

אוניברסיטת בן-גוריון

מדור בחינות

מספר נבחן: _____

תאריך הבחינה: 20.2.2003
שם המורה: ד"ר מיכאל אלחוד
מר אלעד שילר
מר מנחם אדלר
שם הקורס: תכנות מערכות
מספר הקורס: 202-1-2031
מיועד לתלמידי: מדעי המחשב, הנדסת תוכנה
שנה: תשס"ג
סמסטר: א'
מועד: ב'
משך הבחינה: שלש שעות
חומר עזר: אסור

רשום/רשמי תשובותיך בגיליון התשובות בלבד – תשובות מחוץ
לגיליון לא יבדקו.

בהצלחה!

נתונה מחלקה MyVector, מסונכרנת סינכרון מלא, שבנוסף לממשק בסיסי למחלקה Vector מכילה שיטה getEqualObjectsNum המקבלת כפרמטר ווקטור אחר, ומחזירה את מספר האובייקטים בעלי ערך שווה הנמצאים באותו אינדקס בשני הווקטורים:

```
import java.util.Vector;

class MyVector
{
    protected Vector data;

    public synchronized int size() {
        return data.size();
    }

    public synchronized void add(Object obj) {
        data.add(obj);
    }

    public synchronized void remove(int index) {
        data.remove(index);
    }

    public synchronized Object get(int index) {
        return data.get(index);
    }

    public synchronized int getEqualObjectsNum(MyVector vector)
        throws Exception
    {
        synchronized(vector) {
            if (size() != vector.size())
                throw new Exception();

            int EqualObjectsNum = 0;
            for (int i = 0; i < size(); i++)
                if (get(i).equals(vector.get(i)))
                    EqualObjectsNum++;

            return EqualObjectsNum;
        }
    }
}
```

א. תאר/י כיצד ביצוע השיטה getEqualItemsNum יכול לגרום ל'חבק' (deadlock) (7 נקודות)

ב. מתכנת מקורי שיכתב את השיטה `getEqualObjectsNum` באופן הבא:

```
public synchronized int getEqualObjectsNum(MyVector vector)
    throws Exception
{
    synchronized(vector) {
        if (size() != vector.size())
            throw new Exception();
    }

    int EqualObjectsNum = 0;
    for (int i = 0; i < size(); i++) {
        synchronized(vector) {
            if (get(i).equals(vector.get(i)))
                EqualObjectsNum++;
        }
    }
    return EqualObjectsNum;
}
```

האם ביצוע `getEqualObjectsNum` בטוח במודל חישוב מקבילי? הסבר/י. (5 נקודות)

ג. תקן/ני את המחלקה `MyVector` כך שהשיטה `getEqualItemsNum` לא תהיה כרוכה בסינכרון הפרמטר `vector` לכל משך הביצוע של לולאת ה `for`, ובכל זאת החישוב בטוח במודל חישוב מקבילי. (10 נקודות).

שאלה 2**(10 נקודות)**

```

class even {
    private int j_;

    public even() { j_ = 0; }
    public int getJ() { return j_; }
    public int next() {
        j_++;
        j_++;
        return j_;
    }
}

```

א. הגדרי את התכונה נשמרת (invariant) למחלקה even והראה/י תרחיש בו התכונה לא נשמרת תחת חישוב מקבילי. (3 נקודות)

ב. האם מחלקה safe בטוחה? נמק/י. (7 נקודות)

```

class safe {
    private even e_;
    public safe() { e_ = new even; }
    public synchronized even getE() { return e_; }
    public synchronized void setE(even e) { e_ = e; }
}

```

שאלה 3**(10 נקודות)**

נתון הקוד הבא ב-C++:

```

#include <iostream>

class A {
public:
    A(int i) { i_ = i; }
    int getI() {return i_;}
private:
    int i_;
};

A* makeA(int j) {
    A a(j);
    return &a;
}

void main()
{
    A* pa = makeA(7);
    A* pb = makeA(8);
    std::cout << pa->getI() << " " << pb->getI() << std::endl;
}

```

- א. מה מדפיסה main()? מה הבעיה בקוד? (5 נקודות)
ב. תקן/ני את הקוד כך שתובטח הרצה ללא שגיאה. (5 נקודות)

15 נקודות)

שאלה 4

```
#include <vector>

class A {
private:
    std::vector<int> v_;

public:
    A(int size = 100, int init_val = 100) {
        for (int i=0; i<size; i++) {
            v_.push_back(init_val);
        }
    }
    int get(int i) { return v_[i]; }
    void set(int i, int v) { v_[i] = v; }
};

void doA(std::vector<A> v) {
    std::vector<A>::iterator i;
    for (i = v.begin(); i != v.end(); ++i) {
        A current = *i;
        std::cout << "A " << (*i).get(0) << std::endl;
        current.set(0, 200);
        *i = current;
    }
    for (i = v.begin(); i != v.end(); ++i) {
        std::cout << "A " << (*i).get(0) << std::endl;
    }
}

void main() {
    A a(10000, 10);
    std::vector<A> v1;

    v1.push_back(a);
    v1.push_back(a);

    v1[0].set(0, 20);
    v1[1].set(0, 30);

    doA(v1);
}
```

- א. מה יודפס כתוצאה מהרצת הפונקציה main()? (7 נקודות)
ב. הסבר/י מדוע הפונקציה doA אינה יעילה, ושנה/י אותה כך שתהיה יעילה. (8 נקודות)

נתונה מחלקה IncNumber המממשת את הממשק Increasable באופן הבא:

```
public interface Increasable
    extends java.io.Serializable
{
    public void increase();
}

class IncNumber
    implements Increasable
{
    protected long num_;
    protected long offset_;

    public IncNumber(long n, long o) {
        num_ = n;
        offset_ = o;
    }

    public synchronized void increase() {
        num_ += offset_;
    }

    public long getNum() { return num_; }
}
```

ObjectOutputStream היא מחלקה ב Java היורשת את המחלקה OutputStream. בשאלה זו נשתמש בשיטה אחת שהיא מוסיפה על OutputStream:

```
void writeObject(Object obj)
```

השיטה מבצעת serialization על האובייקט obj וכותבת את תאורו ב byte code ל ObjectOutputStream.

באותו אופן ObjectInputStream היא מחלקה ב Java היורשת את המחלקה InputStream. בשאלה זו נשתמש בשיטה אחת שהיא מוסיפה על InputStream:

```
Object readObject()
```

השיטה קוראת מה ObjectInputStream תאור של אובייקט ב byte code מבצעת deserialization ומחזירה את האובייקט שמתקבל.

להלן תיאורם של client ושל server:

```
import java.io.ObjectOutputStream;
import java.net.Socket;

public class client {
    public static void main(String[] args) throws Exception {
        Socket s = null;

        IncNumber element = new IncNumber(2,7);
```

```

Socket server = new Socket(args[0], 2000);
ObjectOutputStream out =
    new ObjectOutputStream(server.getOutputStream());

out.writeObject(element);

System.out.println(element.getNum());

server.close();
out.close();
}
}

```

```

import java.io.ObjectInputStream;
import java.net.Socket;
import java.net.ServerSocket;

public class server {
    public static void main(String[] args) throws Exception {

        ServerSocket serverSocket = new ServerSocket(2000);
        while (true) {
            Socket client = serverSocket.accept();
            ObjectInputStream in =
                new ObjectInputStream(client.getInputStream());
            IncNumber element = (IncNumber) in.readObject();

            for (int i=0; i<3; i++)
                element.increase();

            in.close();
            client.close();
        }
    }
}

```

- א. עם כמה לקוחות יכול השרת לנהל קשר (connection) במקביל? (3 נקודות)
 ב. איזה סוג של קשר נוצר בין השרת ללקוח (TCP או UDP)? (3 נקודות)
 ג. ציין/ני, עבור הקשר שבחרת בסעיף ב, האם התכונה מתקיימת עבורו (כן או לא):
 (6 נקודות)

- Unicast
- Duplex communication
- Infallible
- Reliability
- Checksum
- Retransmissions

- ד. עדכן/ני את הקוד ב client וב server כך שהערך של element המודפס ב client יהיה מעודכן
 (10 נקודות)

(9 נקודות)

שאלה 6

נתונות שלוש אפליקציות המשתמשות במודל אירועים:

אפליקציה 1: סוחרי מניות בבורסה נרשמים כלקוחות של שרת אירועים. שרת האירועים מייצר אירועים מסוג: 'מניה מסוימת עברה סף של סכום נתון'. לדוגמא: סוחר מניות נרשם כלקוח כדי לקבל עדכונים על כך שמניית IBM עברה את מחסום ה-\$100.

אפליקציה 2: מערכת לגיבוי קבצים נרשמת כלקוחה לשרת אירועים. השרת מייצר אירועים מסוג 'התעדכן קובץ מסוים בספריה מסוימת'. נתון כי מערכת גיבוי הקבצים מגבה קבצים שהתעדכנו רק פעם אחת ביום.

אפליקציה 3: לקוחות של שרת דואר אלקטרוני, נרשמים לשרת אירועים, כדי להתעדכן על כל דואר שיגיע. השרת מייצר אירועים מסוג 'נתקבל דואר אלקטרוני'.

עבור כל אפליקציה, סמן/ני בטבלה בדף התשובות אלו מאפיינים של מודל אירועים נדרשים עבורה (כלומר האם המאפיין חיוני עבור הלקוחות של מודל האירועים):

- **Push/Pull**
האם שרת האירועים יוזם את שליחת האירוע (push) או שהאירוע נשלח רק כתגובה לבקשת הלקוח (pull).
- **Persistence**
האם השרת צריך להיות מסוגל לעדכן לקוחות שאינם 'באוויר' בזמן התרחשות האירוע, לכשיתעוררו (אם המודל הוא push) או כאשר ירצו להתעדכן (אם המודל הוא pull).
- **Quality of Service**
האם נדרש מהשרת להתחייב על פרק זמן מקסימאלי מהתרחשות האירוע ועד חשיפתו ללקוח.

(12 נקודות)

שאלה 7

לצורך קביעת הסרט המוקרן בכל יום נעזר ה'סינמטק' הבאר שבעי בתוכנת שיבוץ. התוכנה מורכבת מהרכיבים הבאים:

MovieScheduler – התוכנית הראשית - כוללת מימוש של אלגוריתם המציע אפשרות שיבוץ של סרטים לכל יום מימי החודש.
UserInterface – ספריה עם פונקציות לטיפול בממשק הגרפי עם המשתמש.
ReportGenerator – ספריה עם פונקציות להפקת דוחות מפורטים על לוח הסרטים.

במשך כל החודש, בוחן אחראי התוכן בחברה, בעזרת הממשק למשתמש, אפשרויות שיבוץ שונות לחודש הבא. בסוף החודש הוא בוחר את השיבוץ האהוב עליו ורק אז מפיק את הדו"ח – חוברת הסרטים לחודש הבא.

מעצבי תוכנת השיבוץ החליטו שהתוכנה תכיל קובץ הרצה MovieScheduler.exe המכיל את הרכיב MovieScheduler והמשתמש בשתי ספריות דינאמיות UserInterface ו-ReportGenerator.

- א. איזה סוג של לינק דינאמי (Explicit/Implicit) הייתה בוחרת לכל אחת משתי הספריות הדינאמיות? נמק. (6 נקודות)
- ב. מדוע לא כדאי לבצע לינק סטטי על שתי הספריות? (6 נקודות)