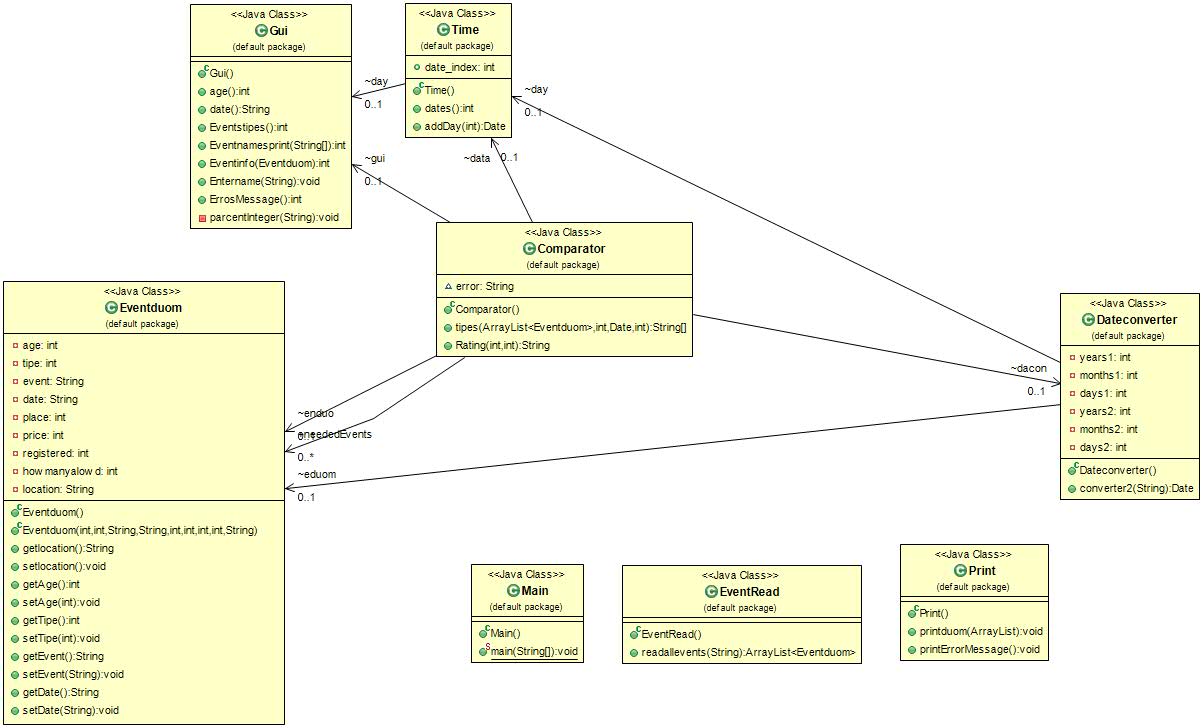
Project Report

1. Arnas Jonynas, Mantas Aušra, Tautvydas Urbaitis. 1-st group of Information Technologies. Topic number: 11.
2. This report is written to introduce a person to the program about events. It presents a short description of the program, shows the class diagriam in UML. Also, it explains the main methods, how they work and how they were implemented. Lastly, it gives infomation about how the project was planned and executed.
3. The program allows to look for events based on user‘s age, date and type of an event. Once chosen, a user can see information of an event and register to it. Changes made from Part 1 are added rating message and implemented input exceptions.
4. 
5. 1) Comparator - This class is responsible for which options should be displayed to user to choose from. It sorts out events that are not fitting for the date and age selected and also assigns rating for the event.

Dateconverter – Used to read date entered by user and converting it from String to Date format.

Eventduom – This class is used to create ArrayList of events.

EventRead – Used to read data from duom.txt file and put it into ArrayList

GUI – This class is responsible for creating graphical interface using JoptionPane.

Time – This class allows user to specify from which date to look for events.

Print – Used to print out all elements of ArrayList.

2) GUI methods – each of 6 primary GUI methods are responsible for a different dialog box and handling the answers provided to that specific choice given. These 6 methods together with comparator class are the core of this program. Comparator class with the help of these mothods collects all information needed for a user to register to an event.

Time methods – one method sets date to today, the other one sets it to a custom date specified by user.

3) GUI methods required the most time to write because of errors. Once written they took a lot of time to test inputs, analyze why certain inputs created errors and how to solve them. Because of this we had to rewrite some of the methods even a couple of times to find ones that did not generate any errors. Same problems arose with time methods, mostly because none of us have worked with a program requiring time before.

1. Execution of the project was a new experience, because no one in the group have worked on a similar project before. It allowed us to learn communicate and plan tasks in a small team. As with all projects, extra time would always be useful. A group of 3 members is large enough, however, it would have been easier and more efficient, if one of members had worked with GUI before this project.
2. The work was distributed a bit unevenly because programming skills of members of the group weren‘t equal. Arnas have done the more difficult part of the programming while Mantas and Tautvydas completed other smaller tasks. Better results would have been achieved if a more concrete plan had been prepared. We had outlined how we imagined our program, however, as the project continued, our plan changed quite a bit. The biggest problem with time management was that we kept delaying the work instead of distributing it evenly over the longer period of time.