

גיליון תשובות

מספר נבחן: _____

על תשובות ריקות יינתן 20% מהניקוד!

(30 נקודות)

שאלה 1

סעיף א (7 נקודות)

שני ת'רדים המפעילים את מתודת ה `move` על אחד משני גלגלים סמוכים מסוג `SimpleGear` הת'רד הראשון תופס את הגלגל הראשון (`this` שלו), המתזמן עובר לת'רד השני תופס את הגלגל השני (`this` שלו). לאחר מכן כל אחד מהת'רדים מנסה לתפוס את הגלגל הסמוך במסגרת ההפעלה של `moveLeft` או `moveRight`.

סעיף ב (18 נקודות)

מיקוד הסנכרון: בזמן ביצוע `moveLeft` או `moveRight` על הגלגלים הסמוכים אין צורך לנעול את הגלגל הנוכחי, כך שניתן לצמצם את הסנכרון לעדכון המצב בלבד

```
public synchronized void move(boolean clockwise) {
    synchronized (this) {
        float offset = getOffset(_radius);
        if (clockwise)
            _state = (_state + offset) % 360;
        else
            _state = (_state - offset) % 360;
    }
    if (_left != null)
        getLeft().moveLeft(!clockwise);
    if (_right != null)
        getRight().moveRight(!clockwise);
}
```

שימוש בסמפור: ננצל את אופציית ה `tryAcquire` כדי לנסות לתפוס את הגלגל השכן. במידה ואי אפשר, נשחרר את הגלגל הנוכחי וננסה שוב.
הערה: הכוונה המקורית בשאלה הייתה למצוא פתרון שאינו מסתמך על מיקוד הסנכרון בסעיף הקודם. הוחלט לקבל תשובות שהסתמכו למעשה על מיקוד הסנכרון, רק עם סמפור במקום סנכרון.

```
class SimpleGear implements Gear
{
    ...
    public Semaphore _sem = new Semaphore(1);

    public synchronized void move(boolean clockwise) {
```

```

boolean l = false, r = false;
while (!b) {
    _sem.acquire();
    l = getLeft()._sem.tryAcquire();
    r = getRight()._sem.tryAcquire();
    if (l & r)
        b = true;
    else {
        _sem.release();
        if (l)
            getLeft()._sem.release();
        if (r)
            getRight()._sem.release();
        Thread.sleep(1000);
    }
}

float offset = getOffset(_radius);
if (clockwise)
    _state = (_state + offset) % 360;
else
    _state = (_state - offset) % 360;

if (_left != null)
    getLeft().moveLeft(!clockwise);

if (_right != null)
    getRight().moveRight(!clockwise);
_sem.release();
getLeft()._sem.release();
getRight()._sem.release();
}
...
}

```

בנוסף, יש לעדכן את המתודות במחלקה כך שבמקום הסנכרון על `this` יעשה שימוש בסמפור `_sem`.

Resource ordering: נתפוס את הגלגל הנוכחי והגלגלים השכנים ע"פ סדר קבוע. שימו לב כי כדי למנוע כל חבק אפשרי יש לתפוס מראש את כל הגלגלים בשרשרת. הוחלט לתת ניקוד מלא גם למי שהתייחס רק לגלגלים השכנים הראשונים.

class SimpleGear implements Gear

```

{

public Semaphore _sem = new Semaphore(1);

```

```

...
public synchronized void move(boolean clockwise) {

    List<Gear> orderedGears = getOrderedGears();
    for (Gear gear : orderedGears)
        gear._sem.acquire();
    float offset = getOffset(_radius);
    if (clockwise)
        _state = (_state + offset) % 360;
    else
        _state = (_state - offset) % 360;
    if (_left != null)
        getLeft().moveLeft(!clockwise);
    if (_right != null)
        getRight().moveRight(!clockwise);
    for (Gear gear : orderedGears)
        gear._sem.release();
}

List<Gear> getOrderedGears() {
    TreeMap<Long, Gear> orderGearMap = new TreeMap<Long, Gear>();
    Gear g = getLeft();
    while (g != null && g != this) {
        orderGearMap.put(System.identityHashCode(g), g);
        g = g.getLeft();
    }
    g = getRight();
    while (g != null && g != this) {
        orderGearMap.put(System.identityHashCode(g), g);
        g = g.getRight();
    }
    return orderGearMap.values();
}
...
}

```

בנוסף, יש לעדכן את המתודות במחלקה כך שבמקום הסנכרון על `this` יעשה שימוש בסמפור `_sem`.

סעיף ג (5 נקודות)

בשתי השיטות האחרונות יש תפיסה מיותרת של משאבים שאינה נדרשת בשביל הנכונות, כך שמיקוד הסנכרון הוא האפשרות היעילה יותר.

סעיף א (15 נקודות)

```

mvector::mvector(const mvector &m){
    _len = m._len;
    _array = new my_node[_len];

    for (unsigned i=0; i<_len; i++)
        _array[i] = m._array[i];
}

void mvector::insertAt(const my_node &n, unsigned pos){
    if (pos>=_len){
        int _pow = (int)floor(log((float)pos)/log(2.0))+1;
        int len = (int)pow ((float)2,_pow);

        my_node *tmp = new my_node[len];

        for (unsigned i=0; i<_len; i++)
            tmp[i] = _array[i];

        delete _array;
        _array = tmp;
        _len = len;
    }

    _array[pos] = n;
    _array[pos].setisFree(false);
}

void mvector::removeAt(unsigned pos){
    _array[pos].setisFree(true);

    bool isDelete = true;

    for (int i=_len/2; i<_len; i++){
        if (!_array[pos].isFree()){

```

```
                isDelete = false;
                break;
            }
        }

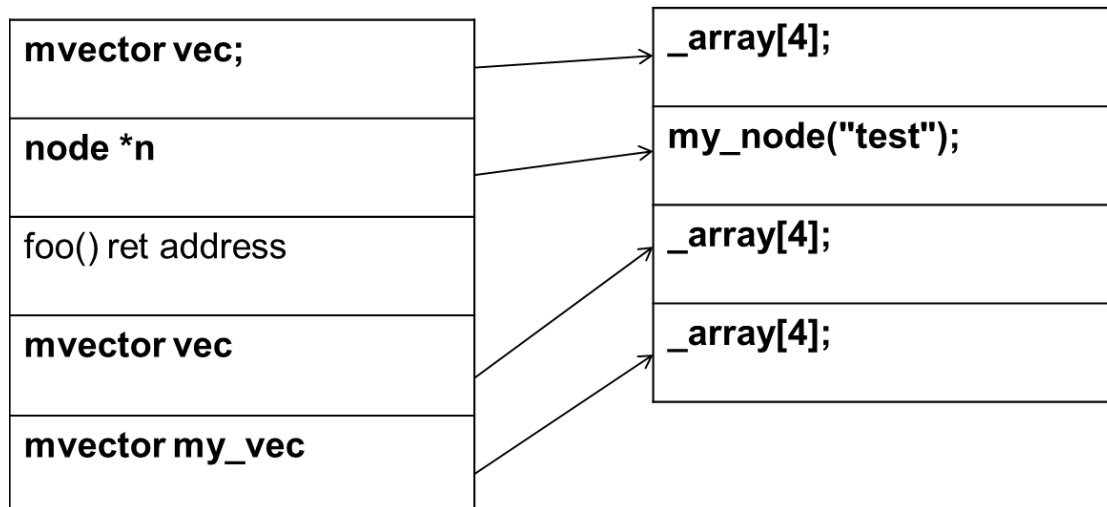
        if (!isDelete)
            return;

        int len = _len/2;
        my_node *tmp = new my_node[len];

        for (unsigned i=0; i<_len; i++)
            tmp[i] = _array[i];

        delete _array;
        _array = tmp;
        _len = len;
    }
```

סעיף ב (10 נקודות)



סעיף ג (5 נקודות)

4

```
class MessagingServer {
    public static void main(String[] args)
        throws NumberFormatException, IOException
    {
        if (args.length != 1) {
            System.err.println("please supply only one argument, the port to bind.");
            return;
        }

        Encoder encoder = new SimpleEncoder("UTF-8");
        ServerSocket socket = new ServerSocket(Integer.parseInt(args[0]));
        while (true) {
            Socket s = socket.accept();
            Tokenizer tokenizer = new MessageTokenizer(
                new InputStreamReader(s.getInputStream(), encoder.getCharset()), '\n');
            MessagingProtocol protocol = new LinePrintingProtocol();
            Runnable connectionHandler = new ConnectionHandler(s, encoder, tokenizer, protocol);
            new Thread(connectionHandler).start();
            ExecutorService executor = Executors.newFixedThreadPool(10);
            executor.execute(connectionHandler);
        }
    }
}
```

```
public class MessageLengthBasedTokenizer implements Tokenizer {

    private final InputStreamReader _isr;
    private boolean _closed;

    public MessageTokenizer (InputStreamReader isr) {
        _isr = isr;
        _closed = false;
    }

    public String nextToken() throws IOException {
        if (!isAlive())
```

```

        throw new IOException("tokenizer is closed");

String ans = null;
try {
    int c;
    StringBuilder sb = new StringBuilder();
    int size = -1;
    while ((c = _isr.read()) != -1) {
        if (c == " ")
            size = Integer.parseInt(sb.toString());
    }
    sb = new StringBuilder();
    for (int i=0;i<size(); i++)
        sb.append((char) _isr.read());
    ans = sb.toString();
} catch (IOException e) {
    _closed = true;
    throw new IOException("Connection is dead");
}
return ans;
}

public boolean isAlive() {
    return !_closed;
}

```

סעיף ג (5 בקודות)

```

class SpellMessageEncoder implements Encoder<SpellMessage> {
    public byte[] toBytes(SpellMessage msg) {
        return new String(msg.getRequest().name() + " " +
            msg.getWord() + " " +
            msg.getLanguage().name()
        ).getBytes("UTF-8");
    }

    SpellMessage fromBytes(byte[] buf) {
        String msg = new String(buf,0,buf.length,"UTF-8");
        String[] toks = msg.split(" ");
        return new SpellMessage(Request.forValue(toks[0]),toks[1],Language.forValue(toks[2]));
    }
}

```



```

public class SpellCheckingProtocol implements MessagingProtocol {

    private boolean _shouldClose;
    private SpellChecker _spellChecker;
    private SpellMessageEncoder _encoder;

    public LinePrintingProtocol() {
        _shouldClose = false;
        _spellChecker = new LexiconBasedSpellChecker ;
        _encoder = new SpellMessageEncoder();
    }

    public boolean shouldClose() {
        return _shouldClose;
    }

    public void connectionTerminated() {
        _shouldClose = true;
    }

    public String processMessage(String msg) {
        String ans = null;

        if (msg != null) {
            if (isEnd(msg))
                _shouldClose = true;
            else {
                SpellMessage sMsg = _encoder.fromBytes(msg.getBytes("UTF-8"));
                switch sMsg.getRequest() {
                    case CHECK:
                        return Boolean.toString(_spellChecker.check(sMsg.getWord(),sMsg.getLanguage()));
                        break;
                    case GET_CORRECTIONS:
                        return _spellChecker.getCorrections(sMsg.getWord(),sMsg.getLanguage()).toString();
                        break;
                }
            }
        }
        return ans;
    }

    public boolean isEnd(String msg) {

```

```

    return msg.equalsIgnoreCase("bye");
}
}

```

```

class SpellingServer {
    public static void main(String[] args)
        throws NumberFormatException, IOException
    {
        if (args.length != 1) {
            System.err.println("please supply only one argument, the port to bind.");
            return;
        }

        Encoder encoder = new SimpleEncoder("UTF-8");
        ServerSocket socket = new ServerSocket(Integer.parseInt(args[0]));
        while (true) {
            Socket s = socket.accept();
            Tokenizer tokenizer = new MessageLengthBasedTokenizer (
                new InputStreamReader(s.getInputStream(), encoder.getCharset()));
            MessagingProtocol protocol = new SpellCheckingProtocol();
            Runnable connectionHandler = new ConnectionHandler(s, encoder, tokenizer, protocol);
            new Thread(connectionHandler).start();
        }
    }
}

```

סעיף ה (5 נקודות)

```

interface SpellChecker extends Remote {
    boolean check(String word, Language lang) throws RemoteException;
    List<String> getCorrections(String word, Language lang) throws RemoteException;
}

class LexiconBasedSpellChecker extends UnicastRemoteObject {
    ...
}

```

סעיף ו (2 נקודות)

```

class SpellCheckingClient {
    public static void main(String[] args) {

```

```
SpellChecking spellChecker = (SpellChecking )Naming.lookup(args[0]);  
if (!spellChecker.check("recieve"))  
    System.out.println(spellChecker.getCorrections("receive"));  
}  
}
```

סעיף א (5 נקודות)

אפשר לתת מבנה טבלאי או קוד SQL – אין צורך בשניהם.
להלן הושמטה הטבלה Activities ממועד א', שאינה נדרשת בפתרון הנתון כאן.

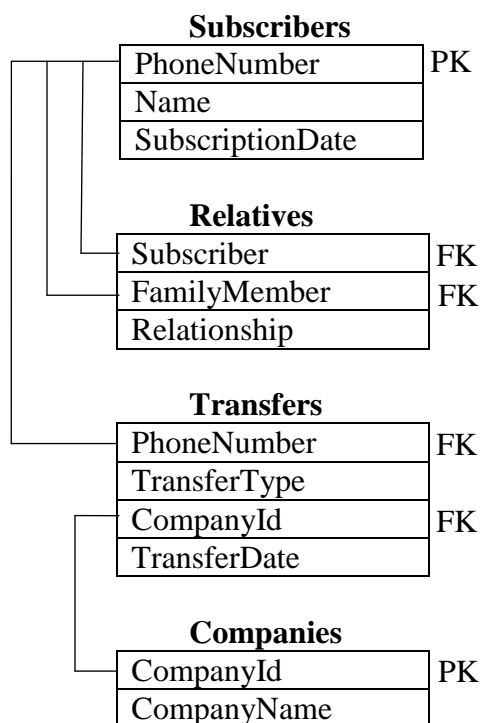
```
create table Subscribers
  (PhoneNumber integer primary key,
   Name varchar(30),
   SubscriptionDate date) --null if currently not subscribed
```

```
create table Companies
  (CompanyId integer primary key,
   CompanyName varchar(30))
```

```
Create table Relatives
  (Subscriber integer references Subscribers(PhoneNumber),
   FamilyMember integer references Subscribers(PhoneNumber),
   Relationship varchar(20))
```

```
Create table Transfers
  (PhoneNumber integer references Subscribers(PhoneNumber),
   TransferType bit, --0/1 indicates transfer from/to BigApple
   CompanyId integer references Companies(CompanyId),
   TransferDate date)
```

הערה: התקבל כל פתרון הגיוני, עקבי ונכון.
למשל, מודל המשתמש בשם החברה כמפתח, במקום במזהה נומרי.
למשל, ניהול הנייד בעזיבות בלבד (ללא TransferType).



סעיף ב (5 נקודות)

השאילתה נתונה להלן בשלש גרסאות שונות.

```
select Companies.CompanyName,Transfers.TransferDate
  from Subscribers join Transfers
    on Subscribers.PhoneNumber=Transfers.PhoneNumber
   join Companies on Transfers.CompanyId=Companies.CompanyId
 where Subscribers.Name="Ringo Starr" and TransferType=0
```

```
select CompanyName,TransferDate
  from Subscribers,Companies,Transfers
 where Subscribers.PhoneNumber=Transfers.PhoneNumber
    and Transfers.CompanyId=Companies.CompanyId
    and Name="Ringo Starr"
    and TransferType=0
```

```
select CompanyName,TransferDate
  from Subscribers join Transfers
    on Subscribers.PhoneNumber=Transfers.PhoneNumber
    and Name="Ringo Starr"
    and TransferType=0
   join Companies on Transfers.CompanyId=Companies.CompanyId
```

סרגל הבדיקה:

סעיף א' – מודל הנתונים

חלוקת ניקוד:

הוספת סיג הקשר המשפחתי: 1 נק'

טבלת חברות הסלולר: 2 נק'

טבלת הניוד: 2 נק'

הורדות:

אי ציון מפתחות ראשיים או זרים: 2- נק'

כפילויות נתונים: 1- נק'

מיקום לא נכון או לא הגיוני של שדות: 1- נק'

שינויים במקומות שלא נדרשו: 1- נק'

מודל נתונים בעברית: 1- נק'

סעיף ב' – שאילתה

הורדות:

שאילתה ללא join: 2- נק'

שימוש ב- join לא מתאים או לא נכון או מיותר: 1- נק'

שדות חסרים ב- select או שנלקחו ממקום לא מתאים: 1- נק' לכל פריט (חברה ותאריך)

חוסר מיקוד (where) עבור רינגו או מיקוד לא נכון: 1- נק'

שגיאת תחביר גסה בשאילתה: 1- נק'

שאילתה בעברית: 1- נק'