**Automated Code Compliance with Java Code**

**Khalsa College of Engineering and Technology, Punjab, Amritsar**

**Prof. H.S. Rai Prof. Bikramjit Singh Rahul Kumar Nanda**

**(B.Tech. C.S.E.)**

(nanda49rahul@gmail.com)

**Abstract**

Ongoing phase of designing, called Building Information modeling (BIM), has been emerged as an effective alternative for building modeling that has transformed the building industry into a digital form, and enhanced its productivity. As BIM software supports parameterized object that stores complete attribute information and their relation to other existing structural components. But still Code Compliance is performed separately to assure the structure constancy, behavior and immunity when concerns atmospheric conditions. Nowadays the Code Compliance Checking is accomplished manually, which is a problematic task and gives the error prone results.

Thus, in this paper a new approach called Automated Code Compliance Checking is adopted. Hence, a software is being created by using the java technology to apply various checks on the structural entities of the building. The checks applied are according to the Indian Standard (IS) Codes , which are formulized by higher authority or member of structural / civil engineer by analyzing the environmental conditions , seismic effect in the region, etc.

**Introduction**

Basically Automated Code Compliance is the Validation of the values fetched from the 3-D Modeling softwares like Archicad, Revit, etc. with the digitized standard values to give a result whether all the entities of the infrastructure are valid or not. It is more efficient than the code compliance. It saves lot of time and the manual work. The automated code compliance is error free.

The format which is used to carry the information about an infrastructure and is compatible with every BIM software is known as IFC. IFC stands for the Industry Foundation Classes. There is another format which is being derived from this basic format i.e. IFCXML. This format is same in structure with an XML file. It consists of the tags representing the entities of an infrastructure. It follows Parent-Child Hierarchy.

Java Technology is used for the Automated Code Compliance. The Data from the IFCXML format is being fetched by the Java Desktop Application and various checks according to Indian Standard Codes are being applied to generate the Report .

**Industry Foundation Classes** (IFC)

IFC is a standard format which is used to describe the BIM. It tells that how the information of the Infrastructure should be provided at all levels of the building lifecycle. It stores every entity of the infrastructure in the form of Object. It tells how the properties of an individual object can be described. Data can be holded by IFC for geometry, calculation, quantities, etc for different professions such as Architectural Engineer, Structural Engineer, Plumbical Engineer, Electrical Engineer.

The graphical notation used to create IFC schema is known as EXPRESS-G. It makes model development and reviewing easier. EXPRESS-G is international standard data definition language. Another version of IFC also exists in the form of XML known as ifcxml.

**Building Information Modeling** (BIM)

Traditional Building design was largely Reliant upon the 2 D Drawings resulting in poor coordination and loss of information. But with the introduction of BIM this problem has been solved upto maximum extent because BIM use 3 D Models instead of 2 D Drawings. BIM has changed the way of Planning, Building and Management of an Infrastructure. It is basically a platform on which all engineers can work together on the information of the infrastructure.

It was very difficult to work together by engineers before the introduction of the BIM. Earlier one engineer develops a model in one software and other in the other software. They were not able to communicate with each other. Then by the Introduction of the BIM , the BIM softwares came into existence . All the BIM softwares store the information of an infrastructure in the Standard Format known as IFC.

**BIM Softwares**

|  |  |
| --- | --- |
| BIM Softwares | Developer |
| Archicad | Graphicsoft |
| Revit | Autodesk |
| EIM | Archibus |
| Spirit | Softtech |
| Envisioneer | CADsoft |

Fig.1. various BIM softwares

**Research Methodology**

Earlier the Technique used for the Code Compliance was not automated. It was being performed manually. In that Case the Model of the Infrastructure was developed and was sent for the Code Compliance to the Code Officers. That method was too much Time Consuming and Error Prone. To Solve this problem the efficient Tools like BIM, AI, etc. are being used. In this case the Modeling work is performed 3-D Modeling Softwares such as Archicad, Revit, etc. Then the Code Compliance is done automatically by applying checks on the various entities of the Infrastructure as Beam, Column, Slab, Wall, etc. For the Automated Code Compliance the Indian Standard Codes are being used. Indian Standard (IS) Codes for validating the thickness, height, depth, etc of the entities of the infrastructure such as Beam, Column, Slab, Wall, etc.

ifcxml File

JVM(JAVA VIRTUAL MACHINE)

Parsing

Classes for Parsing

Report Generated

Fetching DOM Collection

Fetching a particular node from tree

Element Form

Automated Code Compliance

File Handling

DFD of Automated Code Compliance

Figure 2. Data Flow Diagram explaining the Automated Code Compliance.

Java Technology had been used for Code Compliance. As .ifc and .ifcXml File are the output of the 3-D Modeling Softwares like ArchiCAD, Revit, etc. This .ifcXml File is used as the input to the Java Desktop Application. Figure 3 shows the format of the .ifcXml file. .ifcxml file is being read in Java desktop application same as .xml file i.e. by parsing.

|  |
| --- |
| <?xml version="1.0" encoding="utf-8"?>  <doc:iso\_10303\_28 xmlns:exp="urn:oid:1.0.10303.28.2.1.1" xmlns:doc="urn:oid:1.0.10303.28.2.1.3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:oid:1.0.10303.28.2.1.1 ex.xsd" version="2.0">  <uos id="uos\_1" description="" configuration="i-ifc2x3" edo="" xmlns="http://www.iai-tech.org/ifcXML/IFC2x3/FINAL" xsi:schemaLocation="http://www.iai-tech.org/ifcXML/IFC2x3/FINAL ifc2x3.xsd">  <IfcPerson id="i1541">  <FamilyName>Undefined</FamilyName>  </IfcPerson>  <IfcOrganization id="i1543">  <Name>Undefined</Name>  </IfcOrganization>  <IfcPersonAndOrganization id="i1547">  <ThePerson>  <IfcPerson xsi:nil="true" ref="i1541"/>  </ThePerson>  <TheOrganization>  <IfcOrganization xsi:nil="true" ref="i1543"/>  </TheOrganization>  </IfcPersonAndOrganization>  <IfcOrganization id="i1548">  <Id>GS</Id>  <Name>Graphisoft</Name>  <Description>Graphisoft</Description>  </IfcOrganization> |

Figure 3. .ifcXml file as the output of BIM softwares

There are basically 3 parsing techniques:

* DOM Parsing
* SAX Parsing
* StAX Parsing

**DOM Parsing**

DOM stands for Document Object Module . It comes with the Java Development Kit ( JDK). JDK7 is being used for the automated code compliance. The DOM parser loads the complete file into the memory in the form of tree structure. Nodelist and Node are used to fetch the data from the .ifcXml file. Figure 4 shows the DOM parsing of an .ifcXml file.

**Nodelist**: It is basically an interface that provides abstraction of an ordered collection of the nodes without considering how the collection is being implemented. The items in the Nodelist can be accessed by an particular index.

**Node**: It is also an interface that is the primary datatype for entire Document Object Module. It represents a single node of the tree structure created by the Nodelist.

**Element**: The Element interface is used to represent an element in an HTML or XML document. Elements may consist of attributes which are associated to them. It basically used to access the attributes of an Xml tag. hasAttribute () and getAttribute () are two functions used to check the Attribute and Fetch the value of Attribute respectively

**Nodelist**: It is basically an interface that provides abstraction of an ordered collection of the nodes without considering how the collection is being implemented. The items in the Nodelist can be accessed by an particular index.

**DocumentObjectFactory**: It consists of the factory Application Programming Interface(API) that enables the parser to perform DOM parsing and form tree like structure.

**DocumentBuilder**: It defines the API that is used to obtain DOM document instances from an XML document.

**Document**: The Document is a container for text that is used to serve as the model for swing text components. Its main goal is to scale from very simple needs (a plain text textfield ) to complex needs (an HTML or XML document, for example).

**getNodevalue ()**: This Function is used to fetch a value of node of the tree structure generated by Nodelist.

**hasAttribute ()**: This Function is used to check whether a node of the nodelist is having the attribute which is being searched.

**SAX Parsing**

Another parsing method is the SAX parsing. It is completely different from the DOM parsing. In this the xml file is not completely loaded in the memory, instead it is being parsed line by line and different events are triggered on the occurrence of different elements such as opening tag, closing tag, character data, etc.

The Source file is used which is to be parsed and it is mandatory to register a handler which extends default handler class.

* startElement () - It triggers the event when start of the tag is encountered.
* endElement () – It triggers the event when end of the tag is encountered.
* characters () – It triggers the even when any text data is encountered.

**StAX Parsing**

StAX stands for Streaming API for Xml .Stax Parser is Different from DOM as SAX parser. SAX parser pushes the data but pulling of data from Xml is done in case of StAX. In the StAX parser the curser is maintained at the current position and extracts the content available at the cursor but SAX triggers a event when a data is encountered.

XMLInputFactory and XMLStreamReader, are used for the loading of the XML file.

Firstly a parent tag is searched in the .ifcxml File by DOM Parsing. Parent Tag in .ifcxml File is having an attribute “id”. Then the id of the Parent Tag is compared. Then its Child Tag is fetched by the Reference of the Parent Tag. As this we can read value of any Tag by following the Hierarchy of Parent and Child as shown in Figure 5.

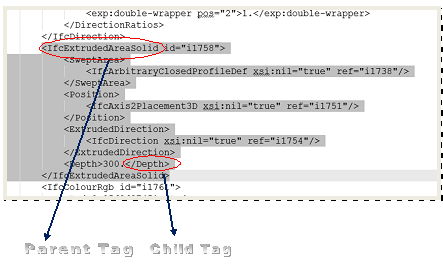


Figure 5. Hierarchy of .ifcxml File

**Steps to Access the Building Data from the .ifcXml file**

1. **Loading**

* Here the technique used for the parsing is DOM Parsing.
* The Parsing of .ifcxml File is done by Object of Document Class.
* The input file is completely loaded in the memory in the form of tree structure.

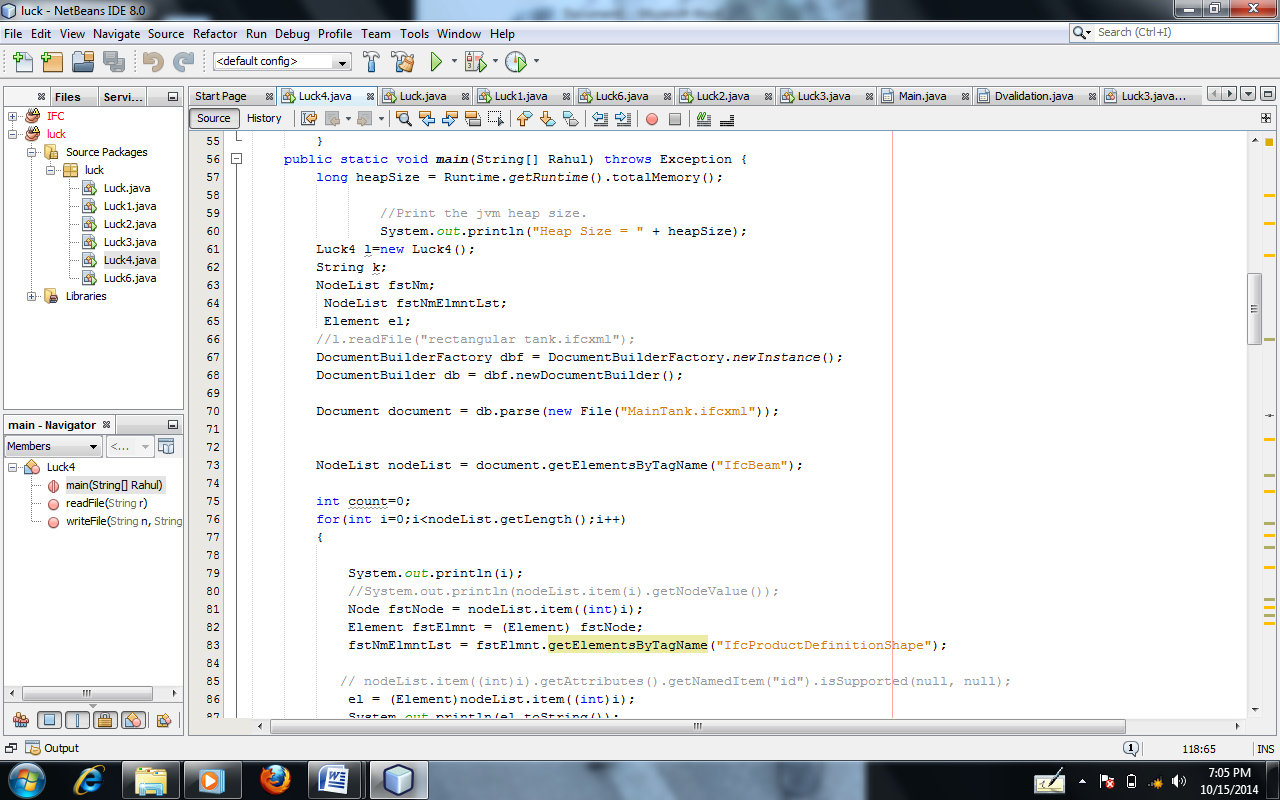


Figure 4. DOM Parsing of an .ifcXml File

2. **Reading**

* Then the NodeList interface is used to Search a Tag in the .ifcXml file as kwon the Figure 6.
* Basically NodeList is the Collection of Dom Elements.
* In the Figure 7, the node is used to get the particular leaf of the tree created by the Nodelist.

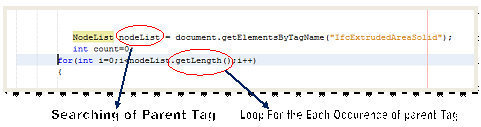


Figure 6. Nodelist and .getLength() function

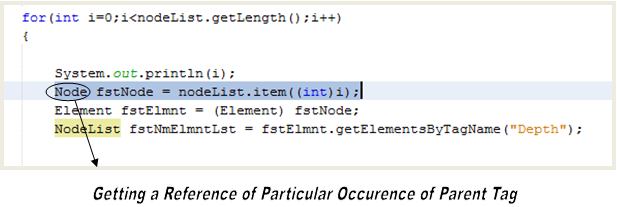


Figure 7. Node is used to get an leaf of the tree

3. **Processing**

* Then the value contained in the NodeList is converted to the Element Form as shown the Figure 8.
* It is Being Converted to Element Form as Some Functions Like getAttribute (), etc are to be applied on it.
* Then Figure 9 shows that the hasAttribute () function, is used to check whether the tag is having a particular attribute or not.

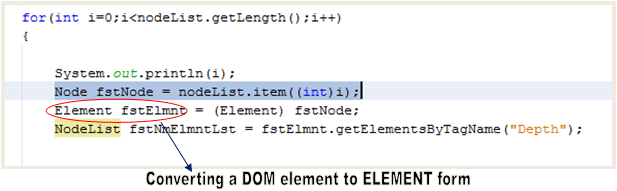


Figure 8. DOM element or the leaf node of the tree converted into Element form

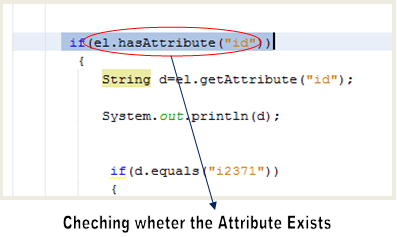


Figure 9. Implementation of the hasAttribute() function.

* Then the function getChildnodes() is used to check whether the parent tag is having a child tag or not. It is being represented in the Figure 10.

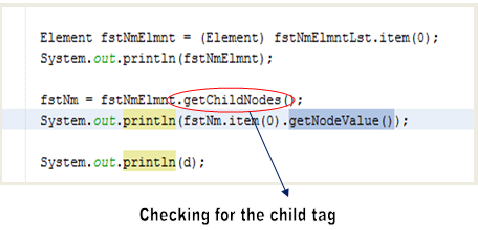


Figure 10. Checking for the child tag

* Then After getting the value of child tag, it is being trimmed as the last Character is “.” , which is being depicted in the Figure 11 .

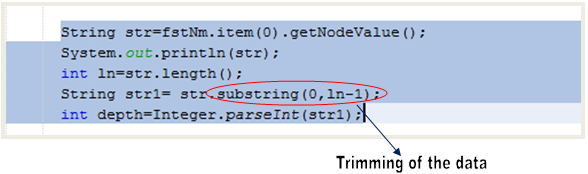


Figure 11. Trimming of the string with substring() function

* Trimming is being performed as the value is in string type and we want to convert it into the Numeric Value.
* Figure 12 shows that the string value is converted to numeric value by using parseInt() function.

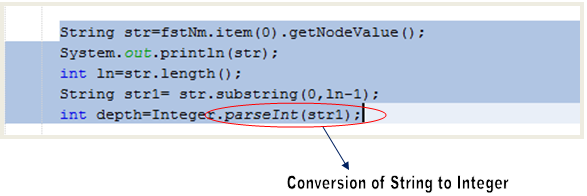


Figure 12. String to Integer conversion

4. **Checks Performed**

* The Code Compliance is done according to the IS (Indian Standard) Codes.
* IS Codes are the Standard Codes which are Accurate according to Indian Conditions for Construction.
* By these Codes we can Validate the Depth, Height, Diameter, etc of the entities used for construction.
* Checks Performed

1. Height of the Column
2. Thickness of Slab
3. Diameter of Reinforcement Bar
4. Depth of Beam

* At the End Limit is Applied According to IS Codes for Code Compliance.
* If the Value fetched from the .ifcxml file lies within the range then the entity of the Infrastructure is valid that means it could be Constructed else the Model needs some Modification at a Particular Entity of Infrastructure.

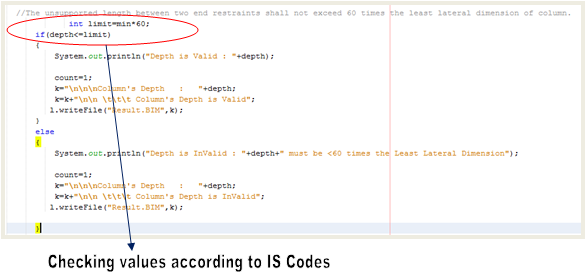
****

Figure 13. Check Applied according to Indian Standard(IS) Codes.

5. **Exception Handling**

Exception: It is an Run-Time Error.

* Null Pointer Exception

It is caused due to

* + - Calling an instance method on the object referred by a null reference.
    - Accessing or modifying an instance field of the object referred by a null reference.
    - If the reference type is an array type, taking the length of a null reference.
    - If the reference type is an array type, accessing or modifying the slots of a null reference.
    - If the reference type is a subtype of Throwable, throwing a null reference.
* Heap Space /Out of Memory Exception
  + - OutOfMemoryError in Java is a subclass **of java.lang.VirtualMachineError** and JVM throws java.lang.OutOfMemoryError when it ran out of memory in heap.
    - OutOfMemoryError in Java can come any time in heap mostly while you try to create an object and there is not enough space in heap to allocate that object.
    - [javavdoc of OutOfMemoryError](https://webproxy.vpnbook.com/browse.php?u=56z1lxrSzbWsava4LEa59VlLRXskKHwQtXvSkaik7pCsDy8NmQyQXWs%3D&b=0) is not very informative about this though.
* Number Format Exception.
  + - This is an unchecked exception and it can occur when you are trying to convert a String to a numeric value, like an Integer or a Float, but the String is not well formatted for the conversion.
    - For example if you are trying to parse an integer but the string is something like : “123.234” your conversion will fail with a NumberFormatException.

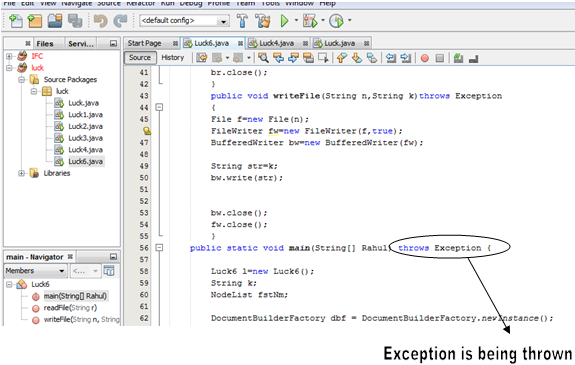


Figure 18: Exception Handling in JAVA technology

**Result** **and** **Discussion**

* If the child tag exists the .getNodevalue() function is used to get the value of the child tag as shown in Figure14.

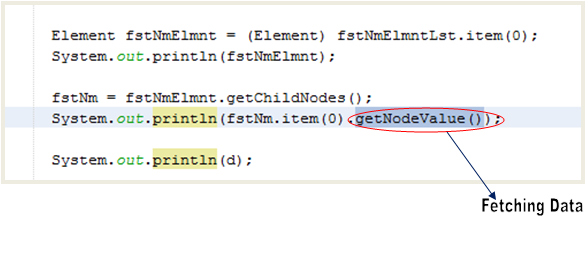


Figure 14. Fetching value from the child tag

* The concept of the File Handling is being used for the Reporting Purpose.
* The Checks are being performed on the values fetched from the ifcXml file , then if the data is according to the Indian Standard Codes then its status is written in the text file by file handling as shown in Figure 15.

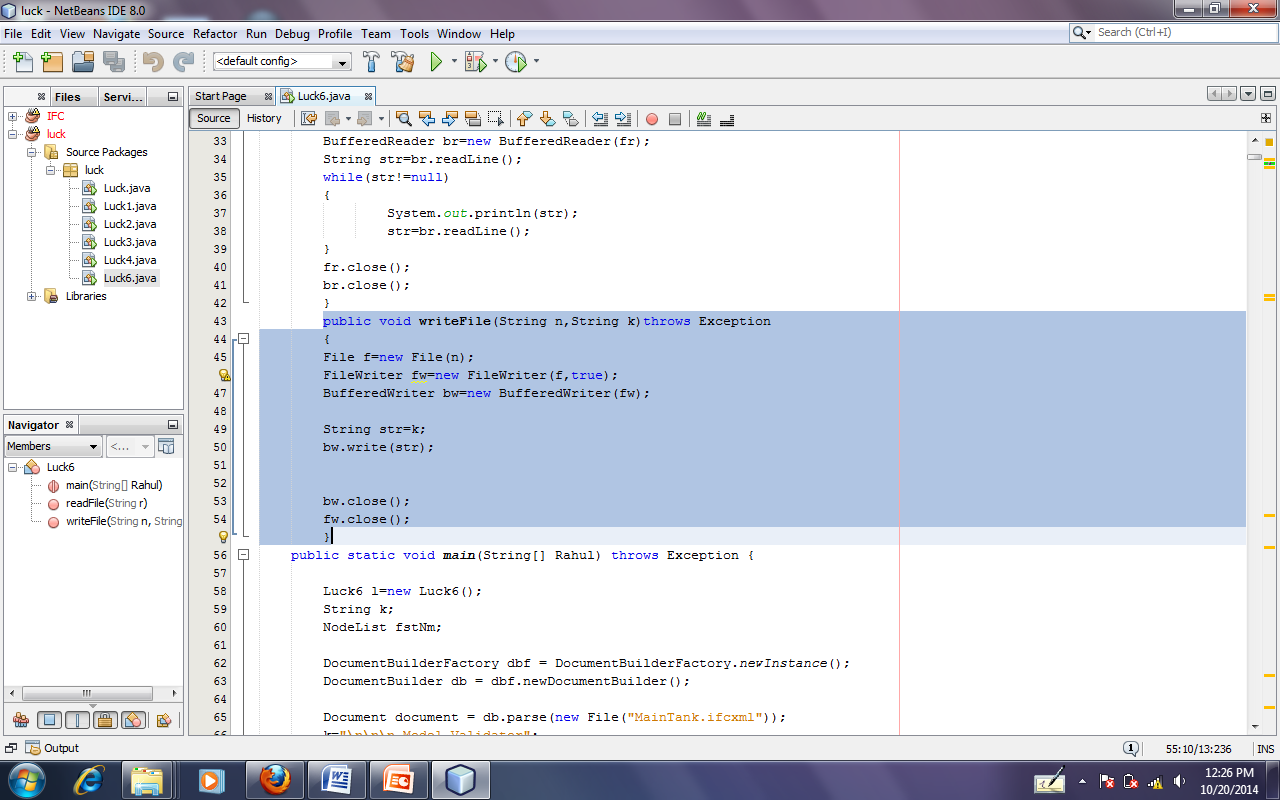


Figure 15. Function used for writing in the text file.

* Status here refers to the Valid/Invalid according to the Indian Standard (IS) Codes (Figure 17) .

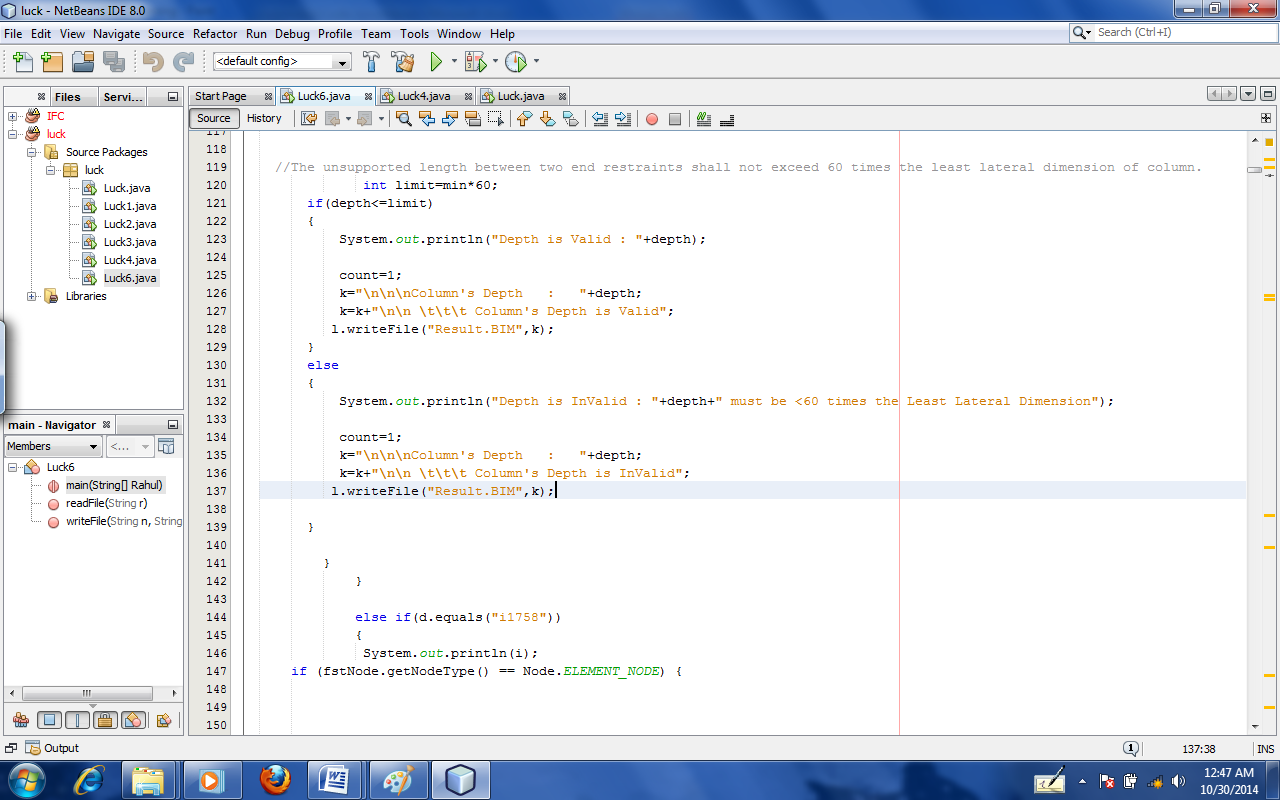


Figure 16. Generation of the Report

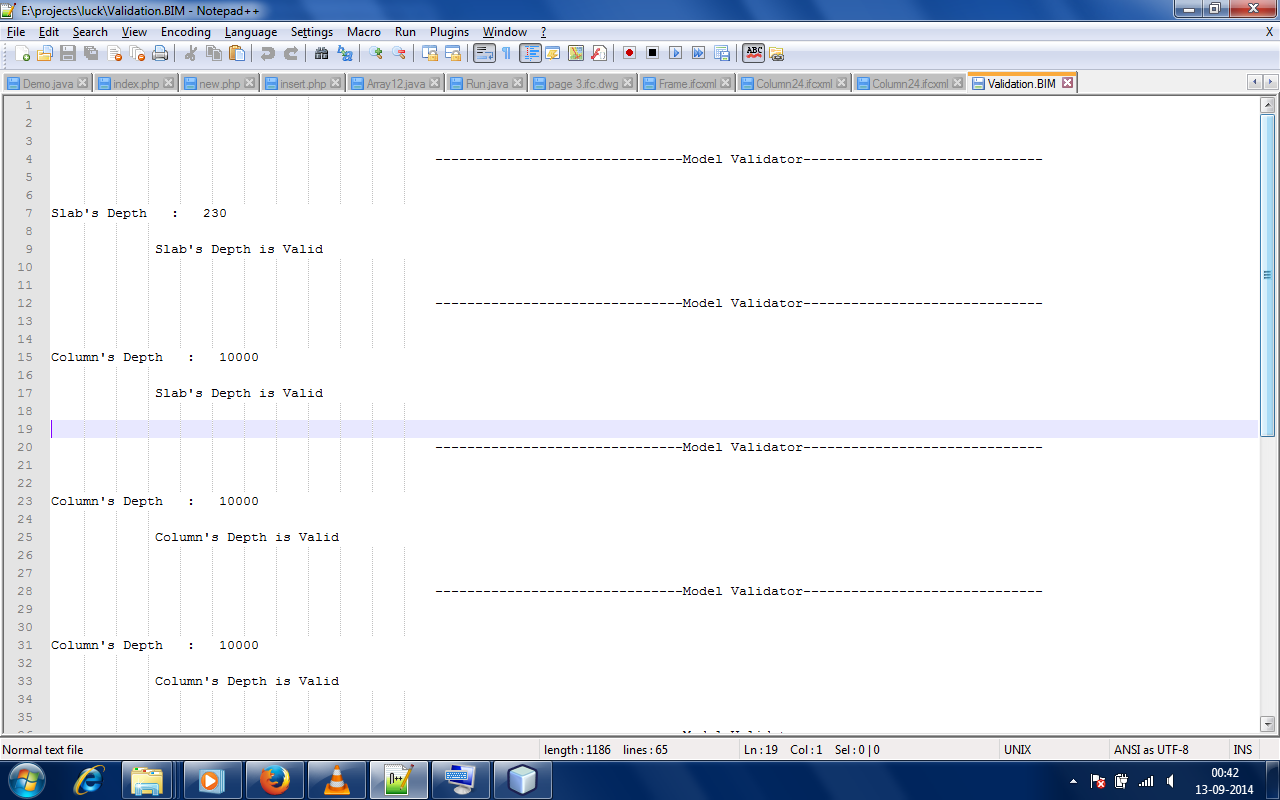


Figure 17. Graphical view of the Report generated

**Conclusion and Future Aspects**

Approach Followed follow is Automated Code Compliance. In this the check is being performed on the depth of Column according to IS (Indian Standard) Codes. Earlier no organization had worked upon the Indian Standard Codes. According to us, it’s the first time that the Indian Standard Codes are being digitalized. Automated code Compliance helps in saving Time and Manual efforts. The biggest advantage is that it gives error free results.

As this Automated Code Compliance is in the Developing stage so more checks could be applied according to the Indian Standard Codes. The Reporting graphics could be improved for graphical representation of the Results.

**References**

1. *Nawari. O. Nawari, Ph.D., P.E.* “ A FRAMEWORK FOR AUTOMATING CODES CONFORMANCE IN STRUCTURAL DOMAIN **”**
2. *Özgün Balaban, Elif Sezen Yağmur Kilimci, Gülen Çağdaş*3 “Automated Code Compliance Checking Model for Fire Egress Codes”
3. David Greenwood, Stephen Lockley Sagar Malsane, and Jane Matthews , “Automated compliance checking using building information Models”
4. http://blog.sanaulla.info/2013/05/23/parsing-xml-using-dom-sax-and-stax-parser-in-java.