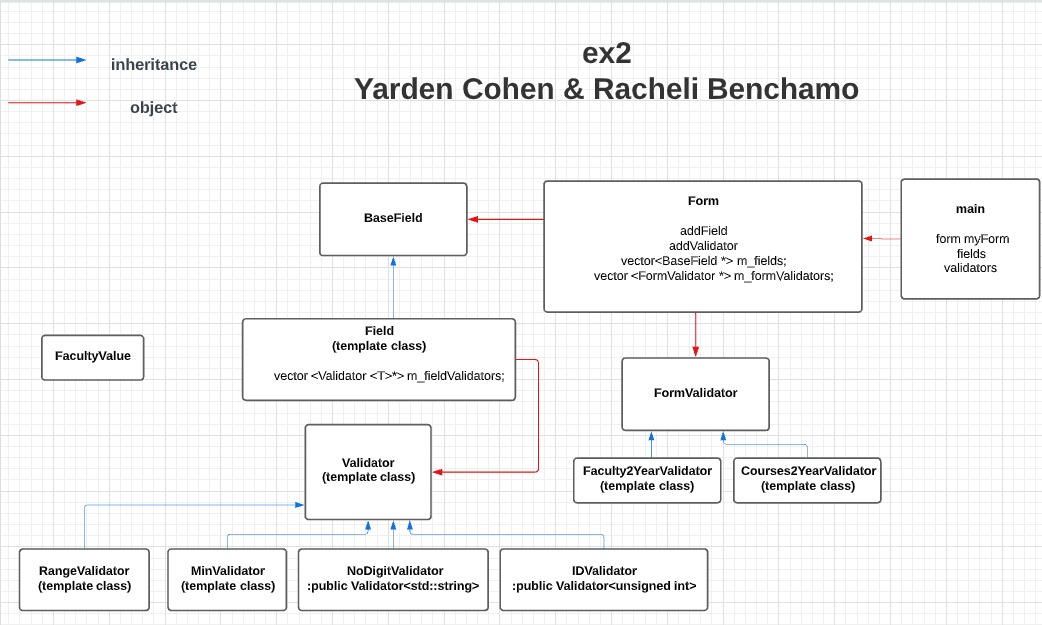
**README**

**ex2, oop2**

**Racheli Benchamo, ID: 209179738**

**Yarden Cohen, ID: 315177832**

**Design:**



**General Description**

In this exercise, a terminal application (Terminal, Console) was built, which is used to fill out a student registration form for certain studies year. The student enters a number of personal details as well as details of his / her planned studies for the spoken school year. The software verifies that the details are correct, including verification of the study details against the information known in advance about study options.

**General description of the program run:**

1.The program presents the user with field after field asking to fill in the required details. All fields are mandatory and if he has not filled them all, the program will wait.

2.Once the user has completed filling out the form, the program checks the integrity of the fields.

3.If there are invalid fields, the program will notify the user, displaying all the fields entered.

When next to the problem fields an appropriate error appears) or after the list of fields, if the problem is

Respectively between fields (. Now the program will ask the user to re-enter the information similar to section 1,

This time, however, she will only request it for the defective fields, and will return to section 2.

4. If all the fields are valid, the program will display a notification thanking you for the registration, display the information you entered, and exit.

**List of fields and legal information types for them:**

1. Name - Cannot contain digits. There is no need to check more than that, i.e. can appear as hyphens, commas

and so'.

2 .I.D. - Verified according to the review digit.

3. Year of birth - the minimum age for registration is 16. Regarding the maximum age, our working assumption is that a person over the age of 100 is not going to enroll in an academy and that is probably a mistake.

4. Faculty - The user will enter a number between 1 and 3 that represents the following classes:

1-Computer Science

2-Medicine

3-Literature

5. School year - We assume that in the rigid academic institution and our fixed do not allow a different study layout from the official track, so we can verify the value against the fixed number of years for each track: 4 for computer science, 7 for medicine and 3 for literature.

6. Number of courses in this year –

Although the number of years is fixed with us,

But there is little flexibility in planning the course layout so not necessarily everyone takes the same courses in the particular year.

Here the number is determined only by the school year (without any difference between the classes).

In the first two years you can take between two and 6 courses, in years 3-4 you can take between two courses to 10, and from year 5 onwards you can take between two courses and 8

**Description of the classes:**

**Main** – the main function**.**

**Form –** hold vector of base field that will contain the all kinds of field

required in the form.

Also contain vector of formvalidator that each one will be responsible for checking the integrity of a specific field.

**FormValidator** - responsible for checking the integrity of the form.

**Faculty2YearValidator -** Checks whether the year the user entered corresponds to the number of years in the faculty he is in.

**Courses2YearValidator –** Checks whether the number of courses the user has entered corresponds to the student's current academic year.

**Base** **field** – base class that will help us to hold vector of field in the form, we have to create this base class because field is u template class and we can't hold vector of a template.

**Field** - once we need a new field we will create it by the template class with the right type.

**Validator** – this is a base class to all kind of validator will check the integrity of a specific minute in the form.

**RangeValidator** – this is a template class that you will be responsible for checking the integrity of a particular value in a given range.

**MinValidator** - checks if a value obtained is greater than a predetermined minimum number.

**NoDigirValidator** - responsible for checking that the name of the form filler was not entered with numbers.

**IDValidator** - responsible for the correctness of an identity card by checking the correctness of the audit digit.

**FacultyValue -** A department of the faculty where the student is studying.

**Other:** there are no mistakes we are aware of.