

גיליון תשובות

מספר נבחן: _____

(30 נקודות)

שאלה 1

סעיף א (10 נקודות)

```
public boolean checkIndex(int index){
    return index >= 0 && index <= _arr.length
}

public boolean checkValue(int value){
    return (value == INITIALIZED) ||
           (value == UNSET) ||
           (value >= 0 && value <= _arr.length);
}

Invariant: (_arr != null) and for each cell in _arr, checkValue returns true.

public boolean checkInv(){
    if (_arr == null)
        return false;
    for (int a : _arr){
        if (!checkValue(a))
            return false;
    }
    return true;
}
```

סעיף ב (10 נקודות)

שימו לב שבסעיפים 1,2 אין מקביליות.

a) 0's leader is 0
1's leader is 0

b) 1's leader is 1
0's leader is 1

c) 1's leader is 0	1's leader is 1	1's leader is 1
0's leader is 0	0's leader is 1	0's leader is 0

השורה של 0 יכולה להיות מודפסת לפני השורה של 1.

סעיף ג (4 נקודות)

```
public void run(){
    synchronized(_memory){
        .
    }
}
```

```

        .
        .
        .
    } //sync
} //run

```

סעיף ד (6 נקודות)

המנהיג יהיה ה Processor הראשון שיבצע את המתודה CBS כדי לברר האם הוא הראשון שולחים את הערך 0 כפרמטר cmp למתודה, כדי לוודא שלא נבחר עדיין מנהיג (השדה _val עדיין שלילי). במידה ולא נבחר עדיין מנהיג יהיה זה ה Processor (טכנית נשלח $_id+1$ כפרמטר swp ולא $_id$ (תוך הפחתת 1 בסיכום) עבור מקרה קצה של $_id=0$)

```

class Processor implements Runnable{
    private CompareNSwap _cns;
    private int _id;

    public Processor(int id, CompareNSwap cns){
        _cns = cns;
        _id = id;
    }

    public void run(){
        int leader = _cns.CNS(0, _id+1) - 1;
        System.out.println(_id+"'s leader is "+leader);
    }
}

```

(16 נקודות)

שאלה 2

```

42 30
10 30
42 30
30 30

```

(14 נקודות)

שאלה 3

סעיף א (4 נקודות)

$A \ B \ \sim B \ \sim A$

סעיף ב (4 נקודות)

בקטע הקוד הנתון מבוצעת השמה של אובייקטים:

$b1 = b2;$

נשים לב כי לא מוגדר במחלקות A ו B אופרטור השמה, כך שתבוצע כברירת מחדל השמה bit by bit. מכך נובעות שתי בעיות בניהול זיכרון:

Memory Leak: הזיכרון שכתובותיו ניתנות על ידי השדות `_p1`, `_p2` של `b1` אינו משתחרר.
Null Pointer: בגמר הבלוק בו הוגדר `b2` מתבצע ה `destructor` שלו, המנקה את הזכרון המוצבע על ידי שדותיו `_p1`, `_p2`, אך עקב פעולת ההשמה הנ"ל שדותיו של `b1` מצביעים לאותם מקומות מחוקים. בביצוע השורה האחרונה מתבצעת אף פניה לזיכרון מחוק זה.

הערה: אין בעיה של non virtual destructor כי `b1` ו `b2` הינם אובייקטים בעלי טיפוס סטטי הידוע בזמן קומפילציה (הייתה בעיה רק אם הם היו טיפוסים דינאמיים, שאופיים המדויק נקבע רק בזמן ריצה, כמו `A*`). בכל מקרה לא ירדו נקודות למי שטען זאת.

סעיף ב (6 נקודות)

יש להגדיר אופרטור השמה עבור המחלקות A ו B:

```
class A {
...
    A& operator=(const A& other) {
        if (this != &other)
            (*_p1) = (*other._p1);
        return (*this);
    }
...
};

class B : public A {
...
    B& operator=(const B& other) {
        if (this != &other) {
            ((A)*this) = ((A)other);
            (*_p2) = (*other._p2);
        }
        return (*this);
    }
...
};
```

(30 נקודות)

שאלה 4

סעיף א (4 נקודות)

```
socket.joinGroup(groupAddress);
...
socket.leaveGroup(groupAddress);
```

סעיף ב (4 נקודות)

הלקוח הוא יוזם ההתקשרות ועליו לשלוח לשרת הודעה עם שם הסרט המבוקש:

```
movie: <name of the movie>
```

סעיף ג (6 נקודות)

```
out = new PrintWriter(outsocket.getOutputStream(), true);
out.println("movie: " + "The Cabinet of Doctor Caligari");
packet = new DatagramPacket(new byte[1024], 1024);
while (keepWatching()) {
    insocket.receive(packet);
    player.play(packet.getData());
}
```

סעיף ד (4 נקודות)

במקרה שלנו רוב העומס נופל על עבודת התקשורת (שליחת סרטים) – ועל כן נעדיף הרבה thread-ים שיטפלו בשליחת סרטים. לעומת זאת, עיבוד ההודעה שבה מבוקש שם הסרט אינו צורך זמן ואין צורך להקצות לו מספר רב של thread-ים.

סעיף ה (12 נקודות)

צריך להוסיף ל ConnectionHandler את השדות

```
DatagramChannel _dgChannel;
String _movie;
InetSocketAddress _clientAddress;
```

@1:

```
_movie = processMessage(message);
_dgChannel = DatagramChannel.open();
_dgChannel.configureBlocking(false);
_clientAddress = new InetSocketAddress(address.getAddress(), 2007);
dgChannel.connect(_clientAddress);
dgChannel.register(_data.getSelector(), OP_WRITE, this);
```

@2+3:

השדות של DataSender יהיו:

```
DatagramChannel _dgChannel;
String _movie;
InetSocketAddress _clientAddress;
```

הבנאי:

```
DataSenderTask(DatagramChannel dgChannel, String movie, InetSocketAddress
clientAddress) {
    _dgChannel = dgChannel;
    _movie = movie;
    _clientAddress = clientAddress;
}
```

יצירת DataSenderTask:

```
_data.getExecutor().execute(
    new DataSenderTask(_dgChannel, _movie, _clientAddress));
```

@4:

```
try {
    File file = new File(_movie);
    FileInputStream fileIn = new FileInputStream(file);
    byte[] buf = new byte[1024];
    while (in.read(buf)!=-1)
        out.send(buf, _clientAddress);
} catch (Exception e){
}
```

(10 נקודות)

שאלה 5

סעיף א (5 נקודות)

```
SELECT DISTINCT Teams.Contact, Games.Stadium
FROM Teams OUTER LEFT JOIN Games
ON (Teams.Club = Games.Club1 AND Teams.City = Games.City1)
OR
(Teams.Club = Games.Club2 AND Teams.City = Games.City2)
```

סעיף ב (5 נקודות)

