**Class: BinToDec** **implements Conversion**

CONSTANT:

* VALUEDESC = “Binary”
* RESULTDESC=”Decimal”

Implement Interface:

* String getValue();
* String GetResult();
* Boolean isValid ();
* ArrayList<String> getProcessLog()

**GUI Form**

{

/\*get input string sValue

If (Binary-To-Decimal)

{

**BinToDec bd = new BinToDec(sValue);**

**If (db.isValid())**

**{/\*Display results\*/}**

}

Else if (Decimal-To-Binary)

{

**Dec2Bin db = new Dec2Bin (sValue,2);**

**If (db.isValid())**

**{/\*Display results\*/}**

}

Else if If (Decimal-To-Hexadecimal)

{

**Dec2Hex db = new Dec2Hex(sValue,16);**

**If (db.isValid())**

**{/\*Display results\*/}**

}

}

}

**Interface: Conversion**

* String getValue();
* String GetResult();
* Boolean isValid ();
* ArrayList<String> getProcessLog()

**Class: Dec2Num implements Conversion**

Member Variables:

* ArrayList<> lReminder;
* ArrayList<String> lProcessLog;
* String sInputValue;

Constructor:

* Public Dec2Num(String sValue, int iBase)

{ /\* valid sValue is a positive numbe\*/}

Methods:

* Private void Conversion()

{

/\*Convert sValue to a positive number \*/

/\* Use algorithm to get lReminder and lProcessLog \*/

/\* Calculate result sResult\*/ ??

}

* **Protected** ArrayList getReminders()

{ return lReminder;}

**Implement Interface:**

* String getValue() { return sValue;}
* String GetResult() {return null;} – need override in sub-class
* Boolean isValid(){/\*check whether input number is Valid to convert\*/
* ArrayList<String> getProcessLog()

{ return lProcessLog;}

**Class: Dec2Hex extends Dec2Num**

CONSTANT:

* VALUEDESC = “Decimal”
* RESULTDESC=” Hexadecimal”

Implement Interface:

* String GetResult();**( @Override)**

**Class: Dec2Bin extends Dec2Num**

CONSTANT:

* VALUEDESC = “Decimal”
* RESULTDESC=” Binary”

Implement Interface:

* String GetResult(); **( @Override)**