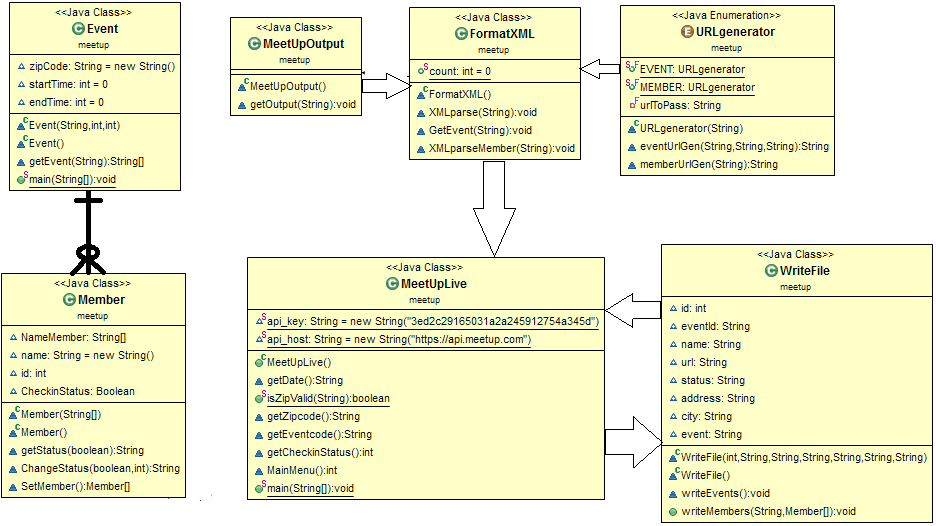
**Meetup Live**

Meetup Live is a java based project inspired by collaboration with meetup API. Meetup.com is a web application where events are hosted and manged. Our goal is to list the events of current date and show the details information about the events to the user. Also for each event, event members can check-in and the record are maintained in a text for future reference.

**Workflow**

1. Showing ‘Main Menu’ to select the choice of operation.
2. API requests to meetup.com with api\_key for today’s event. (input.txt)
3. Get the response into XML format and convert them to readable format and save then a text file. (event.txt)
4. Show the event list.
5. From the event list choose a event for with we want to check-in the members of the event.
6. Send API request to get the members of the event and write the XML format response to a file.File name (<eventidfromMeetup>.txt)
7. Get the members of the event and change the check-in status accordingly.
8. All updated member information are written into a file.File name (<evenntNumber>.txt).

**Class Diagram**



**Important Library and Features**

**Working with API**

* Java.net.URL
* java.net.URLConnection

**Enum**

An *enum type* is a special data type that enables for a variable to be a set of predefined constants.

public enum URLgenerator {

**EVENT**(MeetUpLive.api\_host+"/2/open\_events?format=xml&sign=true&photo-host=public&key="+MeetUpLive.api\_key),

**MEMBER**(MeetUpLive.api\_host+"/2/members?format=xml&sign=true&photo-host=public&key="+MeetUpLive.api\_key);

}

**XML parsing**

* javax.xml.parsers.DocumentBuilderFactory
* javax.xml.parsers.DocumentBuilder
* org.w3c.dom.Document
* org.w3c.dom.NodeList
* org.w3c.dom.Node
* org.w3c.dom.Element
* DocumentBuilderFactory: Defines a factory API that enables applications to obtain a parser that produces DOM object trees from XML documents.
* DocumentBuilder:Defines the API to obtain DOM Document instances from an XML document. Using this class, an application programmer can obtain a Document from XML.
* Document: The Document interface represents the entire HTML or XML document.
* NodeList:The NodeList interface provides the abstraction of an ordered collection of nodes, without defining or constraining how this collection is implemented.
* Node: The Node interface is the primary datatype for the entire Document Object Model. It represents a single node in the document tree.
* Element:The Element interface represents an element in an HTML or XML document.

**File read and Write**

* java.io.File
* java.io.Writer
* java.io.BufferedWriter
* java.io.BufferedReader