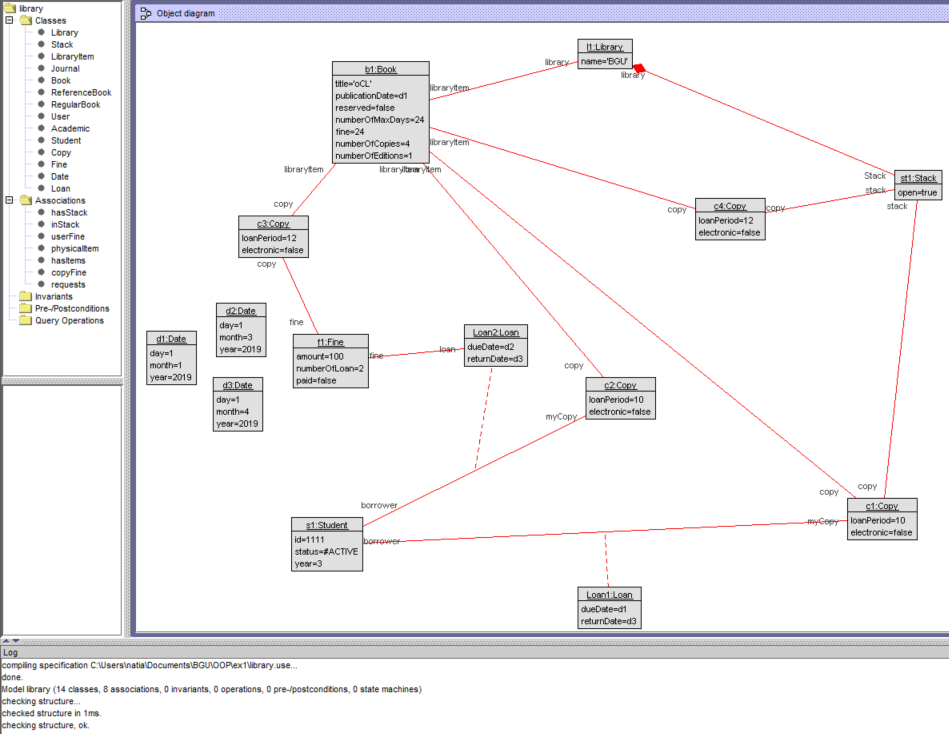
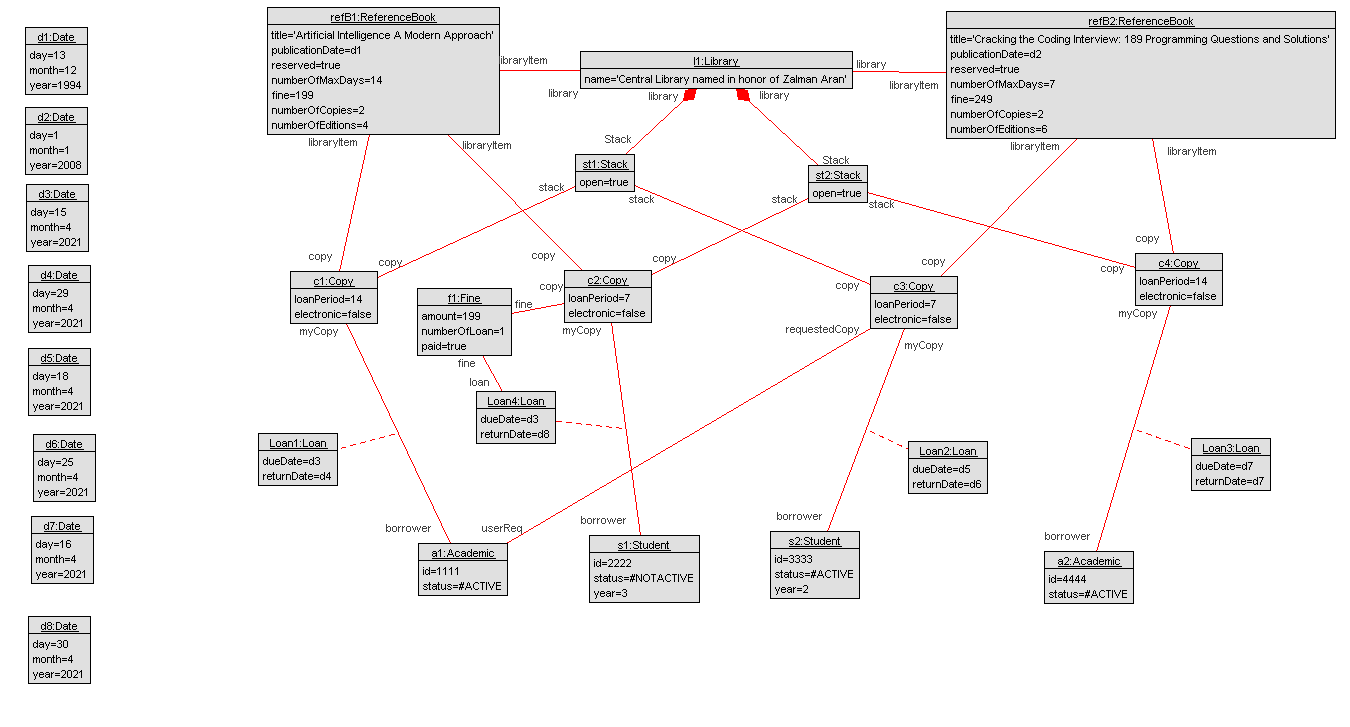
**שאלה 1**

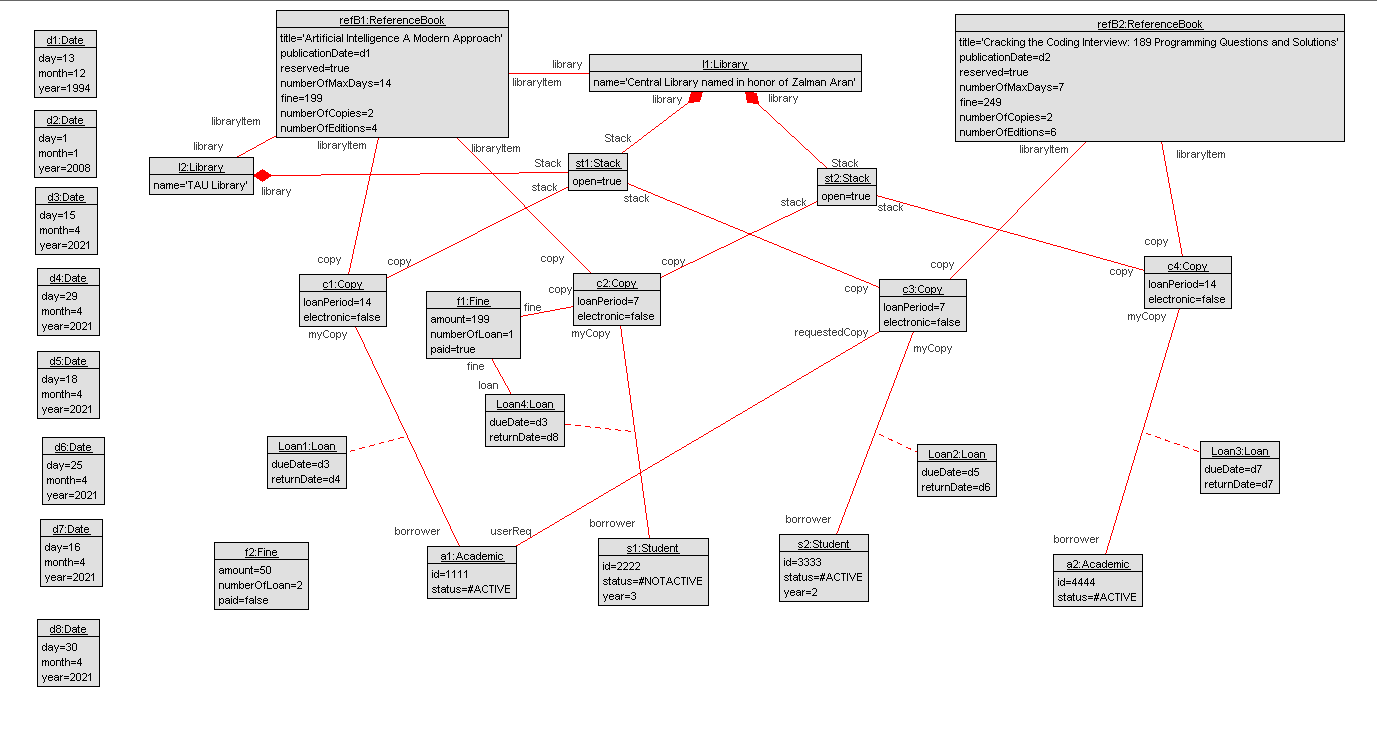
1. **מופע 1 אינו חוקי** – ניתן להבחין כי Loan2 הינו association class המחבר בין u1 גם ל c1 וגם ל c2, בניגוד למה שלמדנו בכיתה כי association classes מוסיפים אילוץ נוסף למודל לפיו קיים מופע יחיד של ה association בין כל 2 מופעים של מחלקות שהוא מחבר ביניהם.

**מופע 2 חוקי** – וידאנו זו בעזרת USE.



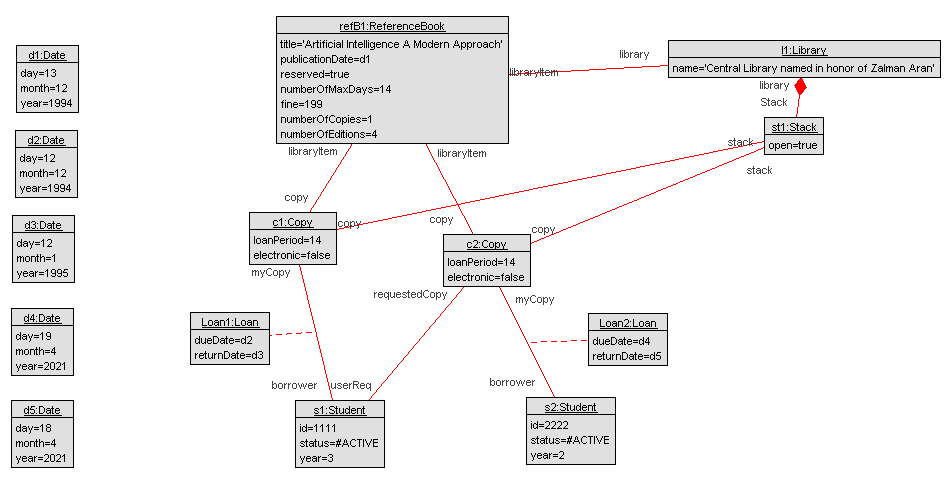
1. **legal**

**illegal**

****

האובייקט הנ"ל אינו חוקי מהסיבות הבאות:

* st1 שייך גם ל l1 וגם ל l2.
* f2 אינו מקושר לאובייקט של המחלקה Copy.
* f2 אינו מקושר לאובייקט של המחלקה Loan.
* refB2 אינו מקושר לאובייקט של המחלקה Library.
* refB1 מקושר גם ל l1 וגם ל l2.

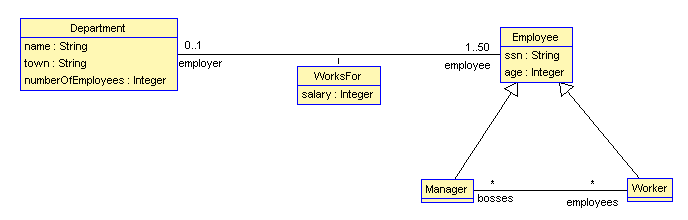
1. 

המופע הנ"ל חוקי אך מפר את הדרישות ההגיוניות הבאות:

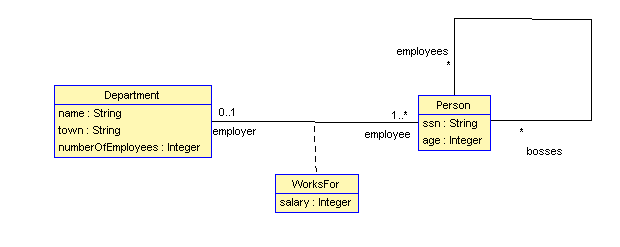
* השדה numberOfCopies של refB1 הינו 1, בעוד refB1 מקושר גם ל c1 וגם ל c2.
* השדה publicationDate של refB1 הינו d1, בעוד השדה dueDate של Loan1 אחד הינו d2, כאשר d2<d1 ע"פ הסדר הכרונולוגי.
* השדה dueDate של Loan1 הינו d2 והשדה returnDate של Loan1 הינו d3, בעוד השדה loanPariod של c1 הינו 14, כאשר d3-d2>14 ע"פ הסדר הכרונולוגי.
* s1 מקושר ב loan ל c1בעוד s1 מקושר ב requests ל c2.
* השדה dueDate של Loan2 הינו d4 והשדה returnDate של Loan2 הינו d5, כאשר d5<d4 ע"פ הסדר הכרונולוגי.

**שאלה 2**

**אילוץ 1 –Generalization Constraint**. נקבע מחלקה abstract בשם Employee שתכליל ע"פ מאפיינים של Complete ו Overlapping את המחלקות Manager ו Worker. אילוץ זה יכול למנוע את האפשרות כי יהיה קיים קשר bosses-employees בין אותו מופע של Person.



**אילוץ 2 –Multiplicity Constraint**. נקבע כי בכל Department יש בין 1 לאינסוף עובדים על מנת למנוע מצב שבמחלקה יש 50 עובדים, ואילו לעובד של מחלקה יש יותר מ 50 עובדים תחתיו.



**שאלה 3**

