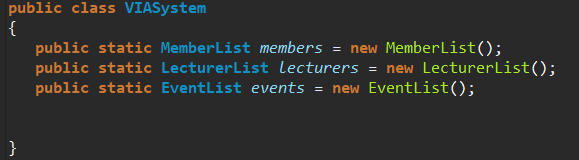
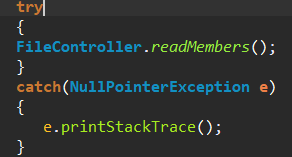
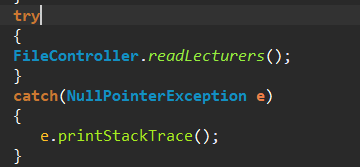
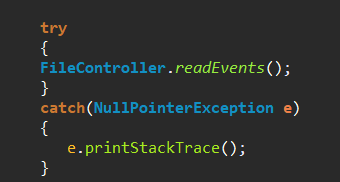
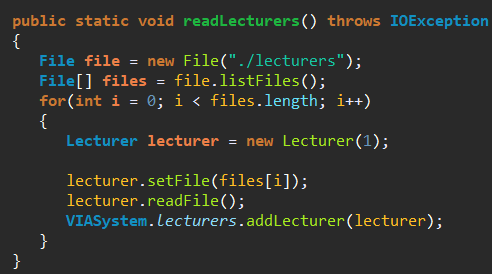
The system developed by our group consists of three different lists of objects that represent the events hosted by the company, the lecturers hired by them and the members of their organization.

The GUI for the system was created using a Java Swing GUI builder provided by the NetBeans IDE.

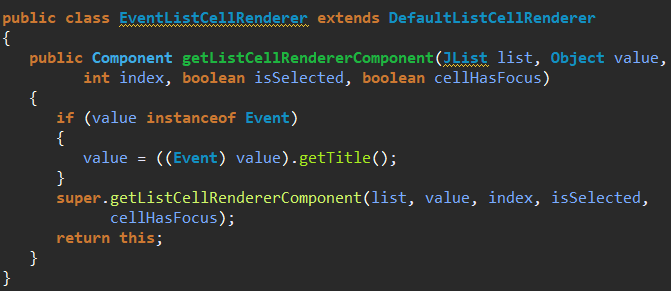
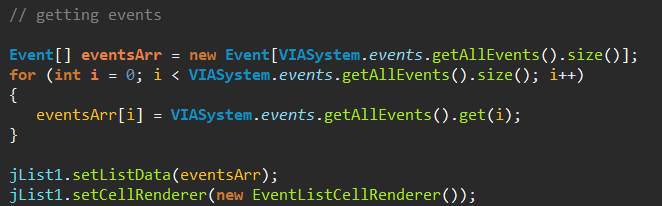
When the user runs the application the system will check to see if there are any files that store previous data and will read them in order to initialize the lists. The system will instantiate the required objects and use a FileController class to access and call the *readFile* methods for each *member, lecturer* or *event* object.



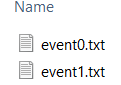
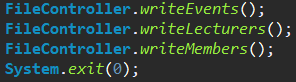
As stated before, the methods called on the FileController class instantiates as many objects as needed and then calls the required method for that object:



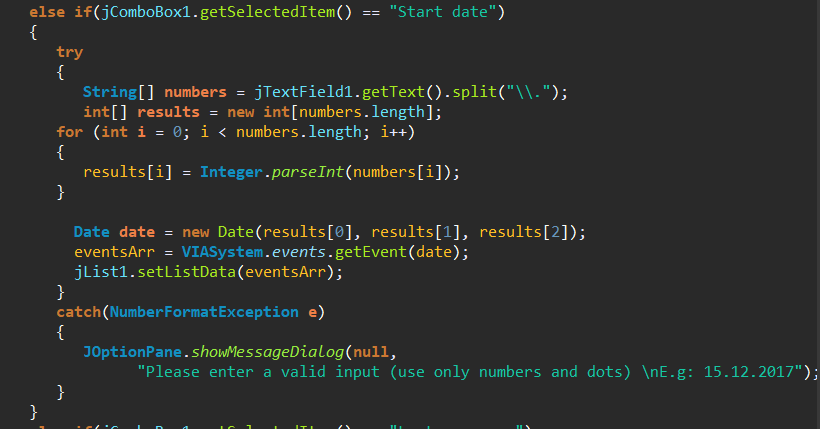
Then the user will be presented with three different choices: Events, Lecturers or Members. If he clicks on any of those buttons he will be taken to another frame of the UI that will display all the objects as a list. We used a *jList* to achieve this in which we stored objects and then used a custom made ListCellRenderer class in order to control what information will be displayed.



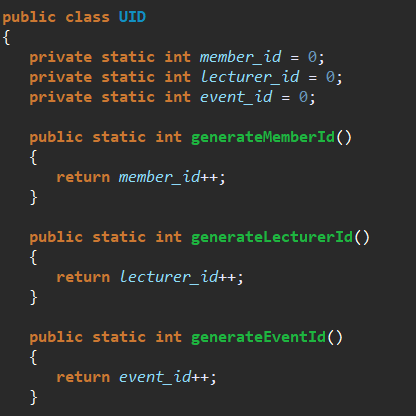
In order to create another event, the user only needs to click on the add button which will take him to a new frame in which he will have to select what type of event he wishes to create (lecture, seminar, workshop or trip). After clicking on one of the four buttons a new frame will open in which he will enter the data required to create that type of event (the *Lecture, Seminar, Workshop* and *Trip* classes are all children of the *Event* class). After finishing writing the date the system will instantiate a new object and add it to the *events* list. When the user chooses to close the system the FileController class will be used again to write everything that has been created up until now to files.



The system also allows the user to search the list by several parameters. Depending on what parameter the user chose the system uses specific code to search the list and display each element.



Furthermore, the system generates a unique ID for every event, lecturer or member and uses them in the naming of their files and for easily distinguishing each object from another.



An example for the usage of these IDs is a *member’s* list of attended events. Instead of creating a list of *event* objects the system instead stores the ID of the event. When creating a member for example the user chooses which events the member attended and the *member* object is being attributed and ArrayList consisiting only of the ID of each event.

