

ביה"ס למנהל עסקים

מבחן בתכנות JAVA סימסטר ב' מועד א'

המרצה: רועי כנעני

תאריך: 26.7.99

משך המבחן: שעתיים וחצי. כל חומר כתוב מותר בשימוש.

לפניך חלק מתוכנת עזר המבצעת ניתוח של קשרים עסקיים בין חנויות לספקים בעזרת המחלקה Relations והאפלט TestRelations.

עליך לכתוב 3 שיטות עבור המחלקה Relations ושיטה אחת עבור האפלט TestRelations.

תאור המחלקה Relations

המחלקה מכילה מערך דו ממדי בשם table המכיל ערכים בולאנים המייצגים קשרים עסקיים בין חנויות לספקים.

במערך 100 שורות שכל אחת מהן מייצגת חנות (מספרי החנויות הם 0-99). המערך מכיל 20

עמודות שכל אחת מהן מייצגת ספק (קוד הספקים 0-19).

אם יש קשר עסקי בין החברה ה- i לספק ה- j יופיע במערך הדו ממדי בשורה ה- i והעמודה ה- j הערך true, ואם לא קיים קשר עסקי יופיע בתא זה הערך false.

* בנוסף למערך הדו ממדי table, מכילה המחלקה מערך חד ממדי בשם supset המייצג קבוצה של ספקים. במערך 20 תאים בולאנים (תא לכל ספק) כאשר התא ה- i מכיל true אם הספק שהקוד שלו i נמצא בקבוצה, ו- false אחרת.

להלן הגדרות של מחלקות, משתנים ושיטות משני קבצים בהם תוכלו להשתמש (אין צורך

להעתיק אותם):

הקובץ TestRelations.java

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;

public class TestRelations extends Applet implements ActionListener {

    Relations r = new Relations(); // object definition

    Label supLabel, storeLabel, maxLabel;
    TextField supField, storeField;
    Button computeMax, intersection;

    public void init() {
        supLabel = new Label("Enter supplier code:");
        add(supLabel);
        supField = new TextField(2);
        add(supField);
        storeLabel = new Label("Enter store code:");
        add(storeLabel);
        storeField = new TextField(2);
        add(storeField);
        computeMax = new Button("compute max");
        add(computeMax);
        maxLabel = new Label(" ");
        add(maxLabel);
        intersection = new Button("intersection");
        add(intersection);
        supField.addActionListener(this);
        computeMax.addActionListener(this);
        intersection.addActionListener(this);
    }

    public void paint(Graphics g)
    {
        r.print_supset(g);
    }

    public void actionPerformed(ActionEvent e)
    { //...
```

```
import java.awt.*;
```

```

public class Relations {
    private final int STORES = 100;
    private final int SUPPLIERS = 20;
    private boolean [][] table;
    private boolean [] supset;

    public Relations() // constructor
    {
        table = new boolean [STORES] [SUPPLIERS];
        supset = new boolean [SUPPLIERS];
    }

    // print the suppliers intersection set array
    public void print_supset(Graphics g)
    {
        ...
    }
}

```

Relations.java הקובץ

המשימות:

public void addRelation(int i , int j) : Relations המחלקה עבור שיטה עבור המחלקה (1 (10%)
 שמקבלת קוד חנות i , וקוד ספק j ומוסיפה קשר עיסקי ביניהם בטבלה.

public int maxStore() : Relations המחלקה עבור שיטה עבור המחלקה (2 (30%)
 שמחזירה את קוד החנות שמקיימת קשרים עיסקיים עם המספר הגדול ביותר של ספקים (אם יש יותר מאחת כזו - החזר את קוד החנות הראשונה).

public void intersection() : Relations המחלקה עבור שיטה עבור המחלקה (3 (30%)
 שמסמנת במערך החד מימדי supset את קבוצת הספקים המקיימים קשרים עסקיים עם כל החנויות.
 השיטה תסמן true בתא ה i במערך supset אם הספק ה i מקיים קשר עיסקי עם כל החנויות,
 ו false אחרת.
 (מערך קבוצת הספקים מאותחל בערכים בולאנים שרירותיים לא ידועים ויש לטפל בכך).

public void actionPerformed(ActionEvent e) : TestRelations המופעלת עבור האפלט (4 (30%)
 המופעלת בשלושה מיקרים שונים :
 א) אם המקור למאורע הוא השדה supField , השיטה תקלוט את מספר החנות מהשדה storeField ואת מספר הספק מהשדה supField ותסמן קשר עיסקי חדש בטבלה.
 ב) לחיצה על הכפתור computeMax , תיקרא לחישוב הנדרש בשאלה 2 ותדפיס את התוצאות ע"י התווית maxLabel .
 ג) לחיצה על הכפתור intersection , תיקרא לחישוב הנדרש בשאלה 3 ותדפיס את התוצאות ע"י הפעלת השיטה print_supset(Graphics g) של המחלקה Relations (הניחו שהשיטה print_supset כבר כתובה ואינכם צריכים לכתוב אותה בעצמכם).

בהצלחה !

* output קובץ :

