBoreRad){**

**def FirstPartUpper = 7.08 \* (1/(Math.pow(10.0, 6.0))) \* perm \* thick \* (avrPressure - avrPressureWellFactor)**

**def FirstPartLower = GFRhigh \* gasViscosity \* formVolFac**

**def SecondPartCalc = 0.75 - Math.log((drainRad/wellBoreRad) as Double)**

**def SkinFactor = (FirstPartUpper/FirstPartLower) + SecondPartCalc**

**//return {**

**if (SkinFactor == 0) {**

**return ["SFgenValue": SkinFactor, "SFgenState": "NORMAL STATE"]**

**} else if(SkinFactor < 0){**

**return ["SFgenValue": SkinFactor, "SFgenState": "STIMULATION STATE"]**

**}else if(SkinFactor > 0){**

**return ["SFgenValue": SkinFactor, "SFgenState": "DAMAGED STATE"]**

**}**

**//}**

**}**

**def SkinFactorLow(GFRlow, perm, thick, avrPressure, avrPressureWellFactor, temp, gasViscosity, gasCompressFac, drainRad, wellBoreRad, skinFac){**

**def FirstPartUpper = perm \* thick \* ((Math.pow((Double)avrPressure, 2.0)) - (Math.pow((Double)avrPressureWellFactor, 2.0)))**

**def FirstPartLower = 1422 \* temp \* (gasViscosity \* gasCompressFac) \* GFRlow**

**def SecondPartCalc = 0.75 - Math.log((drainRad/wellBoreRad) as Double)**

**def SkinFactor = (FirstPartUpper/FirstPartLower) + SecondPartCalc**

**//return {**

**if (SkinFactor == 0) {**

**return ["SFgenValue": SkinFactor, "SFgenState": "NORMAL STATE"]**

**} else if(SkinFactor < 0){**

**return ["SFgenValue": SkinFactor, "SFgenState": "STIMULATION STATE"]**

**}else if(SkinFactor > 0){**

**return ["SFgenValue": SkinFactor, "SFgenState": "DAMAGED STATE"]**

**}**

**//}**

**}**

**}**

**Package FlowRate**

**FlowRate.groovy**

**package FlowRate**

**trait FlowRate {**

**def NormalFlowRate(gasPerm, thick, avrPressure, avrPressureWellFactor, gasViscosity,**

**formVolFac, drainRad, wellBoreRad, scaleFactor){**

**def EqnNum = gasPerm \* thick \* (avrPressure - avrPressureWellFactor)**

**def EqnDenum = 141.2 \* (gasViscosity \* formVolFac) \* (Math.log((drainRad/wellBoreRad) as Double) + scaleFactor)**

**def NormalFlowRate = EqnNum/EqnDenum**

**return NormalFlowRate**

**}**

**def PresentFlowRate(GasPermPresent, perm, thick, avrPressurePresent, gasViscosity, formVolFac){**

**def firstComp = (perm \* thick)/141.2**

**def secondComp = GasPermPresent/(gasViscosity \* formVolFac)**

**def PresentFlowRate = firstComp \* secondComp \* avrPressurePresent**

**return PresentFlowRate**

**}**

**def PresentOut = {GasPermPresent, perm, thick, avrPressurePresent, gasViscosity, formVolFac ->**

**return PresentFlowRate()**

**}.memoize()**

**def FutureFlowRate(GasPermFuture, perm, thick, avrPressureFuture, gasViscosity, formVolFac){**

**def FutureFlowRate = this.PresentOut(GasPermFuture, perm, thick, avrPressureFuture, gasViscosity, formVolFac)**

**return FutureFlowRate**

**}**

**}**

**…………………………………………………………………………………..**