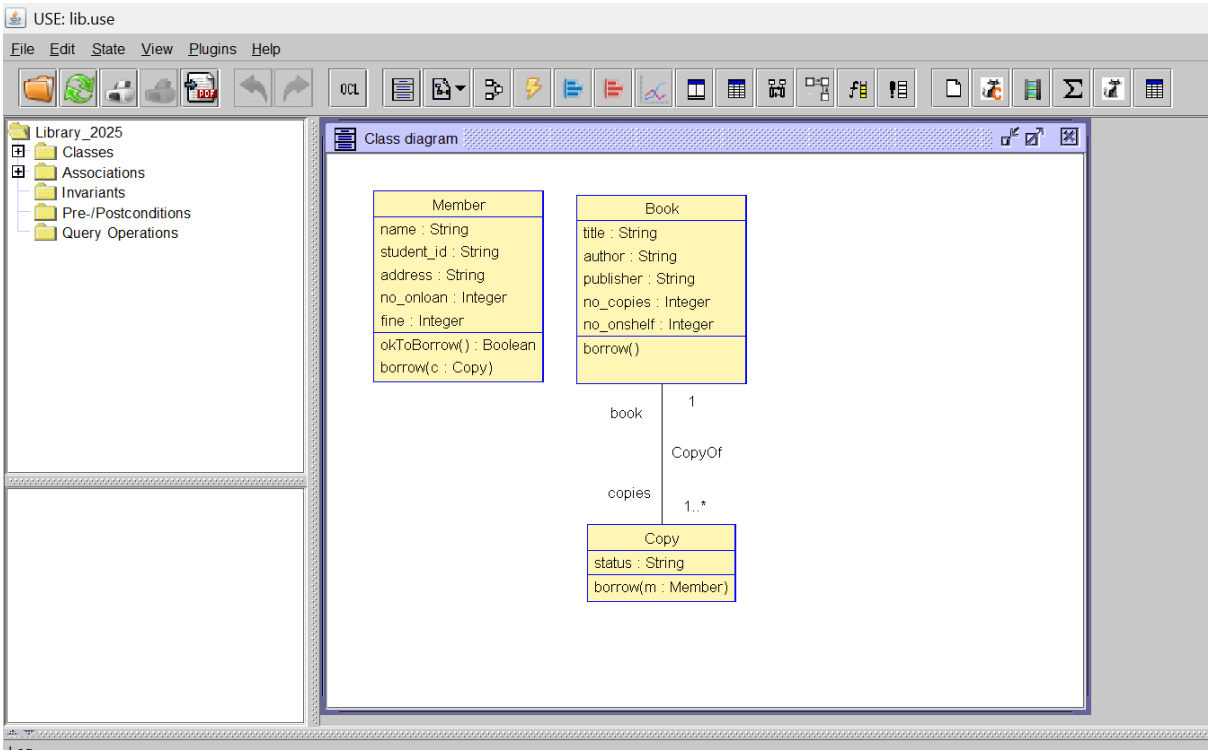
USE Lab Test CA - C24329646 - Software Engineering 1

Name: Lívia Ferreira Guimarães Neves
Student Number: C24329646
Year: TU858/2

0. Load the sample USE code in lib.use and open class diagram view. You should see something like below on rearranging. Save your layout as lib_default.clt, it will be useful later when reloading USE code.

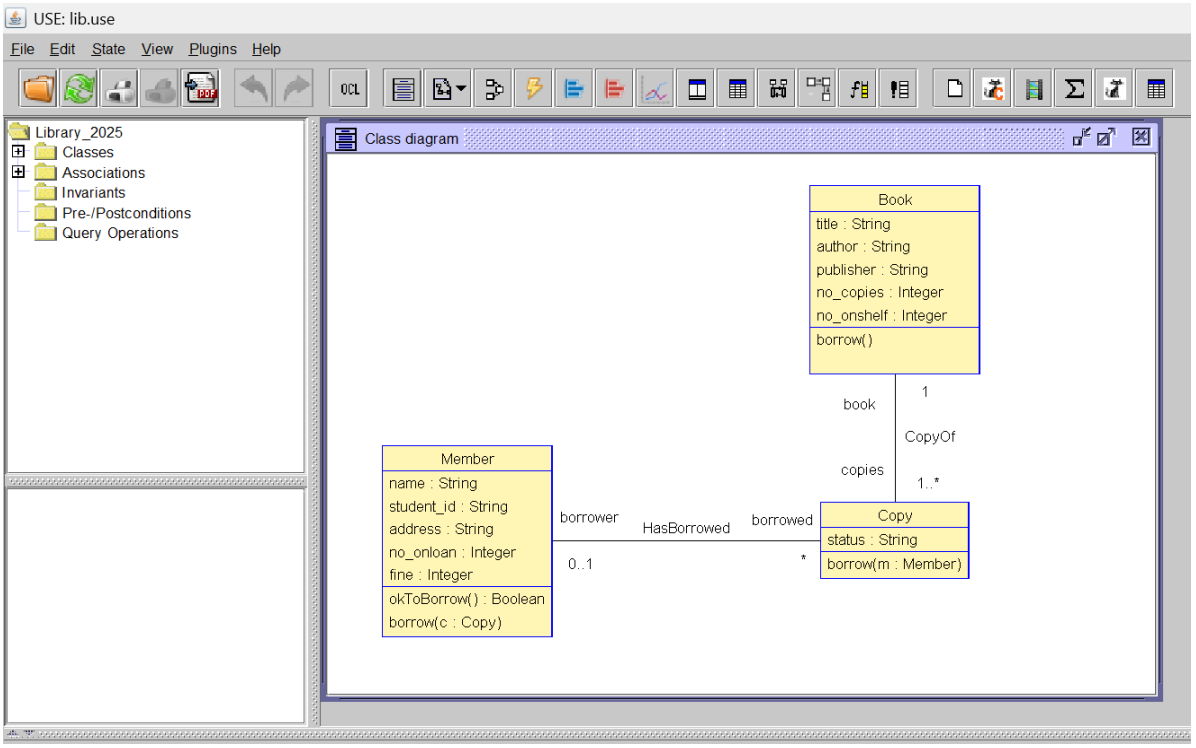


1. Modify the USE code so that class diagram looks like below and copy and paste it into a Word document report.docx.

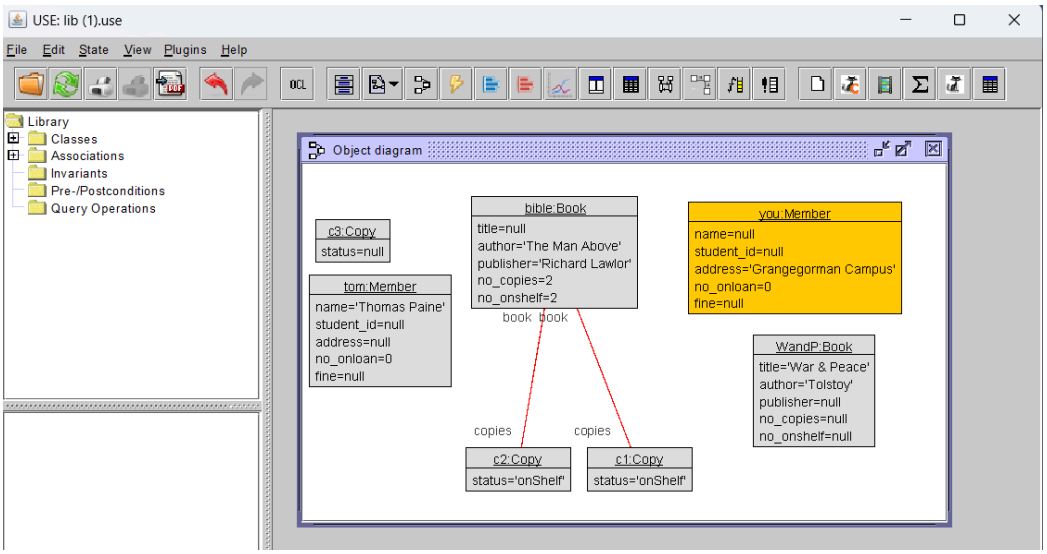
CODE “lib.use”:

```
model Library_2025
-- Nov 28, 2025
class Book
  attributes
    title : String
    author : String
    publisher : String
    no_copies : Integer
    no_onshelf : Integer
  operations
    borrow()

end

class Copy
  attributes
    status : String
  operations
    borrow( m : Member)

end

class Member
  attributes
    name : String
    student_id : String
    address : String
    no_onloan : Integer
    fine : Integer
  operations
```

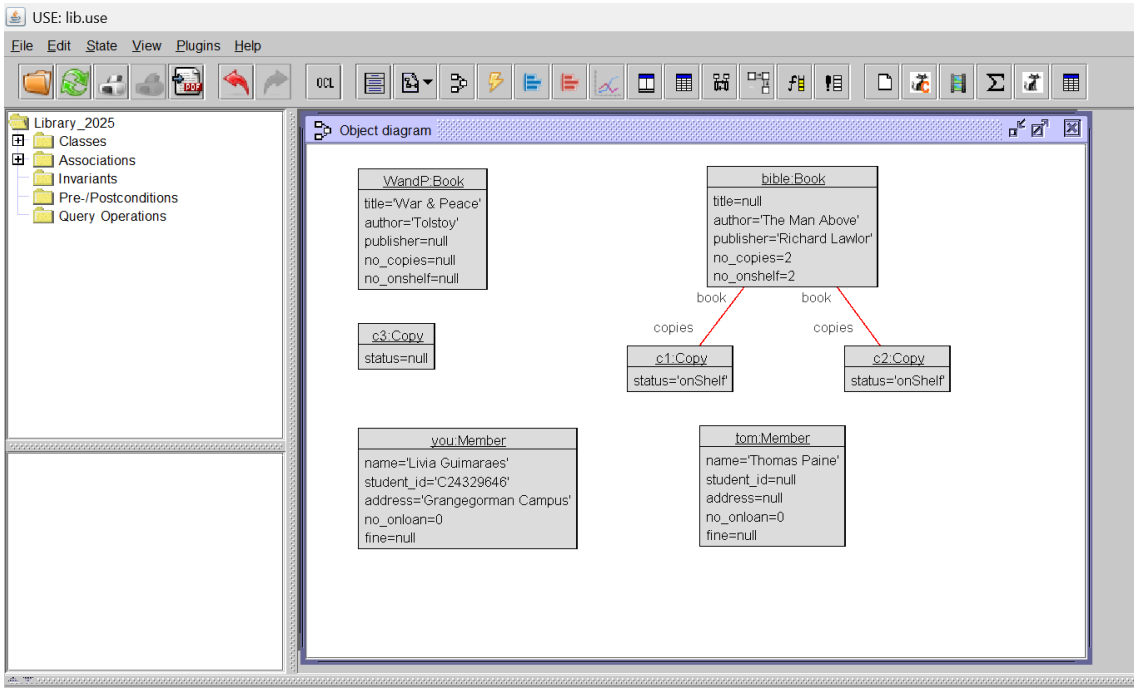
```
    okToBorrow() : Boolean   -- returns true or false

    borrow(c : Copy)
    begin
        declare ok : Boolean;
        -- ok := self.okToBorrow();
        -- missing code
    end
end



-- associations
association CopyOf between
    Copy[1..*] role copies
    Book[1] role book
end

constraints
-- 20 marks in total (2025)
-- changes/modifications in the code:
association HasBorrowed between
    Member[0..1] role borrower
            Copy[*] role borrowed
end
```

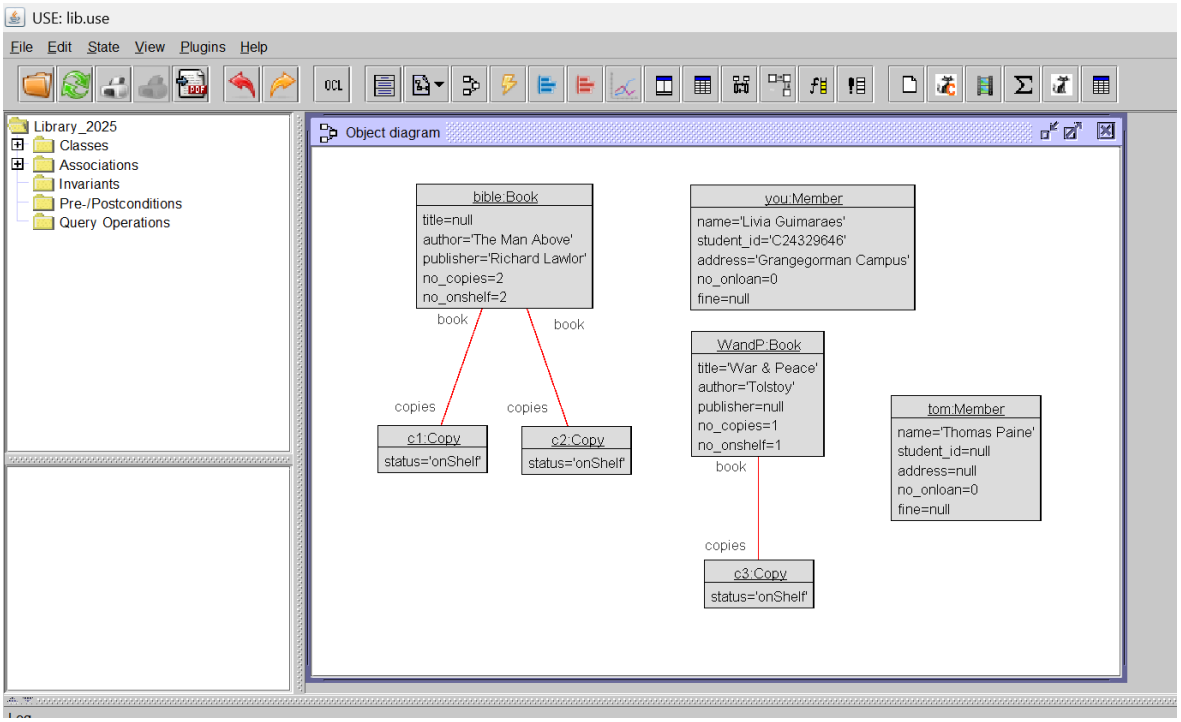

USE: lib.use
File Edit State View Plugins Help
Library_2025
Classes
Associations
Invariants
Pre-/Postconditions
Query Operations
Class diagram
Book
title : String
author : String
publisher : String
no_copies : Integer
no_onshelf : Integer
borrow()
book
1
CopyOf
copies
1..*
Member
name : String
student_id : String
address : String
no_onloan : Integer
fine : Integer
okToBorrow() : Boolean
borrow(c : Copy)
borrower
HasBorrowed
borrowed
0..1
*
Copy
status : String
borrow(m : Member)

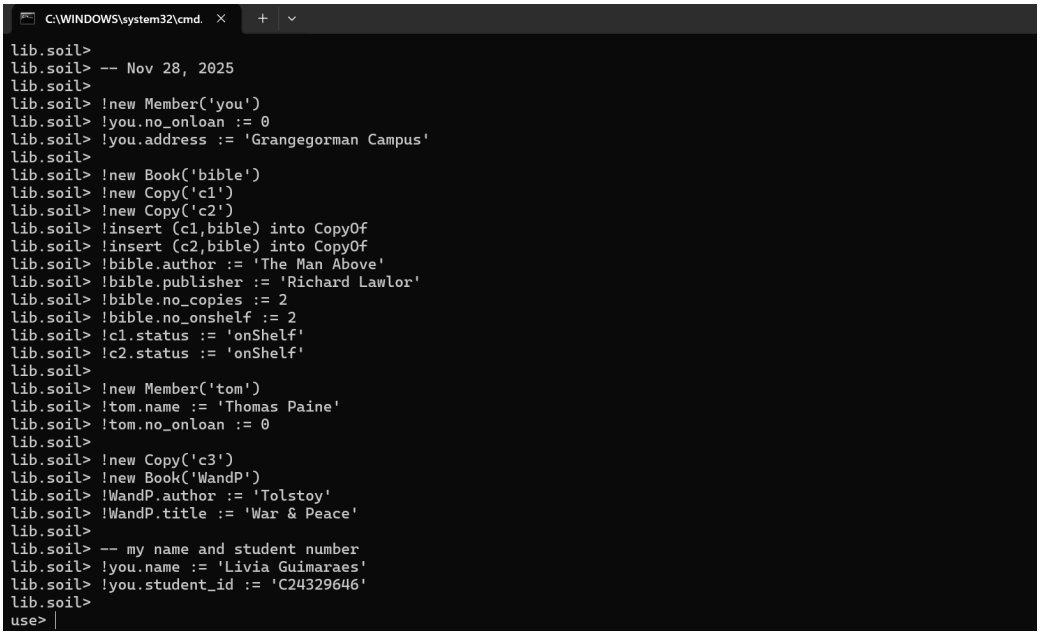
2. Load the test objects provided in lib.soil, open an object diagram view and reorganise the objects to look like (and save the layout as lib.olt) :

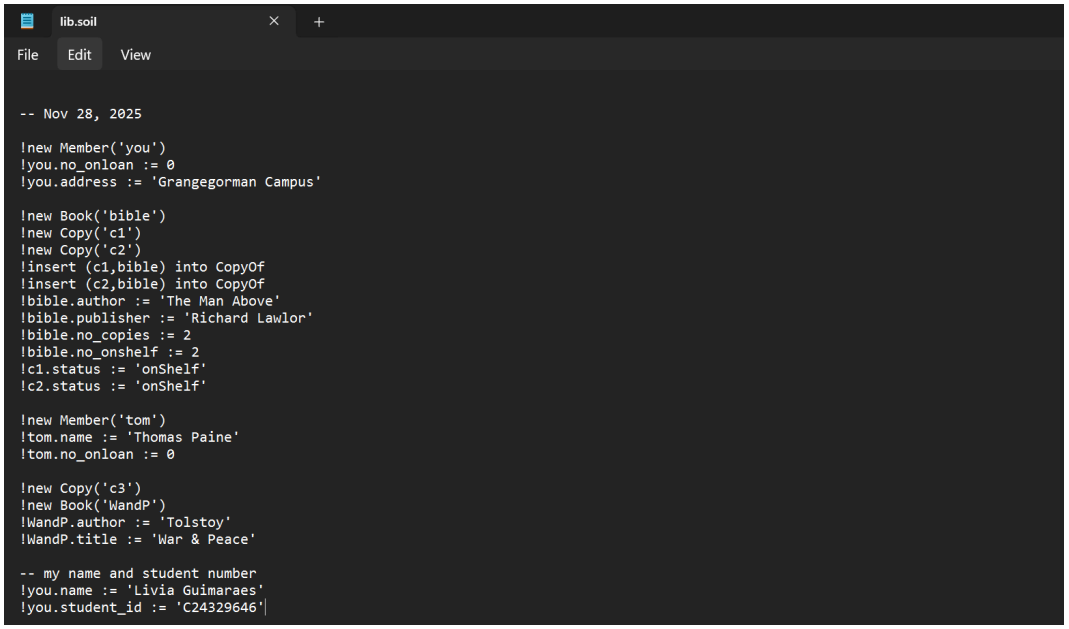


3. Put your name and student number as attributes into the Member object you (two ways to do this). Make c3 a copy of the Book object WandP and update in an appropriate way the attributes no_copies, no_onshelf in WandP; and the status attribute in c3. Save your you updated object model to the SOIL file lib.soil and copy/paste the new object diagram to your report.

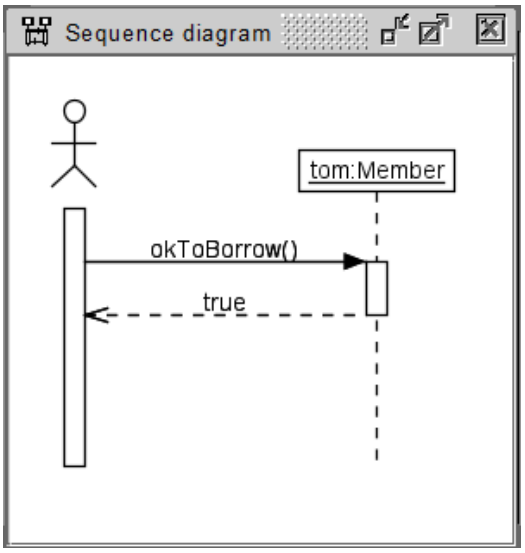


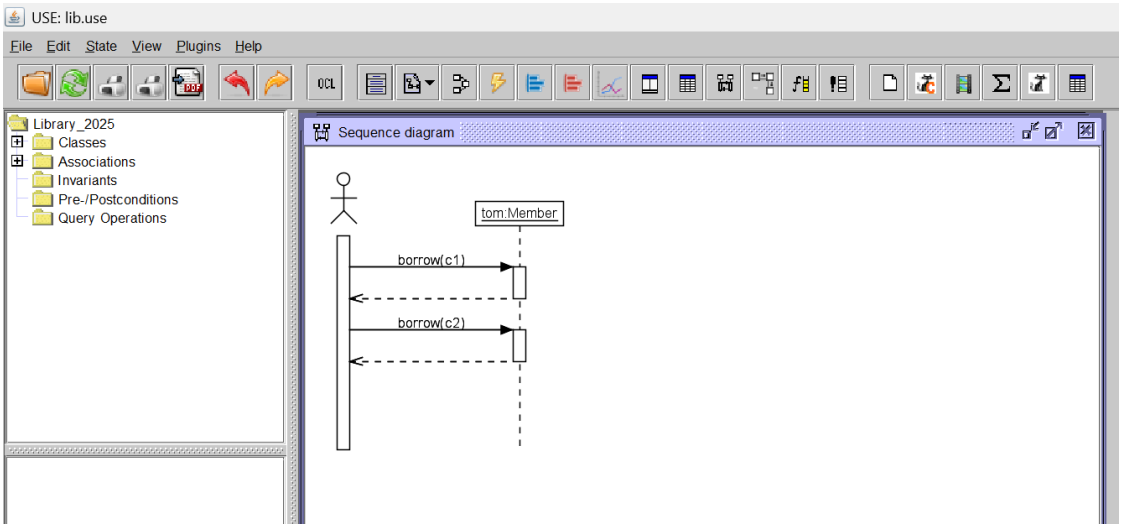



C:\WINDOWS\system32\cmd.
lib.soil>
lib.soil> -- Nov 28, 2025
lib.soil>
lib.soil> !new Member('you')
lib.soil> !you.no_onloan := 0
lib.soil> !you.address := 'Grangegorman Campus'
lib.soil>
lib.soil> !new Book('bible')
lib.soil> !new Copy('c1')
lib.soil> !new Copy('c2')
lib.soil> !insert (c1,bible) into CopyOf
lib.soil> !insert (c2,bible) into CopyOf
lib.soil> !bible.author := 'The Man Above'
lib.soil> !bible.publisher := 'Richard Lawlor'
lib.soil> !bible.no_copies := 2
lib.soil> !bible.no_onshelf := 2
lib.soil> !c1.status := 'onShelf'
lib.soil> !c2.status := 'onShelf'
lib.soil>
lib.soil> !new Member('tom')
lib.soil> !tom.name := 'Thomas Paine'
lib.soil> !tom.no_onloan := 0
lib.soil>
lib.soil> !new Copy('c3')
lib.soil> !new Book('WandP')
lib.soil> !WandP.author := 'Tolstoy'
lib.soil> !WandP.title := 'War & Peace'
lib.soil>
lib.soil> -- my name and student number
lib.soil> !you.name := 'Livia Guimaraes'
lib.soil> !you.student_id := 'C24329646'
lib.soil>
use>


lib.soil
File
Edit
View
-- Nov 28, 2025
!new Member('you')
!you.no_onloan := 0
!you.address := 'Grangegorman Campus'
!new Book('bible')
!new Copy('c1')
!new Copy('c2')
!insert (c1,bible) into CopyOf
!insert (c2,bible) into CopyOf
!bible.author := 'The Man Above'
!bible.publisher := 'Richard Lawlor'
!bible.no_copies := 2
!bible.no_onshelf := 2
!c1.status := 'onShelf'
!c2.status := 'onShelf'
!new Member('tom')
!tom.name := 'Thomas Paine'
!tom.no_onloan := 0
!new Copy('c3')
!new Book('WandP')
!WandP.author := 'Tolstoy'
!WandP.title := 'War & Peace'
-- my name and student number
!you.name := 'Livia Guimaraes'
!you.student_id := 'C24329646'

4. Implement the Member operation okToBorrow(). It should return either true or false. It is ok for a member to borrow if no_onloan < 2, otherwise it is not. Then reload your USE code and the test objects and run this operation on tom. Open a sequence diagram view and copy/paste the diagram to your report.





5. Next implement the three borrow() operations. Recall the sequence diagrams you demoed in StarUML. One borrow() sends a borrow() message to the next object and so on.

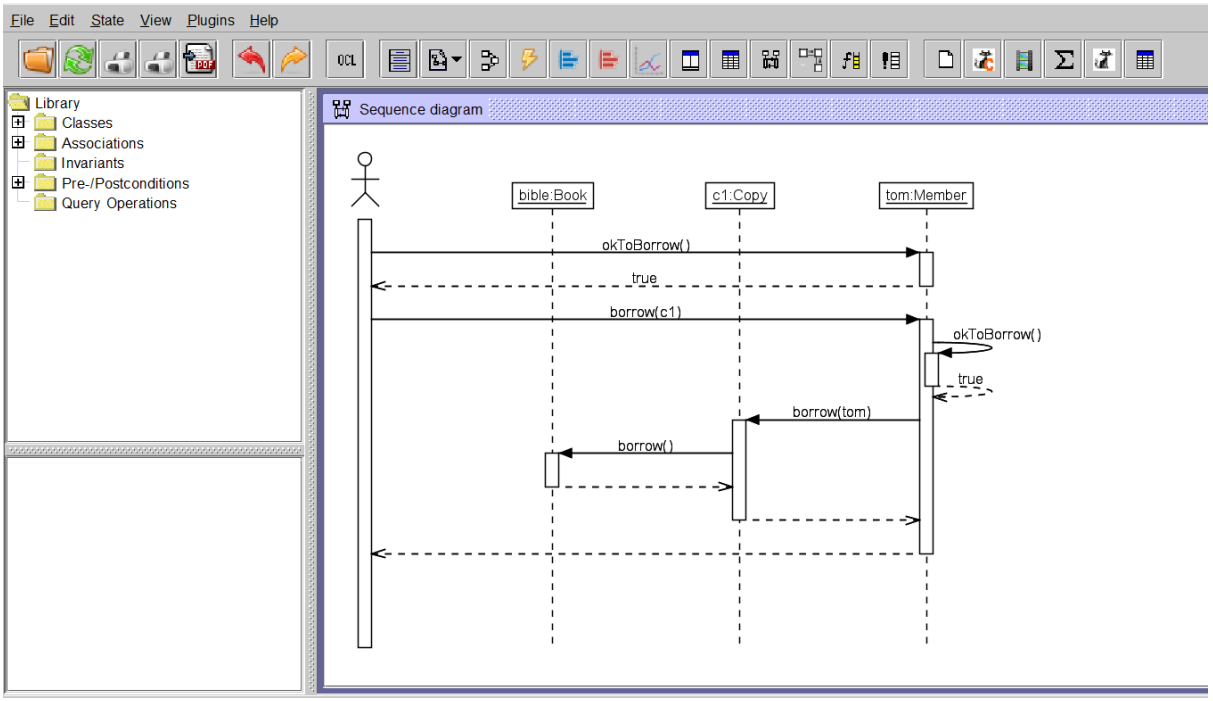
• Member borrow(c:Copy). It should first check that it's ok for the member to borrow and if it is, it updates no_onloan attribute and in turn sends a borrow() message to c.

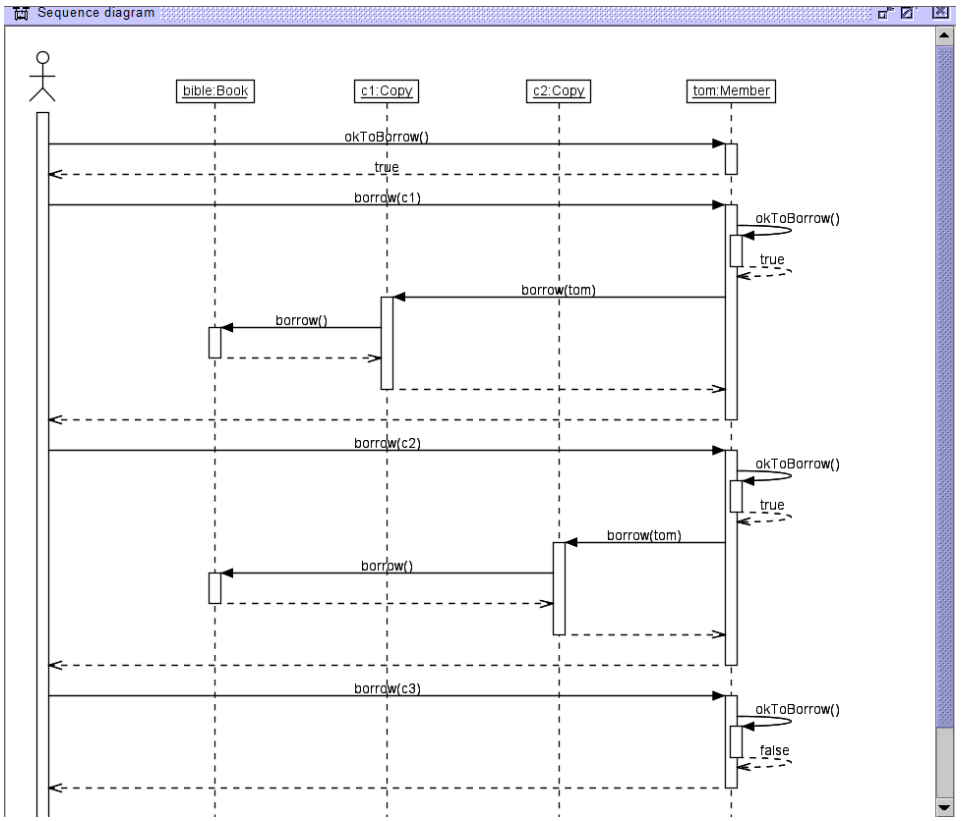
• Copy borrow(m:Member)

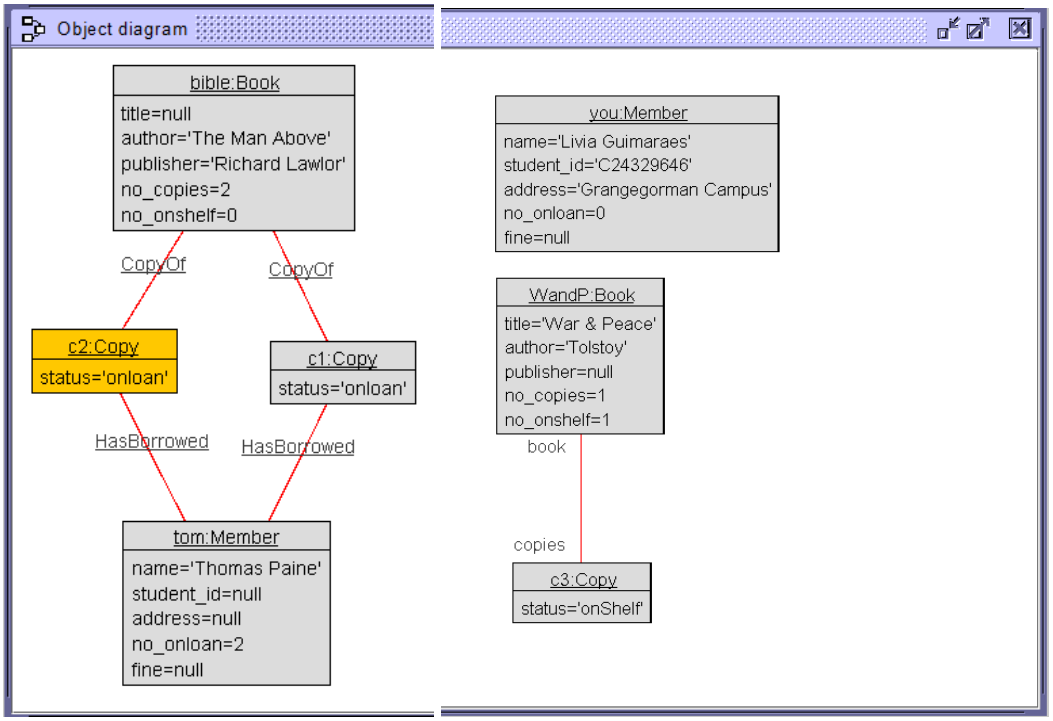
• Book borrow()

Each borrow() operation should update its own object's attributes and should result in the objects being linked.

6. Once implemented, reload your USE model and the objects and try to get tom to borrow c1 , c2, and c3. Your resulting object diagram should result in something like below. If it works, save your updated object model to lib.soil. Open a sequence diagram view, copy/paste it to the report. Also copy/paste your object diagram to the report




Sequence diagram
bible:Book
c1:Copy
c2:Copy
tom:Member
okToBorrow()
true
borrow(c1)
okToBorrow()
true
borrow(tom)
borrow()
borrow(c2)
okToBorrow()
true
borrow(tom)
borrow()
borrow(c3)
okToBorrow()
false


Object diagram
bible:Book
title=null
author='The Man Above'
publisher='Richard Lawlor'
no_copies=2
no_onshelf=0
CopyOf
CopyOf
c2:Copy
status='onloan'
c1:Copy
status='onloan'
HasBorrowed
HasBorrowed
tom:Member
name='Thomas Paine'
student_id=null
address=null
no_onloan=2
fine=null
you:Member
name='Livia Guimaraes'
student_id='C24329646'
address='Grangegorman Campus'
no_onloan=0
fine=null
WandP:Book
title='War & Peace'
author='Tolstoy'
publisher=null
no_copies=1
no_onshelf=1
book
copies
c3:Copy
status='onShelf'

7. In the constraints section, add 2 preconditions and 1 postcondition to specify more precisely the operation Member::borrow(c:Copy). Preconditions should state that the member has not already borrowed c and that c has not been borrowed by someone else. Postcondition should state that no_onloan has been updated.

8. Add an invariant to the Book class asserting that no single library member can borrow more than 1 copy of a book, i.e. if two copies of a book are out on loan, then they must have been borrowed by different members.

CODE “lib.use”:

```
model Library_2025
-- Nov 28, 2025
class Book
  attributes
    title : String
    author : String
    publisher : String
    no_copies : Integer
    no_onshelf : Integer
  operations
    borrow()

end

class Copy
  attributes
    status : String
  operations
    borrow( m : Member)

end

class Member
  attributes
    name : String
    student_id : String
```

```
    address : String
    no_onloan : Integer
    fine : Integer
  operations
    okToBorrow() : Boolean   -- returns true or false

    borrow(c : Copy)
    begin
        declare ok : Boolean;
        -- ok := self.okToBorrow();
        -- missing code
    end
end



-- associations
association CopyOf between
    Copy[1..*] role copies
    Book[1] role book
end
constraints
-- 20 marks in total (2025)
-- changes/modifications in the code:
association HasBorrowed between
    Member[0..1] role borrower
            Copy[*] role borrowed
end

constraints
context Member::borrow(c : Copy)
  pre beforeBorrow1 : c.status <> 'onloan' --<> means not equal to
  pre beforeBorrow2: self.borrowed -> excludes(c)

  post afterborrow1 : self.no_onloan > self.no_onloan@pre
  post afterborrow2 : self.borrowed -> includes(c)
```

9. Save your report as a PDF named report.pdf and upload it along with lib.use, lib.soil.