**Introduction of author:**

Hello, I am Karthik Reddy. I am currently pursuing Master of Sciences in Information and Technology at MSIT-IIITH. ------intro of project one line---- This video gives you the overview and implementation of my project- “**UNumber Calculator”**.

**Backstory of this UNumber Calculator:**

This calculator is mainly developed for the usage in the Mars Orbital mission which is based on Hoffmann Transfer. The main requirement is the calculator should be able to calculate the values even with error terms and units and get the resultant value with more precise significant digits. The integer calculator when used will not be able to find the result with error terms, significant digits and units. The main objective of this UNumber Calculator is to accomplish them avoiding errors like units’ mismatch.

**UNumber calculator:**

The UNumber calculator does the following operations:

1. It properly implements the basic calculators’ operations on UNumber with no compile errors or warnings.
2. It supports the use of Units, adjusts measured values for unit’s compatibility and shows error when not possible.
3. It works in a way that it supports the input, output and result with more significant digits than already taken.
4. In addition to basic operations the calculator also implements square root.

**UI of Calculator:**

This is the view of the calculator. The calculator user interface contains text fields for taking first operand and its error term, second operand and its error term, List view of units for selecting first and second operands units, Text boxes to show the measured value and error term of result along with units and also buttons for the operations required like addition, subtraction, multiplication, division and square root.

**Step by step implementation of the project:**

The project is done in a step by step manner in a disciplined way.

1. Firstly, the integer calculator with the basic operations like addition, subtraction, multiplication and division are implemented.
2. Next, the square root is implemented for integer calculator.
3. All the operations are then modified and implemented for double and long integer values.
4. Then the UNumber is introduced and code is improved in such a way that calculator works taking the UNumber, error terms and significant digits.
5. Then the final part of adding the units and their conversions to the calculator.

**Program Flow:**

The calculator runs when the main method in **Calculator.java** is invoked. The stage is set for the calculator here.

The complete user interface is taken care in the **Userinterface.java**. Labelling, creation of text fields, buttons, setting up the action for them with their respective functions is implemented in this.

For the units to be selected in the calculator we use the **Selectlist.java** in which we use select list form to display the units we use in the calculator. The units we use in this calculator and their conversions are defined, explained and validated in **Units.java**.

The code responsible for performing the calculator functions are present in **BusinessLogic.java**. this method deals with calculator values and performs actions on time. The input is taken from user interface as a string and is returned to it as a string.

The computations in the calculator are handled in **CalculatorValue.java**.

Measured value recogniser and error value recogniser-----

**Test driven Development:**

This calculator is implemented by test-driven development. We generally wait till the end of the project and then do the testing part. If errors present it may take a lot if time to rectify all over again. To overcome this, we have used test driven development. Each phase in the calculator has a test which when passed moved to next phase, else that part is improved and is routed to next step. This helps to have a clear picture of what we are doing and also consumes less time.

Here **TestBusinessLogic.java** and **TestCalculatorValue.java** serves the same purpose. Test cases are set up and performed to check all the possible edge cases. The operations and implementation are tested before hand, then the calculator is added with more modifications and implemented.

I will explain the working of the calculator with a basic example and then taking up a Hoffmann Transfer Equation.