אוניברסיטת בן-גוריון

מדור בחינות

תאריך הבחינה: 27.2.2004
שם המורה: ד"ר מיכאל אלחדד
מר מנחם אדלר
שם הקורס: תכנות מערכות
מספר הקורס: 202-1-2031
מיועד לתלמידי: מדעי המחשב, הנדסת תוכנה
שנה: תשס"ד
סמסטר: א'
מועד: ב'
משך הבחינה: שלש שעות
חומר עזר: אסור

מספר נבחן:_____

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

בהצלחה!

שאלה 1 **(30 נקודות)**

בשאלה זו אנו מדמים שרות חדרים בבית מלון. מספר החדרים במלון קבוע, כל חדר מחדרי המלון יכול להזמין שרות לחדרו מהקבלה. הוא יכול אף להזמין מספר שרותים בזה אחר זה, תוך ציפייה לקבלת כל השירותים לא בהכרח על ידי אותו אדם.
ההדמיה מורכבת משתי מחלקות: אובייקט פסיבי Room המדמה חדר בבית מלון, ואובייקט אקטיבי (= thread) RoomService המדמה את נותן שרות החדרים.

להלן תאור המחלקה Room:

```
class Room {
   private String Id_;
   private boolean serviceNeeded_;
   public Room(String name) { Id_ = name; serviceNeeded_ = false; }
   public String synchronized getName(){return Id_; }
   public boolean synchronized isServiceNeeded() {return serviceNeeded_;}
   public void synchronized requestService() { serviceNeeded_ = true; }
   public void synchronized serviceDone() { serviceNeeded_ = false; }
}
```

המחלקה RoomService מגדירה Thread מעל ווקטור של חדרים, כאשר באחריות המתודה ()run לספק את השרות לכל אחד מהחדרים שהזמין שרות (= החזרת ערך true במתודה ()isServiceNeeded שהוא מממש).
ניתן להניח כי ווקטור החדרים לא משתנה – כלומר לא נוספים אליו או מוסרים ממנו חדרים.

השיטה service מבצעת בפועל את השרות הנדרש עבור החדר (אופן מימושה אינו רלבנטי לשאלה זו)

```
class RoomService extends Thread {
    private Vector rooms_;
    RoomService(Vector rooms)  { rooms_ = rooms; }
    public void run()  {
        try {
            while (true) {
                sleep(1000);
                for (int j=0;  j<rooms_.size();  j++)   {
                    Room room = (Room)rooms_.get(j);
                    if (room.isServiceNeeded())
                        service(room);
                }
            }
        } catch (java.lang.InterruptedException ie) {
            return;
        }
    }
    private void service(Room room)    {
      // service is performed here…
      room.serviceDone();
      System.out.println("Room " + room.getName() + " is serviced");
    }
    public static void main(String args[]) {
        Vector rooms = new Vector();
        rooms.add(new Room("room 0"));
        rooms.add(new Room("room 1"));
        for (int i = 0; i < rooms.size(); i++) {
            ((Room)rooms.get(i)).requestService();
        }
        RoomService S1 = new RoomService(rooms);
        S1.start();
        ((Room)rooms.get(1)).requestService();
        try {
            sleep(2000);
        } catch (java.lang.InterruptedException ie) {
        }
        S1.interrupt();
   }
}
```

א. מהו הפלט של התוכנית? (10 נקודות)

ב. עדכן/י את המחלקה Room כך שאם הועלו מספר בקשות למתן שרות (על ידי השיטה requestService) הן תבוצענה, כולן, על ידי ה RoomService.
אין לשנות את טיפוסי השדות ו/או להוסיף שדות חדשים לשתי המחלקות. (10 נקודות)

ג. עדכן/י את הקוד כך שאם מבוצע interrupt על הת'רד של שרות החדרים, הוא לא יפסיק פעולתו מיד (כך המצב בקוד הנוכחי) אלא אחר שיינתן השרות לכל הבקשות שהועלו, וכן לבקשות הממתינות להעלאה.
ניתן להוסיף שדות שאינם מטיפוס מערך, ווקטור רשימה וכדומה למחלקות השונות (10 נקודות)

שאלה 2 (15 נקודות)

נתונה הגדרתן של המחלקות A ו B ב ++C:

```
class A {
public:
```

```
      A() { p_ = new int(0); }
      virtual ~A() {  std::cout << "A: destructor"; delete p_; }
protected:
   int* p_;
};

class B : public A {
public:
      B(int i) { p_ = new int(i); }
      virtual ~B() {  std::cout << "B: destructor"; }
};
```

א. מה מדפיסה התוכנית הבאה? (5 נקודות)

```
{
   A*  pa = new B(2);
   delete pa;
}
```

ב. האם כל הזיכרון המוקצה בקטע הקוד הנ"ל משתחרר? הסבר/י.
במידה וקיימת נזילת זכרון שנה/י את ההגדרה של אחת המחלקות כך שנזילת הזכרון לא תתרחש, מבלי להוסיף פקודת delete ו/או destructor נוסף (10 נקודות)

שאלה 3 (15 נקודות)

א. נתונות שתי פונקציות ב C++.
עבור כל אחת מהפונקציות, אבחן/ני, לעצמך, מה היא מבצעת:
- אם ניתן להמירה ל Java כך שהיא מבצעת בסופו של דבר את משימתה - גם במחיר שינוי טיפוסי המשתנים - המר/המירי אותה ל Java.
- אם הדבר לא ניתן להעשות הסבר/י מדוע. (10 נקודות)

א.1.

```
boolean F1(int& i) {
      return ++i < 100;
}
```

א.2.

```
boolean F2(int* i) {
      return ++i < 100;
}
```

ב. המר/י את קטע הקוד הבא מ Java ל C++ (5 נקודות):

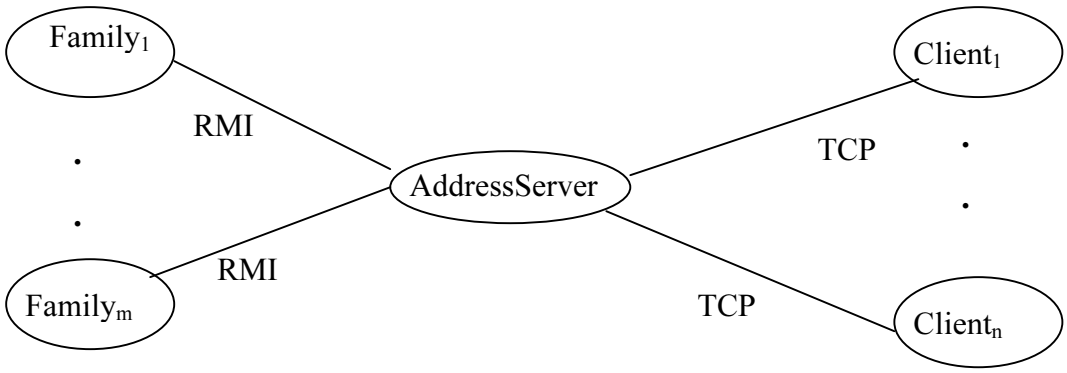
```
interface I  {
   boolean F();
}

class A {
   A(int i)   { i_ = i; }
   protected int  i_;
}

class B extends A implements I {
   B(int i) { super(i); }
   public boolean F()   {   return i_ < 100; }
}
```

שאלה 4	(15 נקודות)

AddressBook הינה מערכת המספקת כתובת וטלפון של אנשים על פי דרישה.
ניתן לפנות לשרת הכתובות (AddressServer) בבקשה לקבלת הכתובת והטלפון על פי שם המשפחה. השרת מחזיק טבלה הקושרת שם משפחה ל RemoteObject הממשש את הממשק RemoteAddressInterface. כאשר מגיעה בקשה לכתובת וטלפון של משפחה הוא פונה ל RemoteObject המתאים, מקבל את הפרטים דרך הממשק RemoteAddressInterface ומספק אותם ללקוח.
הארכיטקטורה של המערכת מתוארת בדיאגרמה הבאה:



הממשק RemoteAddressInterface מוגדר באופן הבא:

```
public interface RemoteAddressInterface extends java.rmi.Remote
{
 public String getPhoneNumber() throws java.rmi.RemoteException;
 public String getAddress() throws java.rmi.RemoteException;
}
```

להלן הגדרת המתודה ()getData ב AddressServer המקבלת שם משפחה ומחזירה מחרוזת המכילה את הנתונים שלה – כתובת ומספר טלפון.
עבור משפחה לא ידועה נזרק UnknownFamilyException.
<u>הערה</u>: הטבלה mapFamilyname2RemoteAddress - מטיפוס HashMap - ממפה שם משפחה ל RemoteObject הממשש את הממשק RemoteAddressInterface. המתודה get של HashMap מחזירה עבור מפתח נתון את הערך הממופה לו (ובמקרה שלנו עבור שם משפחה, את ה RemoteObject שלו) אם המפתח לא קיים (במקרה שלנו אם שם המשפחה לא מופיע בשרת הכתובות) היא מחזירה null.

```
Public String getData(String familyName)
          throws UnknownFamilyException, java.rmi.RemoteException
{
   RemoteAddressInterface  family;
   family = mapFamilyname2RemoteAddress.get(familyName);
   if (family == null)
      throw new UnknownFamilyException(familyName);

   return "Family: " + familyName +
               ",   Address: " +  family.getAddress() +
               ",   Phone: " + family.getPhoneNumber();
}
```

א. כמה פעולות תקשורת – כלומר מעברים של הלוך/חזור מתהליך לתהליך – מתרחשות כתוצאה מביצוע ()getData? פרט/י (5 נקודות)

ב. שנה/י את הממשק RemoteAddressInterface ואת המתודה getData ב AddressServer המשתמשת בו, כך ששם המשפחה ומספר הטלפון יתקבלו על ידי פעולת תקשורת אחת (הלוך/חזור)? (5 נקודות)

ג. עקב דרישת הקהל הורחבה המערכת, וכעת נדרש, מעבר לכתובת ולטלפון, גם כתובת ה email שלה. עדכן/י את הממשק RemoteAddressInterface כך שיתמוך בקבלת הנתון הנוסף:

(i) על פי הגישה המקורית

(ii) על פי הגישה בה הלכת בסעיף הקודם

וציין/י את היתרון והחסרון בכל שיטה. (5 נקודות)

שאלה 5	**(15 נקודות)**

נתונים שני תהליכים P1 ו P2 המקיימים ביניהם קשר TCP המבטיח אמינות בעזרת אלגוריתם Go-Back-N עבור N=100.

P1 שולח הודעה ל P2, שכבת ה TCP שלו מחלקת את ההודעה ל 87 יחידות ושולחת אותן ל שכבת ה TCP של P2.

א. כיצד יודעת שכבת ה TCP של P2 כי יחידה התקבלה ללא שגיאות? (5 נקודות)

ב. נתון כי מלבד היחידה העשירית כל היחידות הגיעו ללא שיבושים לשכבת ה TCP של P2. כמה יחידות ישלחו שנית ב timeout הבא? (5 נקודות)

ג. הציע/י דרך בה תישלח בשנית ב timeout הבא רק יחידה אחת (5 נקודות)

שאלה 6	**(10 נקודות)**

וועדת התכנון בקיבוץ הציעה שני מודלי נתונים המתארים את תפוקת החלב היומית של הפרות ברפת ואת המשקל היומי של העגלים לפיטום:

הצעה ראשונה:

Cows

ID
NickName
BornDate

Primary Key: ID

Production

ID
Year
DayInYear
MilkQuantity

Primary Key: ID + Year + Day In Year
Foreign Key: ID

Weight

ID
Year
DayInYear
Weight

Primary Key: ID + Year + Day In Year
Foreign Key: ID

הצעה שניה:

Cows

ID
NickName
BornDate
Type

Primary Key: ID

כאשר השדה type מציין האם מדובר בפרה (type = 1) או בעגל (type = 2)

Production

ID
Year
DayInYear
Data

Primary Key: ID + Year + Day In Year
Foreign Key: ID

כאשר השדה Data מציין את תפוקת החלב בליטרים עבור פרות ואת המשקל בק"ג עבור עגלים.

כתובי/כתבי שאילתת SQL המציגה את שמות הפרות שהניבו 70 ליטר חלב ביום השלישי של שנת 2003 ואת שמות העגלים ששקלו 134 ק"ג ביום זה:

א. על פי המודל הראשון. (5 נקודות)
ב. על פי המודל השני. (5 נקודות)

בהצלחה!