

הצהרה: ידועה שמידת האמינות
נבחן שימצאו ברשותו הומוגר
עוד אסורים או יונקס בדומה
יענש בחומרה על כזי הרחקתו
מחברותו.

כפ"ד
אוניברסיטת בר-אילן
התחלקה לסדעי המחשב

תכנות מתקדם ו
מירי כן-ניסן (קופל)
התשע"ב / מועד א'



שאלון סגור

תכנות מתקדם 2

מועד ב' תשע"ב

31.8.12

המחלקה למדעי המחשב

89-211

שנת: תשע"ב סמסטר: 2 מועד: 2 מטלה: 1
קורס: 01 211 89 תכנות מתקדם 2
מס' מח': 9



22	30 נק'	1
25	25 נק'	2
15	15 נק'	3
15	15 נק'	4
15	15 נק'	5
92	100	סה"כ

מרצה:

יריב טל.

מתרגל:

איגור רוכלין.

משך הבחינה:

שלוש שעות. אין הארכה.

חומר עזר:

אסור להכניס כל חומר עזר.

הנחיות כלליות:

רצוי לענות בגוף הבחינה. אחרי כל שאלה יש מקום לתשובות.
אם עניתם על שאלה במחברת – ציינו זאת בגוף הבחינה!
חובה להגיש את המחברת ביחד עם טופס הבחינה בסיום הבחינה.
חובה לענות על השאלות בעברית, אלא אם יש אישור מהדיקן.

הנחיות טכניות:

במידה ונדרשתם לתת נימוק, אזי הוא חובה. כלומר, תשובה לא מנומקת לא תקבל נקודות כלל.
נימוק לא נכון יגרור פסילת השאלה.
בכל השאלות – מספר השורות שניתנו לפתרון אינו מרמז על אורך התשובה.

בהצלחה ☺

2000
12
12
12
12
12

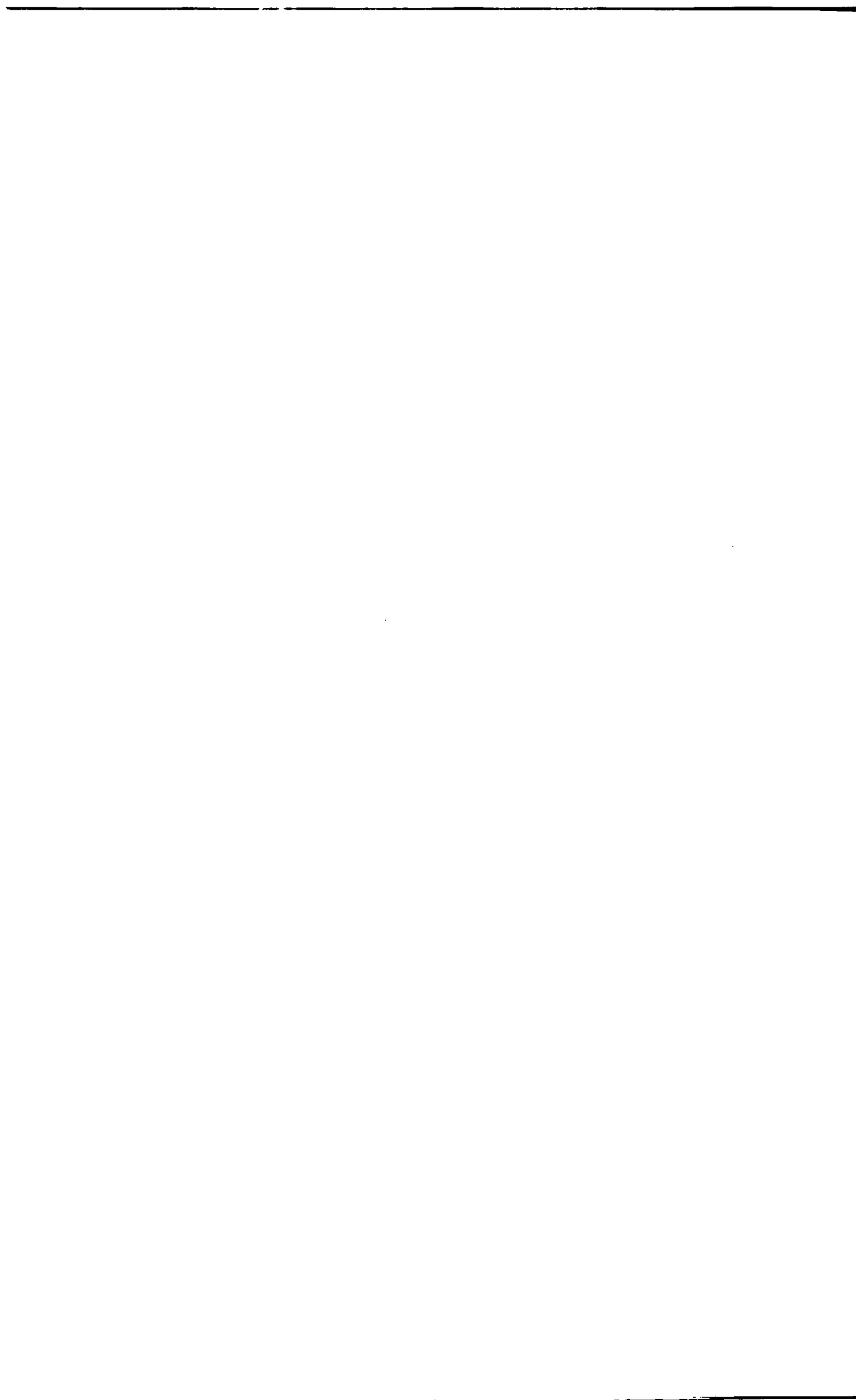


שאלה 1: Code Improvements & Testing (30 נקודות)

נתון הקוד הבא:

```
// OptionalValue is used when a function has an optional value.
// Example:
//     time_t getCurrentTime(OptionalValue timezone) {
//         // get cur time (in local timezone) somehow
//         time_t curtime = ...
//         if (timezone.HasValue()) {
//             curtime = convertToTimeZone(curtime, timezone.getValue());
//         }
//         return curtime;
//     }
//
//     void printTime() {
//         OptionalValue defaultTimeZone, grinchTimeZone(0);
//         std::cout << "Current time: " << getCurrentTime(defaultTimeZone);
//         std::cout << ", GMT Time: " << getCurrentTime(grinchTimeZone);
//     }
//
1) class OptionalValue {
2) public:
3)     OptionalValue() : hasValue(false) { }
4)     OptionalValue(int value) : hasValue(true), m_value(value) { }
5)
6)     // Return true if contains a value, false if not
7)     bool HasValue() { return hasValue; }
8)
9)     // Get the stored value. Assumes there *is* a value (use hasValue() first
10)    // to check if there is a value)
11)    int getValue() { return m_value; }
12)
13)    // Assign a value
14)    void setValue(int VALUE) { hasValue = true; m_value = VALUE; }
15)
16)    // Set back to no-value
17)    void clear() { hasValue = false; }
18)
19)    // operator* is used to access internal value as if OptionalValue is a
20)    // pointer to int. Note that it assumes that OptionalValue actually has a
21)    // value.
22)    // Example usage:
23)    //     OptionalValue noValue, withValue(5);
24)    //     *withValue = 7; // .ok
25)    //     *noValue = 3; // undefined behaviour - do not do this!
26)    int &operator*() { return m_value; }
27)
28) private:
29)     bool hasValue;
30)     int m_value;
31) };
```

קוד זה מממש מחלקה שמשמשת לייצוג ערך אופציונלי – אשר לא חובה להעביר לפונקציה. שימוש ב-default ctor מייצר אובייקט שמייצג "אין ערך", וכך מיידע את הפונקציה שמקבלת את ה-OptionalValue שלא צריך להשתמש בערך האופציונלי.



(3) $(int + VALUE) - 1$ $(int + value) - 1$
 כנסת 14 - $(int + VALUE) - 1$ $(int + value) - 1$
 תכנות מתקדם 1
 מירי כן-ניסן (קופל)
 היתשע"ב / מועד א

כס"ד
 אוניברסיטת בר-אילן
 המחלקה למדעי המחשב

שימוש ב-ctor שמקבל ערך (value) מייצר אובייקט שמייצג את הערך שבו רוצים שהפונקציה אליה מועבר ה-Optional Value תשתמש. לנחותכם, לצד הקוד ישנם מספרי שורה.

א. (6 נקודות) מיצאו לפחות שתי חריגות מה-Naming Conventions בהם יש שימוש בקוד.

- 1) new משהי `hasValue` עם `m` - `m = hasValue`
- 2) כמתודה `bool HasValue` עם `H` במקום `h` - `bool hasValue()`

ב. (6 נקודות) אילו פונקציות יכולות להיות מוגדרות כפונקציות `const`? נמקו ורישמו את החתימה של הפונקציות לאחר הוספת ה-`const`.

- 1) `bool HasValue() const;`
- 2) `int getvalue() const;`
- 3) `int operator*(int) const;`

חזרתי - תיקון

`const` reference

נמקו!

ג. (6 נקודות) בקוד מתועדות הנחות שצריכות להתקיים בעת הקריאה לחלק מהפונקציות. הוסיפו `assert` לקוד הבודקים שההנחות מתקיימות.

- 1) `int operator*(int)`
`{ assert(true == HasValue);`
`return m-value;`
`}`

- 2) `assert(true == HasValue)`

`true == HasValue`

1ד. (6 נקודות) הסבירו מדוע הפעלת אופרטור * על המשתנה readonly גוררת שגיאת קומפילציה.

האם יש יחסיות בין λ ל- λ_{rest} ?

~~XXXXXXXXXXXXXXXXXXXX~~

הפס"ו המוצע על ידי הקצב (בתורה שגבולות להחזיר
(חז"ל) וזהו שמו של הקצב
 $5 = \text{readonly}$ 1

2/6

~~Time complexity of $x()$ is $O(1)$, const.~~
~~The value of x is const. after 1st~~
 ~~m value of x is 1 of 2 of 3 of 4~~

[Signature]

202

127505 1/2

int operator*() const;

const is a keyword for declaring a constant variable.
(readonly) const is a keyword for declaring a constant variable.

הנהגת המטה החדשה כפי שהיא מוגדרת ב-const
היא, והיא נמצאת ב-const זהו שם המשתנה
החדש. המטה החדשה - const זהו שם המשתנה.

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100.

100

2000-2001
2001-2002

בס"ז
אוניברסיטת בר-אילן
המחלקה למדעי המחשב

תכנות מתקדם ו
מירי בן-גיסן (קופל)
ה'תשע"ב / מועד א'



```
Magic<2>::exec(p1);
Magic<3>::exec(p1);
```

ד. (5 נקודות) ממשו את p1 ב-Functor בשם Print. הדגימו כיצד תקראו ל-Magic<2>::exec עם ה-Functor שכתבתם.

2. 7/6/25

```
void operator()( )
```

3

print pg 1

```
magic(2)::exec(prt()),
```

ה. (5 בקורות) שנו את Print כך שאפשר יהיה להעביר לו פרמטר (מטיפוס std::string) של מה להדפיס. הדגימו קריאה ל-Magic<2>::exec() עם ה-Print החדש.

private:

std::string m_myString;

public;

```
print(std::string{-string}): myString{-string} } }
```

void operator()()

```
} std::cout << m.myString << std::endl,
```

3

```
print pg1("Walker");
```

$\text{magic} \langle 2 \rangle \text{exec}(\text{prn}_6())$

LINE

1. *Chlorophyll *a** and *Chlorophyll *b** were determined by the method of Arar and Collins (1971).

2.



שאלה 3: RAII + Templates (15 נקודות)

בתכנות multithreading ו-multithreading ישנם מקרים רבים בהם יש להגן על קטע מסוים כך שרק thread או process אחד יוכלו לגשת אליו בכל פעם. הצורה לבצע זאת היא בעזרת אובייקט נעילה. ישנם הרבה סוגים של אובייקטי נעילה, ברמות שונות של השפעה (ברמת ה-thread או ברמת ה-process), לדוגמא: Semaphore, ProcessMutex, ThreadsMutex. בספרייה שכתבתי לכל האובייקטים המשמשים לנעילה Concept אחד: יש להם פונקציה lock() לביצוע הנעילה ופונקציה unlock() לשיחרור הנעילה. יש לשים לב שכמו שבניהול זיכרון לכל new חייב להיות delete כך גם באובייקטי נעילה על כל lock() חייב להתבצע unlock(). הנה דוגמת קוד המשתמש ב-ThreadsMutex:

```
void foo(ThreadsMutex &mutex) {
    bool quickEscape = false;

    mutex.lock();

    // do something that mutex protects
    // and that might change quickEscape
    ...

    if (quickEscape) {
        return;
    }

    mutex.unlock();
}
```

אם `quickEscape` יהיה `true`?

א. (2 נקודות) איזה bug ייקרה אם `quickEscape` יהיה `true`?

אם `quickEscape` יהיה `true`, אז `unlock` לא יבוצע, ולכן `mutex` לא יתפסק. \Rightarrow `mutex` לא יתפסק.



ב. (8 נקודות) בכדי למנוע פוטנציאל לבאגים ולהקל על ניהול הנעילות הוחלט לכתוב Guard אשר יבצע את ה-
lock() וגם יבטיח את ביצוע ה-unlock().
כיתבו את ה-Guard והדגישו את השימוש בו בקוד של foo() (שימו לב: ה-Guard צריך להיות אחראי גם על
ביצוע את ה-lock-!).

class Guard {
public:
Guard(ThreadsMutex &mutex): m_mutex(mutex)
{ m_mutex.lock(); }
~Guard()
{ m_mutex.unlock(); }
private:
ThreadsMutex m_mutex;
};

Guard myGuard = Guard(mutex); | foo |
bool quickEscape = false; |
bool quickEscape = true; |

ג. (5 נקודות) מכיוון שלכל האובייקטים לנעילה בספרייה של Concept אחד, הוחלט לממש את ה-Guard כ-
template שמתאים לכל אובייקטי הנעילה.
כיתבו את ה-Guard מחדש כך שיהיה טמפלט.

template<typename T>
class Guard
{
public:
Guard(T &t): m_mutex(t)
{ m_mutex.lock(); }
~Guard()
{ m_mutex.unlock(); }
private:
T m_mutex;
};



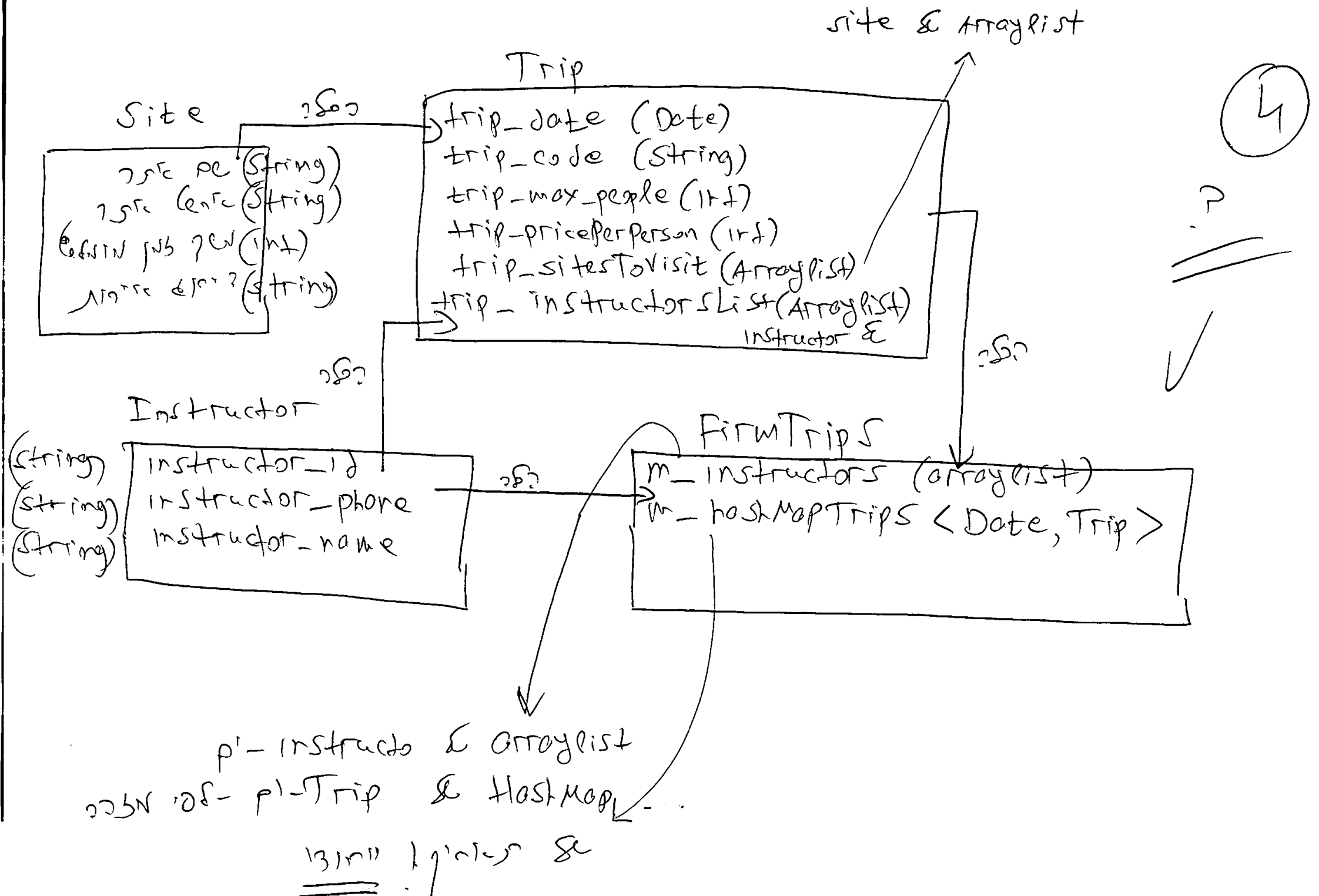
שאלה 4: Java (15 נקודות)

חברת טיולים מנהלת נתונים על מדריכים, סוגי טיולים, טיולים בפועל ומטיילים. לכל מדריך יש מספר מזהה, שם וטלפון. החברה מכינה תוכניות לתיאור של טיולים (להלן סוג טיול). לכל סוג טיול יש קוד מזהה, משך (מספר ימים), מכסת משתתפים, מחיר למשתתף ורשימת אתרים שבהם יבקרו. לכל אתר יש שם אתר והוא מצוי בארץ מסוימת. שם האתר ייחודי לארץ שבה הוא מצוי. עבור כל סוג טיול מתכננים את סדר הביקורים באתרים השונים (אתר ראשון, שני וכו'). לכל אתר יש גם משך זמן מומלץ לביקור (בשעות) ודרוג איכות (מס' כוכבים). החברה מבצעת טיולים מכל סוג במשך השנה. כל סוג טיול יכול להתבצע מספר פעמים בשנה אולם בתאריך מסוים יכול לצאת לכל היותר טיול מסוג טיול מסוים. בטיול יכולים להשתתף מספר מדריכים ויש לדעת מי הדרך בכל טיול. כן יש לדעת את מספר המשתתפים. א. (4 נקודות) מי הם הישויות (actors) של המערכת?

1. מדריך Instructor
2. חברה Firm
3. אתר (site)
4. טיול Trip

ב. (11 נקודות) צרו תרשים המחלקות של המערכת (class diagram). עבור כל מחלקה, רשום את תכונותיה והקשרים המבניים למחלקות אחרות.

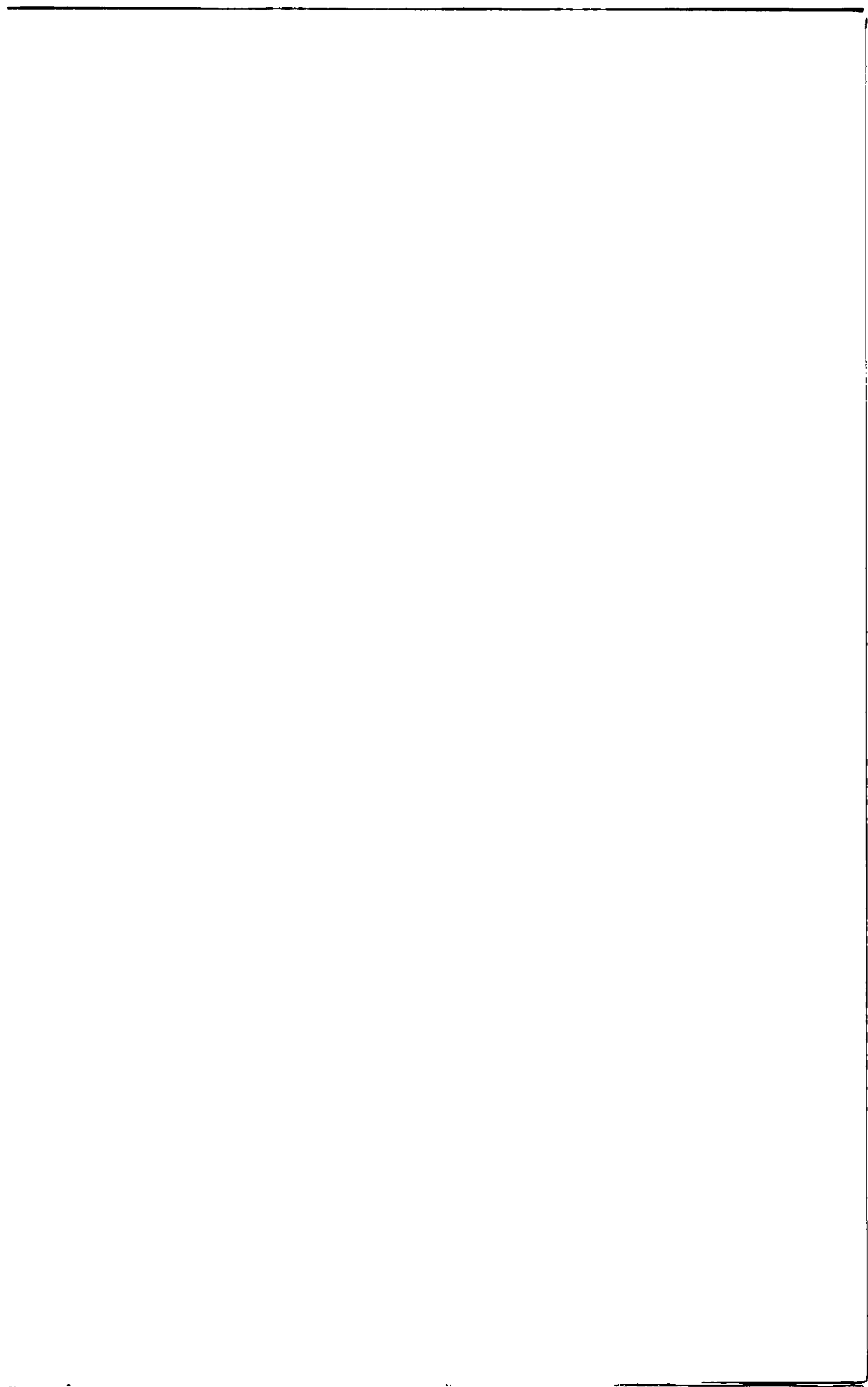
נסים



DOM ו-XML מונחים שונים לחלוטין. DOM הוא מודל של מסמך XML, בעוד שXML הוא סט של כללים להגדרת מסמכים. DOM מאפשרת לך לראות את המסמך כעץ, בעוד שXML מאפשרת לך לכתוב מסמכים בצורה קצרה וברורה.

[illegible]

```
public interface XMLParser {
    public Document loadXML(String nameOfFile);
    public List<Node> getChildNodes(Document document);
    public List<Node> getChildNodes(Node node);
    public Double getChildData(Node node);
}
```





}

עליכם לתכנת את המחלקה הבאה ולהשלים את הפונקציה loadTree:

public class Tree{

.....

public static Tree loadTree(XMLParser parser, String nameOfFile){

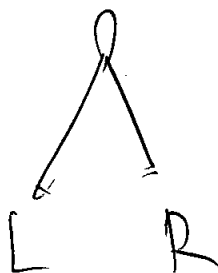
.....

}

}

המחלקה מייצגת עץ (לא רק עץ בינארי). הפונקציה loadTree צריכה לקבל XML מהצורה הבאה:

```
<Tree>
  <data>1.25</data>
  <Tree>
    <data>5</data>
    <Tree>
      <data>3.25</data>
      <Tree>
        <data>4.25</data>
      </Tree>
    </Tree>
  <Tree>
    <data>1.75</data>
  </Tree>
</Tree>
<Tree>
  <data>4</data>
</Tree>
<Tree>
  <data>2</data>
  <Tree>
    <data>1.05</data>
  </Tree>
  <Tree>
    <data>22</data>
  </Tree>
  <Tree>
    <data>33</data>
  </Tree>
  <Tree>
    <data>111</data>
  </Tree>
</Tree>
```







ולהחזיר עץ מלא בנתונים מ XML. יש להוסיף משתנים, פונקציות עזר למחלקה Tree במידת הצורך. אם יש צורך אתם יכולים לתכנת עוד מחלקה/ות. יש להיעזר בממשק XMLParser אשר פועל על עיקרון DOM (אין צורך לדאוג למימושו). יש להניח שהקלט תקין.

```

void
private void addTreeNode(List<Node> myNodes, XMLParser parser)
{
    if (myNodes.size() == 1) // ישו צורה
    {
        myTree.addNode(parser.getChildData(
            myNodes.at(0)));
        return;
    }

    List<Node> current = null; // תלוי
    myNodes.add(current); // קורה סימן
    myNodes.add(new Node(myNodes));
    int size = myNodes.size(); // get size of list
    int i = 0;
    while (i < size)
    {
        current = parser.getChildNodes(myNodes.at(i));
        int size2 = current.size();
        myNodes.add(current);
        if (myNodes.add (size2)) // אם יש צורה
        {
            addTreeNode(current, parser);
        }
        i++;
    }
} // end of addTreeNode
    
```

~~Parser~~ ~~Abstract class~~ ~~XML Parser~~ ~~main~~

```
public static Tree loadTree(XMLParser parser, String
                             nameOfFile)
```

```
{ Document doc = parser.loadXML(nameOfFile);
  Tree myTree = new Tree();
  List<Node> myNodes = parser.getChildNodes(doc);
  addToTree(myNodes, parser); // call recursive
  return myTree;
}
```

public : ~~Tree~~ ~~Tree~~ ~~Tree~~

```
class Tree {
```

① public static Tree loadTree(XMLParser parser, String nameOfFile);

② private void addToTree(List<Node> myNodes, XMLParser parser);

③ private void addNewData(Double value) *
 myTree & value & n

private one of Java's Tree, myTree;

הנה פירוט של מה שכתבתי
 בדף הקוד - הנה פירוט של מה שכתבתי
 בדף הקוד (עמוד 3) - הנה פירוט של מה שכתבתי
 בדף הקוד