

100

זמן המבחן: שעותיים, יש לענות על 6 מתוך 6 שאלות, בגוף השאלון בלבד. חומר סגור.

(83)

שאלה 1: (20 נק') לאפליקציית ה web שלנו יש בעיית סקלביליות.

- אהרון טוען שיש לבצע scale out, ואז load balancing באמצעות אלג' round robin
- כלומר הטלת המשימות (טיפול בבקשות הלקוחות) על השרתים תתבצע ע"פ סבב קבוע.
- ברכה טוענת שיש לבצע scale up, ומכיוון שמדובר ב sticky session עם הלקוח, אז ממילא חלוקת המשימות צריכה להתבצע באמצעות session affinity. נילס מוסיף
- כלומר אותו השרת שטיפל בלקוח מסוים ימשיך לטפל גם בבקשות הבאות שלו, ולכן אין צורך ב scale out.
- גדעון לא מסכים עם ברכה. הוא לא מאמין ב session affinity והוא טוען שיש צורך ב scale out.
- כל שרת יוכל לטפל בכל משימה שתהיה, כי את ה data של ה session נשמור בשרת אחר המהווה central session store, וכך נוכל לבצע load balancing טוב יותר!
- דקלה מציעה גם scale out וגם session affinity באופן הבא:
- כבר בבקשה הראשונה נשמור ב cookie אצל הלקוח את ה ID של ה session. בכל בקשה נוספת מהלקוח הוא יעביר את המידע שב cookie וכך נוכל להקצות את המשימה אל השרת שטיפל בו בעבר.
- הרשלה טוען שיש לבצע scale out, IP Address Affinity, בהינתן בקשת לקוח, נריץ hash על ה IP שלו, והתוצאה מודולו N תקבע מי מתוך N השרתים צריך לטפל בבקשה שלו.

עבור על אחד תארו בקצרה מהם היתרונות והחסרונות בשיטה שהציע (5 נק' לכל שיטה).

אהרון יתרונו: כל הפרטים יעקבו קמקסל וכן מילא, יולד מצב של המטח

י"ב) בל שנה אחת אלא כן יתחיל בשלום שנה כן ב השלום

חסרונות: אין יסטי משמח של אמת וקיום והן נקלצו כל שמש שמש ויהיו אל

[illegible]

ברכה יתרונית: מניין שאלו שנים יסל נסאו הקקלה, נרעש, ש, סמנו סקרו הנקסו הלא סוני

[illegible]

חסרונות: סכנת ויתור, קושי להשיג מידע, חוסר יכולת להעריך את המידע, חוסר יכולת להעריך את המידע.

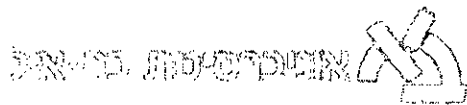
טען העקלע פאר אונז שוין ערשט אלס קינד ווייל פארה' יחיד

גדען יתרונות: מעין, עקצ, sale out יתרון גרוס סימון אה (המכיר שו המל)

וְכֵן עַל לֵב אֶל כָּל הַיְּהוּדִים וְהַכֹּהֲנִים שֶׁלֹּא יִהְיֶה מִתְּחִילָה שׂוֹמֵר מִצְוָה אֶחָד

סכרונות: יגל, אורנג'י, ציאור, (קדמק) (י) central session ~~for network~~

מדובר בפרק זה



תכנות מתקדם 2, מרצה: ד"ר אליהו חלסצ'י, מתרגל: מר חמי ליבוביץ

דקלה יתרונות: לא נאנקז את פסטי הסק, מניין שהקח ייבא להתבטא את זה, נצטק להחליף
שם, נספס שיטתני את פסטי הסק (בלינק וטל) להיות קטן בקוץ וטל

חסרונות: אולי שר מכלל המוט לקוח ויוצר, מלבד ששר אחד סוכר כל המון וישם שמי
נשים שילד לקוח את העקר. מוטו שר נחם לנהל כלן

הרשלה יתרונות: נצינו את העקר. של המוט למנון שיה וליק כל המוט יבדדו ולי ויוצו
מלט ששר אחד ליל מכלד דבר כמון ששר שר מכלד וקעטר הרותק את המוט של

חסרונות: קמדה מוטו לקוח ואחר המוט קום נצטק למטה את כל המוט מחדש

במטח ה מוטו. ומו דבר יקר שמוט כ- זמט Consistent מוטו מוטו יק כל
המט: ש מוטו שר מוטו לקוח ולכן נצטק שר שמוט את כל פסטי הסק

שאלה 2: (12 נק) וצק יק שר אחד ימטו מכלד ששר הרותק הן ומוט דם כלס קשר הקום
הקיפו בעיגול את התשובות הנכונות: ש אלה זקית.

א. ניתן באמצעות double check locking להתגבר על בעיית ה singleton בסביבה שהיא
multithreaded.

MVVM ניתן ממש רק ב .NET. בטכנולוגיית WPF. דם כאנדרטאור

Service Locator מהווה Runtime linker

ב MVP שכבות המודל וה View צריכות להכיר רק את עצמן.

מיומנות עיצוב קוד (Design) וכתיבת קוד

שאלה 3: (20 נק)

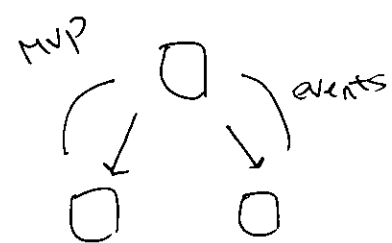
לצורך איתור ואבחון תקלות ברכבים אנו ממדלים את ההתנהגות הצפויה של כל רכיב ורכיב - כל רכיב
(Component) זוכה לאובייקט משלו. המחלקה WarningLight תומכת בפעולות on(), off() ו blink(). יש
לה מימושים שונים כמו LowFuel, CheckEngine וכו'. כאשר רכיב כלשהו הופך לתקול אז נורית האזהרה
צריכה לפעול בהתאם. לדוגמא, כאשר הרכיב TimingChain הופך לתקול אז ה CheckEngine צריכה להיות
דולקת, וה OverdriveOff צריכה להבהב. באופן דומה גם אם המנוע תקול אז CheckEngine צריכה לדלוק.
לכל נורת אזהרה קיימים תנאים שונים שתחתם עליה לפעול בצורה הרצויה.

המתכנתים של המחלקות מסוג WarningLight יודעים להגדיר את התנאים לפעולה הרצויה (להידלק,
להבהב, או להיכבות) אך הם אינם יודעים *מתי* ליזום את הפעולה הרצויה (כלומר מתי הרכיב הופך
לתקול)

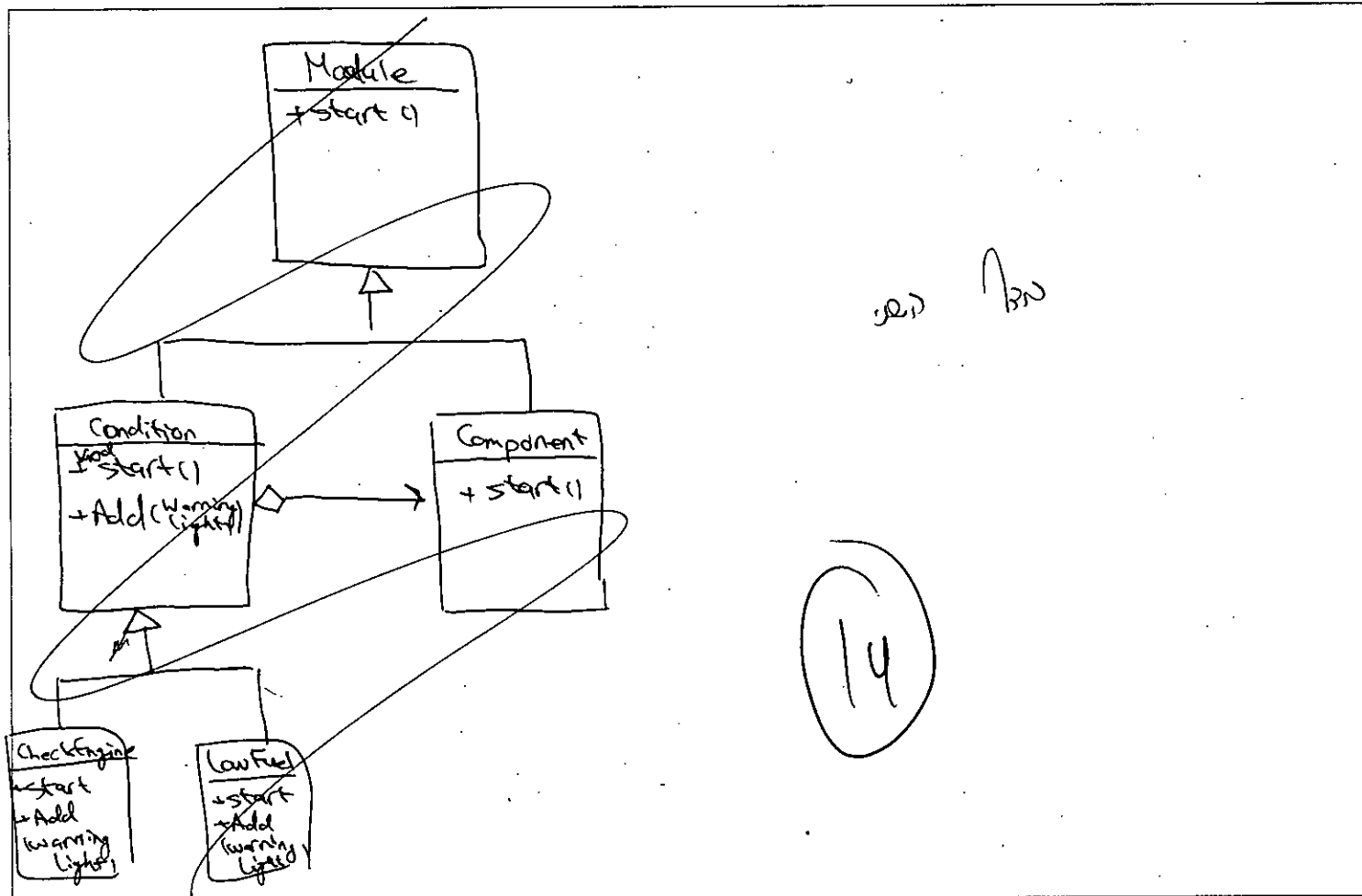
עליכם לשרטט תרשים מחלקות (class diagram) ב UML, המציג עיצוב שמאפשר ליזום את הפעולה
הרצויה של כל נורת אזהרה כאשר הרכיבים הרלוונטיים הופכים לתקולים (ללא busy waiting).

Active
object

סטור
לי נהיג



תשובה:



שאלה 4: ממשו בקוד (באיזו שפה מונחית עצמים שתמצאו) את הפתרון העיצובי של שאלה 3. (16 נק')

תשובה:

```

Public Interface Module {
    public string Start()
}

```

```

Public class Component implement Module() {
    public String start ( ) {
        // ...
    }
}

```

אם יש דמיון וקאן
אם יש

4

הפעלה
הכבה

public Interface Warning Light

On();
Off();
Blink();
Is On;
Is Off;
Is Blinking;
}

הפעלה והכבה
הפעלה והכבה
הפעלה והכבה

הפעלה והכבה

in eq. Add (5)

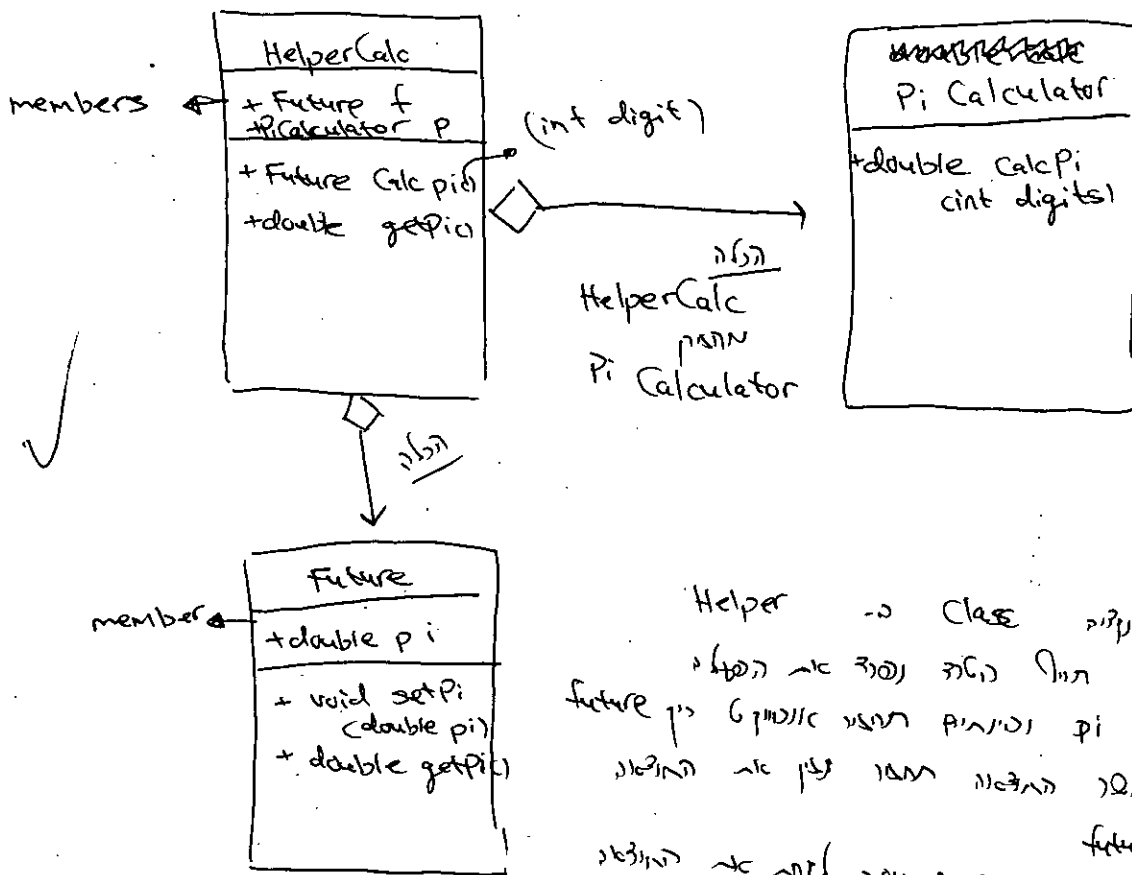
pend

שאלה 5: (16 נקודות)

נותנה המחלקה PiCalculator שממשה את המתודה double calcPi(int digits) המחשבת את פאי עד ל digits ספרות לאחר הנקודה. לצערנו זה עלול לקחת לא מעט זמן וזה קוד סגור שלא ניתן לשינויים. עליכם לשרטט תרשים מחלקות (class diagram) ב UML, המציג עיצוב שמאפשר

- להריץ את calcPi בצורה אסינכרונית - תשעה
- ולטפל בערך המוחזר כך ש
 - הלקוח יוכל לשלוף אותו בעת הצורך
 - ורק אם הלקוח מנסה לשלוף את הערך לפני הזמן הוא יאלץ להמתין עד לסוף החישוב.

תשובה:



הסימנים Class - Helper
 calc תורן הוצא נסר את הפא
 ו pi וסימנים תוצא אונטק בין future
 נאמר המצא תמו זמן את המצא
 future-
 וסמך, והממש יפה לקח את המצא
 נ future לסימן המן המן יונק
 ב שמו המצא

שאלה 6: ממשו בקוד (באיזו שפה מונחית עצמים שתמצו) את העיצוב של שאלה 5 (16 נק')

בהצלחה!

תשובה 6:

```
public class Helper Calc {
    private Future f; //member
    private PiCalculator p; //member
    private HelperCalc () {
        this.p = new PiCalculator(); // C-tor
        this.f = new Future()
    }
}
```

```
public Future Calc(int digits) {
    Thread t = new Thread( delegate {
        // ...
    });
    t.start();
}
```

```
public Future Calc(int digits) {
    Thread t = new Thread( delegate {
        (1) float pi = f.setPi( p.calcPi(digits));
    });
    t.start();
    return f;
}
```

```
public double getPi() { return f.getPi(); }
}
```



```

public class Future {
    private double pi; // member

    private Future() { this.pi = null; }

    public void setPi (double pi) {
        this.pi = pi;
        notify();
    }

    public double getPi() {
        if (pi == null) {
            wait; return pi; // guarded suspension
        }
        else
            return pi;
    }
}

```

main:

```

Helper c = new Helper (0);
Future f = c.calculate (3);
cf.getPi();

```

