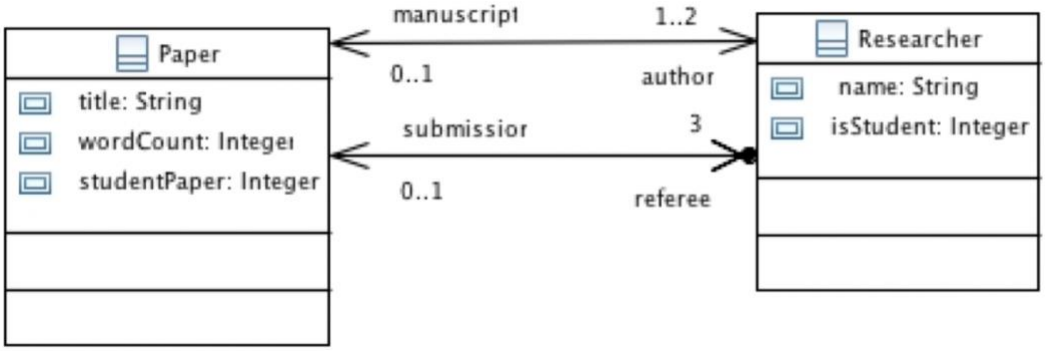
Deadline:. 2/6/2023

Work submission

Submit this assignment in groups. The group leader has to submit the work (in both DOCX and PDF formats + java code for Q2). The submitted files must be named swe312-Spring22-<group number>. Late submission will not be accepted.



Question 1

Consider the following class diagram that models a registration system.

Express the following constraints in OCL.

1. **Invariants**

   (a) A researcher must submit one manuscript.

   (b) A research must be assigned one submission

(c) A paper must have at most 10000 words

(d) Students are not allowed to review any paper.

Question 2

In the lecture slides, we have seen how the Z notation can be used to specify software modules. Here you will use Z notation to implement a java program from the specifications.

Please submit the answers as a code with screenshots of input and output in a word document (.docx), as well as the source code as java program (.java).

Implementing the birthday book

In your implementation, you choose to represent the birthday book with two arrays, which might be declared by

names : array [1 ..] of *NAME*;

dates : array [1 ..] of *DATE*;

The Figure below describe the schema of the program. The variable *hwm* (for 'high water mark') shows how much of the arrays is in use.

$$
\begin{array}{|l}
\hline
BirthdayBook1 \\
names : \mathbb{N}_1 \longrightarrow NAME \\
dates : \mathbb{N}_1 \longrightarrow DATE \\
hwm : \mathbb{N} \\
\hline
\forall\, i, j : 1 \,..\, hwm \bullet \\
\quad i \neq j \Rightarrow names(i) \neq names(j) \\
\hline
\end{array}
$$

A) First operation is *Add Birthday*, to add a new name, we increase *hwm* by one, and fill in the name and date in the arrays. Write a java code for this operation.

AddBirthday 1

$\Delta BirthdayBook\ 1$

$name? : NAME$

$date? : DATE$

$\forall i : 1 .. hwm \bullet name? = names(i)$

$hwm^0 = hwm + 1$

$names^0 = names \oplus \{hwm^0 \rightarrow name?\}$

$dates^0 = dates \oplus \{hwm^0 \rightarrow date?\}$

B) The second operation is *FindBirthday*:

FindBirthday 1

$\Xi BirthdayBook\ 1$

$name? : NAME$

$date! : DATE$

$\exists i : 1 .. hwm \bullet$
$\quad name? = names(i) \land date! = dates(i)$

Hence: The predicate says that there is an index *i* at which the *names* array contains the input *name*?, and the output *date*! is the corresponding element of the array *dates*. For this to be possible, *name*? must in fact appear somewhere in the array *names*: this is the pre-condition of the operation. Write a java code for this operation.

C) The third operation is *Remind*: it produces as outputs *cardlist*

*Remind*1

$\Xi BirthdayBook1$

$today? : DATE$

$cardlist! : \mathbb{N}_1 \longrightarrow NAME$

$ncards! : \mathbb{N}$

$\{ i : 1 .. ncards! \bullet cardlist!(i) \}$
$\quad = \{ j : 1 .. hwm \mid dates(j) = today? \bullet names(j) \}$

The set on the right-hand side of the equation contains all the names in the *names* array for which the corresponding entry in the *dates* array is *today*?. The program code for *Remind* uses a loop to examine the entries one by one. Write a java code for this operation.