**K.J. Somaiya College of Science and Commerce**

**Dept. of Information Technology**

Subject: **Advance Java**

Roll No.: **17-6839**

Name: **Vishal Himmat Kataria**

Project Name: **Calculator**

***Index***

* Concept
* Classes
* Flow Chart
* Output
* Conclusion
* References

Concept

* Calculator is use for to reduce the human efforts.
* Nowadays we have many different types of Calculators such as Simple Calculator and Scientific Calculators.
* With the help of calculators many works get completed within fraction of minutes.

Classes:-

1. JFrame :-

The *JFrame* class is slightly incompatible with Frame. Like all other JFC/Swing top-level containers, a *JFrame* contains a *JRootPane* as its only child. The content pane provided by the root pane should, as a rule, contain all the non-menu components displayed by the *JFrame*. This is different from the AWT *Frame* case. As a conveniance *add* and its variants, *remove* and *setLayout* have been overridden to forward to the *contentPane* as necessary.

1. StringBuffer :-

A thread-safe, mutable sequence of characters. A string buffer is like a [String](https://docs.oracle.com/javase/7/docs/api/java/lang/String.html), but can be modified. At any point in time it contains some particular sequence of characters, but the length and content of the sequence can be changed through certain method calls.

String buffers are safe for use by multiple threads. The methods are synchronized where necessary so that all the operations on any particular instance behave as if they occur in some serial order that is consistent with the order of the method calls made by each of the individual threads involved.

1. **Arraylist :-**

Resizable-array implementation of the List interface. Implements all optional list operations, and permits all elements, including null. In addition to implementing the List interface, this class provides methods to manipulate the size of the array that is used internally to store the list. (This class is roughly equivalent to Vector, except that it is unsynchronized.The size, isEmpty, get, set, iterator, and listIterator operations run in constant time. The add operation runs in *amortized constant time*, that is, adding n elements requires O(n) time. All of the other operations run in linear time (roughly speaking). The constant factor is low compared to that for the LinkedList implementation.

Each ArrayList instance has a *capacity*. The capacity is the size of the array used to store the elements in the list. It is always at least as large as the list size. As elements are added to an ArrayList, its capacity grows automatically. The details of the growth policy are not specified beyond the fact that adding an element has constant amortized time cost. An application can increase the capacity of an ArrayList instance before adding a large number of elements using the ensureCapacity operation. This may reduce the amount of incremental reallocation.

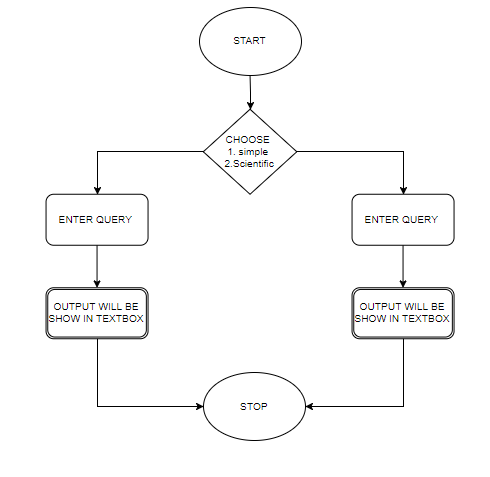
1. **ScriptEngine:-**

ScriptEngine is the fundamental interface whose methods must be fully functional in every implementation of this specification.   
The values are key/value pairs of two types. The first type of pairs consists of those whose keys are reserved and defined in this specification or by individual implementations. The values in the pairs with reserved keys have specified meanings.

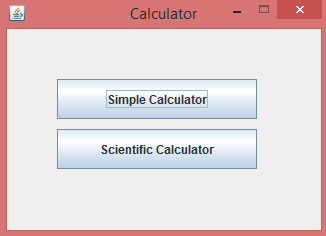
## ScriptEngineManager

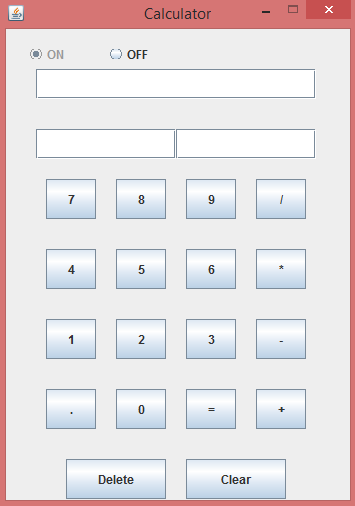
The ScriptEngineManager implements a discovery and instantiation mechanism for ScriptEngine classes and also maintains a collection of key/value pairs storing state shared by all engines created by the Manager. The ScriptEngineManager provides a method to return a list of all these factories as well as utility methods which look up factories on the basis of language name, file extension and mime type.

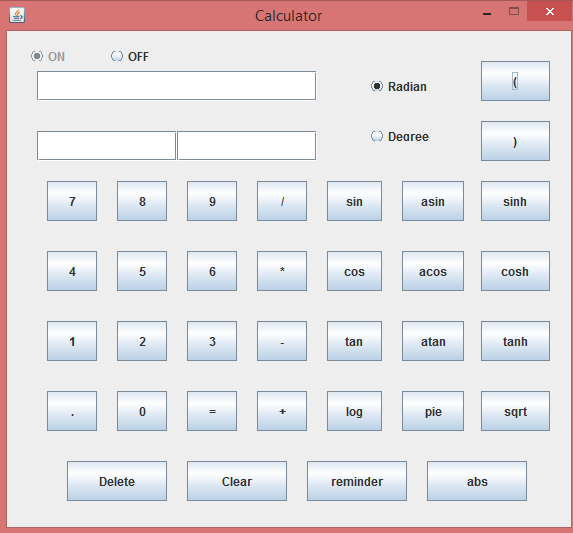
Flowchart:-



Output:-







Conclusion

* Designed and Developed a Java Swing Application using NetBeans IDE.
* Implemented ActionListener.
* Improved JAVA Programming skills.
* Improved Understanding of different JAVA programming and Listeners.
* Improved Program Flow and Logical Thinking

References

* Google
* JAVA Docs
* JAVATPoint.com

Calulator