מדינת ישראל
משרד החינוך

סוג הבחינה: בגרות
מועד הבחינה: קיץ תשע"ח, 2018
מספר השאלון: **899381**
תרגום לערבית (2)

مدعي المحسب

הוראות לנבחן

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה:
בשאלון זה שלושה פרקים.
פרק ראשון: בפרק זה שלוש שאלות, ענה על פי ההוראות בפרק.
(10x1)+(15x1) — 25 נק'
פרק שני: בפרק זה שלוש שאלות, ומהן עליך לענות על שתיים.
(25x2) — 50 נק'
פרק שלישי: בפרק זה שאלות בארבעה מסלולים שונים. ענה על שאלה אחת במסלול שלמדת.
(25x1) — 25 נק'
סה"כ — 100 נק'

ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשב הניתן לתכנות.

ד. הוראות מיוחדות:
1. את כל התוכניות שאתה נדרש לכתוב בשפת מחשב בפרקים הראשון והשני, כתוב בשפה אחת בלבד — Java או C#.
2. **רשום על הכריכה החיצונית** של המחברת באיזו **שפה** אתה כותב — Java או C#.
3. **רשום על הכריכה החיצונית** של המחברת את שם ה**מסלול שלמדת**. המסלול הוא אחד מארבעת המסלולים האלה: מערכות מחשב ואסמבלי, מבוא לחקר ביצועים, מודלים חישוביים, תכנות מונחה עצמים.

הערה: בתוכניות שאתה כותב לא יורדו לך נקודות, אם תכתוב אות גדולה במקום אות קטנה או להפך.

دولة إسرائيل
وزارة التّربية والتّعليم

نوع الامتحان: بچروت
موعد الامتحان: صيف 2018
رقم النّموذج: **899381**
ترجمة إلى العربيّة (2)

علم الحاسوب

تعليمات للممتحَن

أ. مدّة الامتحان: ثلاث ساعات.

ب. مبنى النّموذج وتوزيع الدّرجات:
في هذا النّموذج ثلاثة فصول.
الفصل الأوّل: في هذا الفصل ثلاثة أسئلة، أجب حسب التعليمات في الفصل.
(10x1)+(15x1) – 25 درجة
الفصل الثّاني: في هذا الفصل ثلاثة أسئلة، عليك الإجابة عن اثنين منها.
(25x2) – 50 درجة
الفصل الثّالث: في هذا الفصل أسئلة في أربعة مسارات مختلفة. أجب عن سؤال واحد في المسار الذي تعلّمتَه.
(25x1) – 25 درجة
المجموع – 100 درجة

جـ. موادّ مساعدة يُسمح استعمالها: كلّ مادّة مساعدة، عدا الحاسوب الذي يمكن برمجته.

د. تعليمات خاصّة:
1. اكتب جميع البرامج التي يُطلب منك كتابتها بلغة حاسوب في الفصلين الأوّل والثاني، بلغة واحدة فقط – Java أو C#.
2. **سجّل على الغلاف الخارجيّ** للدّفتر بأيّة **لغة** تكتب – Java أو C#.
3. **سجّل على الغلاف الخارجيّ** للدّفتر، اسم **المسار الذي تعلّمتَه**. المسار هو أحد المسارات الأربعة التّالية: أنظمة حاسوب وأسمبلي، مقدّمة لبحث الأداءات، موديلات حسابيّة، برمجة موجّهة كائنات.

ملاحظة: في البرامج التي تكتبها، لن تُخصم درجات إذا كتبتَ حرفًا كبيرًا بدلاً من حرف صغير أو بالعكس.

اكتب في دفتر الامتحان فقط، في صفحات خاصّة، كلّ ما تريد كتابته مسوّدة (رؤوس أقلام، عمليّات حسابيّة، وما شابه).
اكتب كلمة "مسوّدة" في بداية كلّ صفحة تستعملها مسوّدة. كتابة أيّة مسوّدة على أوراق خارج دفتر الامتحان قد تسبّب إلغاء الامتحان!

التّعليمات في هذا النّموذج مكتوبة بصيغة المذكّر وموجّهة للممتحَنات وللممتحَنين على حدّ سواء.

ب ה צ ל ח ה ! **نتمنّى لك النّجاح!**

الأسئلة

في هذا النموذج ثلاثة فصول.

عليك الإجابة عن أسئلة من ثلاثة الفصول، حسب التعليمات في كلّ فصل.

الفصل الأوّل (25 درجة)

ملاحظة: في كلّ سؤال يُطلب فيه استقبال، لا حاجة لفحص سلامة المدخَلات.
للّذين يحلّون بلغة Java: في كلّ سؤال يُطلب فيه استقبال، افترِضْ أنّ الأمر التالي مكتوب في البرنامج:

```
Scanner input = new Scanner (System.in);
```

أجب عن السؤال 1 – إلزاميّ (10 درجات).

1. معطاة الفئة **AllNumbers** التي لها صفة واحدة: مصفوفة أحاديّة الأبعاد — arrayNum ، من نمط صحيح.
توجد في المصفوفة أعداد موجبة أكبر من صفر، قسم منها أعداد زوجيّة وقسم منها أعداد فرديّة.
اكتب عمليّة داخليّة lastOddValue بلغة Java أو LastOddValue بلغة C# ، تُعيد قيمة العدد الفرديّ الأخير في المصفوفة.
مثال: بالنسبة للمصفوفة arrayNum بِكبَر 6 التي أمامك، تُعيد العمليّة العدد 3 ، الذي هو قيمة العدد الفرديّ الأخير في المصفوفة.

	0	1	2	3	4	5
arrayNum	7	5	8	9	**3**	4

افترض أنه يوجد في المصفوفة عدد فرديّ واحد على الأقلّ.

/ يتبع في صفحة 3 /

أجب عن أحد السؤالين 2-3 (15 درجة) .

2. في مسابقة للغناء تتنافس 40 أغنية أرقامها من 1 حتّى 40. يُدرِّج الأغاني 25 حَكَمًا.
يختار كلّ حَكَم أفضل ثلاث أغنيات حسب رأيه. يَمنَح الحَكَم للأغنية في المرتبة الأولى (الأفضل) 7 نقاط،
ويَمنَح للأغنية في المرتبة الثانية 5 نقاط، ويَمنَح للأغنية في المرتبة الثالثة نقطة واحدة.
الأغنية التي تُجمِّع أعلى عدد من النقاط من جميع الحكّام تفوز بالمرتبة الأولى.
تحضيرًا لمسابقة الغناء عُرِّفت الفئة **Vote**.
للفئة ثلاث صفات:
first — رقم الأغنية التي درّجها الحَكَم في المرتبة الأولى، من نمط صحيح.
second — رقم الأغنية التي درّجها الحَكَم في المرتبة الثانية، من نمط صحيح.
third — رقم الأغنية التي درّجها الحَكَم في المرتبة الثالثة، من نمط صحيح.
افترض أنّه تمّ لكلّ صفة تعريف عمليّتَي get وَ set بلغة Java وعمليّتَي Get وَ Set بلغة C#.
اكتب عمليّة خارجيّة theWinner بلغة Java أو TheWinner بلغة C# ، تتلقّى مصفوفة من نمط **Vote**
لتدريجات الحكّام، وتطبع رقم الأغنية الفائزة بالمرتبة الأولى.
افترض أنّه توجد أغنية فائزة واحدة.
افترض أنّ قيَم المصفوفة سليمة.

/ يتبع في صفحة 4 /

3. معطاة الفئة **Time** التي لها صفتان:

hour – تمثّل ساعة بين 0 وَ 23 ، بما في ذلك 0 وَ 23، من نمط صحيح.

minute – تمثّل دقيقة بين 0 وَ 59 ، بما في ذلك 0 وَ 59، من نمط صحيح.

افترِضْ أنّه تمّ لكلّ صفة تعريف عمليّتَي get وَ set بلغة Java ، وعمليّتَي Get وَ Set بلغة C#.

أ. اكتب بلغة Java أو بلغة C# في الفئة **Time** عمليّة بنائيّة تتلقّى قيَمًا بالنسبة **لكلّ** صفة.

بالنسبة لكلّ صفة – إذا كانت القيمة المتلقّاة غير ممكنة حسب تعريفات الصفة، فإنّ القيمة تُستبدَل بـ 0.

ب. معطاة الفئة **Flight** التي لها أربع صفات: اسم شركة الطيران – name من نمط نصّ،

اسم بلد الغاية – destination من نمط نصّ، كود الرحلة – flightCode من نمط نصّ،

زمن الرحلة – flightTime من نمط **Time**.

اكتب بلغة Java أو بلغة C# عنوان الفئة وصفات الفئة.

جـ. معطاة الفئة **Airport** التي لها صفة واحدة: مصفوفة أحاديّة الأبعاد – flights من نمط **Flight**.

افترِضْ أنّه تمّ لكلّ صفة في الفئة **Flight** تعريف عمليّتَي get وَ set بلغة Java، وعمليّتَي Get وَ Set بلغة C#.

في الفئة **Airport** معطاة عمليّة داخليّة بوليانيّة، isFly بلغة Java وَ IsFly بلغة C#، هدفها فحص إذا كانت في مصفوفة الرحلات flights رحلة لشركة الطيران "Sky". تُعيد العمليّة true إذا كانت في المصفوفة flights رحلة لشركة الطيران "Sky". خلاف ذلك – تُعيد العمليّة false.

افترِضْ أنّ جميع الخلايا في المصفوفة ليست null.

العمليّة المعطاة التي أمامك (isFly بلغة Java وَ IsFly بلغة C#) **خاطئة**.

Java

```
public boolean isFly()
{
    for  (int i = 0; i < this.flights.length; i++)
    {
        if (this.flights[i].getName().equals ("Sky"))
            return true;
        return false;
    }
}
```

C#

```
public bool IsFly()
{
    for (int i = 0; i < this.flights.Length; i++)
    {
        if (this.flights[i].GetName() == "Sky")
            return true;
        return false;
    }
}
```

(انتبه: البنود الفرعيّة للبند "جـ" في الصفحة التالية.)

/ يتبع في صفحة 5 /

(1) اكتب ماذا تعُيد العمليّة **الخاطئة** المعطاة بالنسبة للمصفوفة flights بِكبَر 4، التي قيَم صفة name للكائنات في المصفوفة هي:

في المكان الأوّل: "Cloud" ،

في المكان الثاني: "Air" ،

في المكان الثالث: "Sky" ،

في المكان الرابع: "Travel" .

(2) اشرح ما هو الخطأ في العمليّة.

(3) صحِّح العمليّة كي تنفّذ المطلوب، وانسخ العمليّة المصحَّحة إلى دفترك.

/ يتبع في صفحة 6 /

الفصل الثاني (50 درجة)

انتبه: في كلّ سؤال يُطلب فيه تطبيق، بإمكانك استعمال عمليّات الفئات: دور، وراصّة، وشجرة بيناريّة وحلقة، بدون تطبيقها. إذا استعملتَ عمليّات إضافيّة، عليك تطبيقها.

أجب عن اثنين من الأسئلة 4-6 (لكلّ سؤال – 25 درجة).

4. **أ.** اكتب عمليّة خارجيّة lastAndRemove بلغة Java أو LastAndRemove بلغة C# ، تتلقّى راصّة من نمط صحيح، وتمحو الحدّ الأسفل في الراصّة، وتُعيد قيمته.
في نهاية العمليّة تبقى الحدود الأخرى في الراصّة بدون تغيير.
افترِضْ أنّ الراصّة ليست فارغة.

مثال، بالنسبة للراصّة التي أمامك:

1	← رأس الراصّة
6	
32	
5	
5	
7	
4	
9	

بعد العمليّة تُعاد القيمة 9 ، وتبدو المصفوفة هكذا:

1	← رأس الراصّة
6	
32	
5	
5	
7	
4	

(انتبه: البند "ب" في الصفحة التالية.)

/يتبع في صفحة 7/

ب. معطاة الفئة:

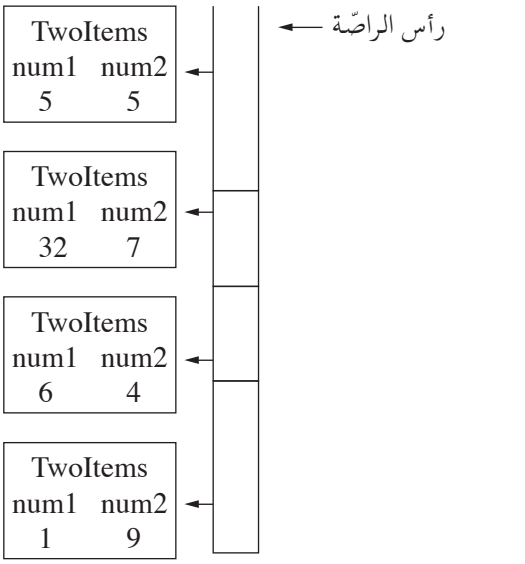
TwoItems	
الصفات:	int num1 int num2
عمليّة بنائيّة:	TwoItems (int number1, int number2)

افترِضْ أنّه تمّ لكلّ صفة تعريف عمليّتَي get وَ set بلغة Java ، وعمليّتَي Get وَ Set بلغة C# .
اكتب عمليّة خارجيّة stackTwoItems بلغة Java أو StackTwoItems بلغة C# ، تتلقّى راصّة stk1 ليست فارغة من نمط صحيح وبِكبَر زوجيّ، وتُعيد راصّة من نمط TwoItems .

الحدّ الأسفل في الراصّة المُعادة يحوي توجيهًا إلى ظهور TwoItems الذي صفته num1 هي الحدّ الذي في رأس الراصّة stk1 ، وصفته num2 هي الحدّ الأسفل في الراصّة stk1 .
الحدّ الذي فوق الحدّ الأسفل في الراصّة المُعادة يحوي توجيهًا إلى ظهور TwoItems الذي صفته num1 هي الحدّ الذي تحت الحدّ الذي في رأس الراصّة stk1 ، وصفته num2 هي الحدّ الذي فوق الحدّ الأسفل في الراصّة stk1، وهكذا، بحيث الحدّ الذي في رأس الراصّة المُعادة يحوي توجيهًا إلى ظهور TwoItems الذي صفتاه num1 وَ num2 هما حدّان متجاوران موجودان في وسط الراصّة stk1 .

مثال، بالنسبة للراصّة stk1 التي أمامك:

بعد العمليّة تُعاد الراصّة:



يجب عليك الاستعانة بالعمليّة التي كتبتَها في البند "أ".
ملاحظة: لا حاجة لحفظ المحتوى الأصليّ للراصّة stk1 .

/ يتبع في صفحة 8 /

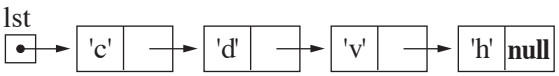
5. انتبه: لهذا السؤال صيغة بلغة Java في صفحة 8، وصيغة أخرى بلغة #C في صفحة 9.

للّذين يحلّون بلغة Java

معطاة العمليّة sod1 :

```
public static Node <Character> sod1(Node <Character> lst, char ch)
{
    if (lst == null)
        return null;
    if (lst.getValue() == ch)
        return lst;
    return sod1(lst.getNext(), ch);
}
```

أ. (1) تتبّع العمليّة، واكتب ماذا يُعاد بالنسبة لِـ ch = 'v' والتوجيه lst إلى سلسلة حلقات الرموز:



(2) ما هو هدف العمليّة sod1 ؟

(3) ما هي تعقيدات زمن تشغيل العمليّة sod1 ؟ علّل.

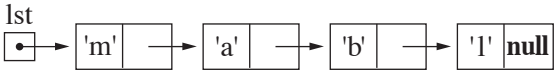
ب. معطاة العمليّة sod2 :

```
public static boolean sod2(Node <Character> lst)
{
    if (sod1(lst,'a') != null && sod1(lst,'b') != null)
        return true;
    return false;
}
```

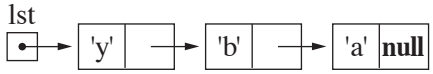
ما هو هدف العمليّة sod2 ؟

جـ. اكتب عمليّة بوليانيّة تتلقّى توجيهًا إلى سلسلة حلقات رموز، وتُعيد true إذا ظهرت فيها حلقتان متجاورتان قيمتاهما 'a' 'b' أو 'b' 'a'. خلاف ذلك — تُعيد العمليّة false .

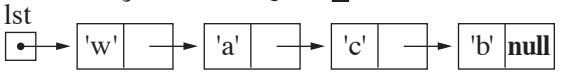
مثال لسلسلة حلقات يظهر فيها 'b' 'a' بالتسلسل:



مثال لسلسلة حلقات يظهر فيها 'a' 'b' بالتسلسل:



مثال لسلسلة حلقات لا يظهر فيها 'b' 'a' وَ 'a' 'b' بالتسلسل:



يجب عليك استعمال العمليّة sod1 .

افترِضْ أنّ جميع الرموز تختلف عن بعضها البعض.

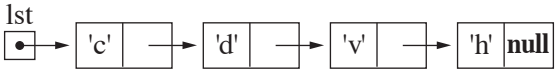
/ يتبع في صفحة 9 /

للّذين يحلّون بلغة #C

معطاة العمليّة Sod1 :

```
public static Node <char> Sod1(Node <char> lst, char ch)
{
    if (lst == null)
        return null;
    if (lst.GetValue() == ch)
        return lst;
    return Sod1(lst.GetNext(), ch);
 }
```

أ. **(1)** تتبّع العمليّة، واكتب ماذا يُعاد بالنسبة لِـ ch = 'v' والتوجيه lst إلى سلسلة حلقات الرموز :



(2) ما هو هدف العمليّة Sod1 ؟

(3) ما هي تعقيدات زمن تشغيل العمليّة Sod1 ؟ علّل.

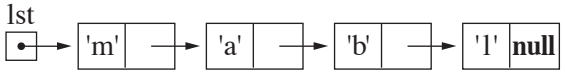
ب. معطاة العمليّة Sod2 :

```
public static bool Sod2(Node <char> lst)
{
    if (Sod1(lst,'a') != null && Sod1(lst,'b') != null)
        return true;
     return false;
}
```

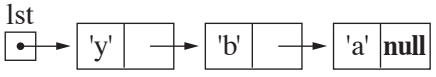
ما هو هدف العمليّة Sod2؟

جـ. اكتب عمليّة بوليانيّة تتلقّى توجيهًا إلى سلسلة حلقات رموز، وتُعيد true إذا ظهرت فيها حلقتان متجاورتان قيمتاهما 'a' 'b' أو 'b' 'a'. خلاف ذلك – تُعيد العمليّة false .

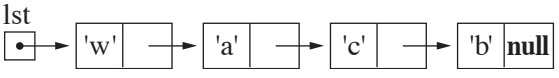
مثال لسلسلة حلقات يظهر فيها 'a' 'b' بالتسلسل :



مثال لسلسلة حلقات يظهر فيها 'b' 'a' بالتسلسل :



مثال لسلسلة حلقات لا يظهر فيها 'a' 'b' وَ 'b' 'a' بالتسلسل :



يجب عليك استعمال العمليّة Sod1 .

افترِضْ أنّ جميع الرموز تختلف عن بعضها البعض.

/ يتبع في صفحة 10 /

6. معطاة العمليّة:

بلغة Java:

```
public static boolean lessThanTree (BinNode <Integer> t, int x)
```

بلغة C#:

```
public static bool LessThanTree (BinNode <int> t, int x)
```

تُعيد العمليّة true إذا كان x أصغر من جميع القيَم في الشجرة t . خلاف ذلك – تُعيد العمليّة false .
تعقيدات زمن تشغيل العمليّة هي O(n). n يمثّل عدد العُقد في الشجرة t .

أ. اكتب عمليّة خارجيّة treeLessThanTree بلغة Java أو TreeLessThanTree بلغة C#، تتلقّى شجرتين بيناريّتين t1 وَ t2 لقيَم صحيحة. معطى أنّه في t2 موجودة عقدة واحدة على الأقلّ. تُعيد العمليّة true إذا كانت كلّ قيمة في الشجرة t1 أصغر من كلّ واحدة من القيَم في الشجرة t2 ، خلاف ذلك – تُعيد العمليّة false .
إذا كانت t1 هي null – تُعيد العمليّة true .
يمكن استعمال العمليّة المعطاة بدون تطبيقها. إذا استعملتَ عمليّات أخرى، عليك تطبيقها.

ب. ما هي تعقيدات زمن تشغيل العمليّة التي كتبتَها في البند "أ"؟ علّل.

/ يتبع في صفحة 11 /

الفصل الثالث (25 درجة)

في هذا الفصل أسئلة في أربعة مسارات:
أنظمة حاسوب وأسمبلي، ص 11-13.
مقدّمة لبحث الأداءات، ص 14-17.
موديلات حسابيّة، ص 18-19.
برمجة موجّهة كائنات بلغة Java، ص 20-23؛ برمجة موجّهة كائنات بلغة C# ، ص 24-27.

أجب عن سؤال واحد في المسار الذي تعلّمتَه.

أنظمة حاسوب وأسمبلي

إذا تعلّمتَ هذا المسار، أجب عن سؤال واحد من السؤالين 7-8 (25 درجة).

7. في هذا السؤال بندان "أ–ب". لا علاقة بينهما. أجب عن البندين.

أ. أمامك قطعة برنامج بلغة أسمبلي.
افترِضْ أنّ الأعداد غير موجَّهة (Unsigned).

```
        CMP  AH , AL
        JAE  AA
        XCHG AH, AL
AA:     CMP AL, AH
        JAE CC
        INC AL
        DEC AH
        JMP AA
CC:     MOV SI, 100H
        MOV [SI], AL
```

(1) محتوى الخازن AX هو 0109H. تتبّع بواسطة جدول متابعة تنفيذ قطعة البرنامج، واكتب ماذا ينتج في الخليّة التي عنوانها 100H بعد التنفيذ.

(2) ماذا تنفّذ قطعة البرنامج بالنسبة لأعداد فرديّة في الخازنين AH, AL؟ تطرّق في إجابتك أيضًا إلى محتوى الخازنين AH, AL قبل تنفيذ قطعة البرنامج.

/ يتبع في صفحة 12 /

ب. (لا علاقة له مع البند "أ" .)

أمامك ستّة أقوال، حدِّد بالنسبة لكلّ واحد منها إذا كان صحيحًا أم غير صحيح، إذا كان القول غير صحيح، فسّر لماذا.

(1) القطعتان 1 وَ 2 اللتان أمامك تنفّذان نفس الشيء.

القطعة 1	القطعة 2
SHL AX, 1	SHR AX, 1
SHR AX, 1	SHL AX, 1

(2) القطعتان 1 وَ 2 اللتان أمامك تنفّذان نفس الشيء.

القطعة 1	القطعة 2
AND AX, 0FEh	SHR AX, 1
	SHL AX, 1

(3) القطعتان 1 وَ 2 اللتان أمامك تنفّذان نفس الشيء.

القطعة 1	القطعة 2
PUSH AX	XCHG AX, BX
PUSH BX	
POP AX	
POP BX	

(4) القطعتان 1 وَ 2 اللتان أمامك تنفّذان نفس الشيء.

القطعة 1	القطعة 2
SUB AX, 0	CMP AX, 0
JNE A1	JNZ A1

(5) في مقطع المعطيات عُرِّفت المصفوفة:

```
ARR   DW 50 dup (?)
```

افترِضْ أنّه في الخازن BX مخزون عنوان حدّ معيّن في المصفوفة ARR .
قطعة البرنامج هذه تخزن في الخازن BX مؤشّر نفس الحدّ.

```
LEA SI, ARR
SUB BX,SI
INC BX
```

(6) قيمة العدد 11101011 المخزون حسب الطريقة: مكمِّل لِـ 2 في 8 بِتّات هو (21—) في الأساس 10.

/ يتبع في صفحة 13 /

8. في مقطع المعطيات الذي أمامك عُرِّفت المعطيات:

ARR DB 100 DUP (?)

REZ DB ?

افترِضْ أنّ الأعداد موجَّهة (Signed).

افترِضْ أنّ جميع الأعداد في المصفوفة تختلف عن بعضها البعض.

تُسمّى المصفوفة "موجيّة" إذا كانت تحقّق الشروط التالية:
الحدّ الأوّل أصغر من الحدّ الثاني، والحدّ الثاني أكبر من الحدّ الثالث، والحدّ الثالث أصغر من الحدّ الرابع، والحدّ الرابع أكبر من الحدّ الخامس، وهكذا.
مثال لمصفوفة تُسمَّى "موجيّة":

0	1	2	3	4
−2	8	6	17	−7

شرح للمثال: الحدّ الأوّل (−2) أصغر من الحدّ الثاني (8)، والحدّ الثاني (8) أكبر من الحدّ الثالث (6)، والحدّ الثالث (6) أصغر من الحدّ الرابع (17) ، والحدّ الرابع (17) أكبر من الحدّ الخامس (−7).

اكتب قطعة برنامج تفحص إذا كانت المصفوفة ARR هي مصفوفة "موجيّة". إذا كانت كذلك، عليك التعويض في المتغيّر REZ القيمة 1، خلاف ذلك – عليك التعويض في المتغيّر REZ القيمة 0.

/ يتبع في صفحة 14 /

مقدّمة لبحث الأداءات

إذا تعلّمتَ هذا المسار، أجب عن سؤال واحد من السؤالين 9-10 (25 درجة) .

9. معطاة مسألة تخطيط خطّيّ :

$$\max\{z = 4x_1 + 6x_2\}$$

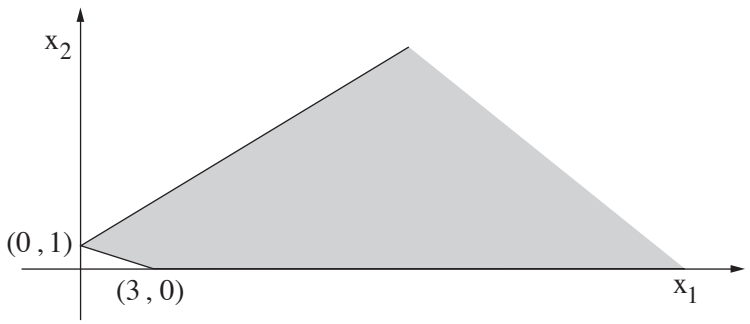
تخضع للاضطرارات التالية :

(1) $-5x_1 + 4x_2 \leq 4$

(2) $x_1 + 3x_2 \geq 3$

(3) $x_2 \geq 0$

أمامك رسم لمجال الحلول الممكنة للمسألة المعطاة .



كلّ واحد من البنود "أ–و" التي في الصفحة التالية يتطرّق إلى مسألة التخطيط الخطّيّ المعطاة .
البنود "أ–و" لا تتعلّق ببعضها البعض . أجب عن جميع البنود .
معطاة أربعة أقوال i-iv . لكلّ واحد من البنود "أ–و" التي في الصفحة التالية يوجد قول واحد صحيح فقط .

i يوجد فقط حلّ أمثل وحيد .

ii توجد لانهاية من الحلول المثلى .

iii الحلّ الأمثل ليس محصورًا .

iv لا يوجد حلّ أمثل .

/ يتبع في صفحة 15 /

بالنسبة لكلّ واحد من البنود "أ–و"، حدِّد أيّ قول من الأقوال i-iv هو الصحيح. اذكر البند، وانسخ القول الصحيح إلى دفترك، وعلّل تحديدك.

- إذا اخترتَ القول i في أحد البنود، عليك إيجاد الحلّ الأمثل الوحيد وقيمة دالّة الهدف في هذا الحلّ.
- إذا اخترتَ القول ii في أحد البنود، عليك كتابة الحلّ الأمثل العامّ للمسألة، وقيمة دالّة الهدف في مجال الحلول المثلى.

أ. أيّ قول هو الصحيح بالنسبة لمسألة التخطيط الخطّيّ المعطاة في بداية السؤال؟ علّل إجابتك.

ب. نغيّر فقط دالّة هدف المسألة المعطاة في بداية السؤال إلى $\min\{z = 4x_1 + 6x_2\}$.
أيّ قول هو الصحيح بعد التغيير؟ علّل إجابتك.

جـ. نغيّر فقط دالّة هدف المسألة المعطاة في بداية السؤال إلى $\max\{z = 2x_1 + 6x_2\}$.
أيّ قول هو الصحيح بعد التغيير؟ علّل إجابتك.

د. نغيّر فقط دالّة هدف المسألة المعطاة في بداية السؤال إلى $\min\{z = 2x_1 + 6x_2\}$.
أيّ قول هو الصحيح بعد التغيير؟ علّل إجابتك.

هـ. نُضيف اضطرارًا إلى المسألة المعطاة في بداية السؤال، وهو: $x_1 + x_2 \leq 1$.
أيّ قول هو الصحيح بعد إضافة الاضطرار؟ علّل إجابتك.

و. نُضيف اضطرارًا إلى المسألة المعطاة في بداية السؤال، وهو: $x_2 \leq -x_1$.
أيّ قول هو الصحيح بعد إضافة الاضطرار؟ علّل إجابتك.

/ يتبع في صفحة 16 /

10. في هذا السؤال بندان "أ–ب". لا علاقة بينهما. أجب عن البندين.

أ. $G = (V , E)$ هو رسم بيانيّ **موجَّه** ممثَّل بواسطة مصفوفة المجاوَرات التي أمامك.

	a	**b**	**c**	**d**	**e**
a	0	0	1	1	0
b	1	0	0	1	0
c	0	0	0	0	1
d	0	1	0	0	0
e	1	0	0	0	0

(1) ارسم الرسم البيانيّ G الممثَّل بواسطة مصفوفة المجاوَرات.

(2) جد مركِّبات الربط القويّ (Strong Connected Components) التي في الرسم البيانيّ المعطى. بالنسبة لكلّ مركِّب ربط قويّ وجدتَه، اكتب مجموعة رؤوسه.

(3) حدِّد ما هو العدد الأقصى للأقواس التي يمكن إزالتها من الرسم البيانيّ المعطى، ومع ذلك يبقى الرسم البيانيّ يحوي نفس عدد مركِّبات الربط القويّ الذي وجدتَه في البند الفرعيّ "أ (2)". ما هو القوس أو ما هي الأقواس؟

ب. (لا علاقة له مع البند "أ".)

(1) في الجدول الذي أمامك معطاة مسألة نقل وجزء من حلّ أساسيّ ممكن: $x_{12} = 1$, $x_{11} = 9$.

المصادر	**الغايات**			**العرض**
	1	2	3	
1	1 (9)	5 (1)	7	10
2	1	8	4	11
3	5	2	8	10
الطلب	9	12	10	

انسخ الجدول إلى دفترك، وأكمل القيَم فيه حسب طريقة الزاوية الشماليّة – الغربيّة.

/ يتبع في صفحة 17 /

(2) في الجدول الذي أمامك معطى جزء من حلّ أساسيّ ممكن لمسألة نقل، ومعطاة قيَم $u_1 ، u_2 ، u_3 ، v_1 ، v_2 ، v_3$.

المصادر	الغايات			العرض	u_i
	1	2	3		
1	3 (20)	5	7	20	1
2	2	8 (10)	14	10	0
3	2	6	8 (10)	15	-2
الطلب	20	15	10		
v_j	2	8	10		

انسخ الجدول إلى دفترك، وأكمله مع الأخذ بالحسبان قيَم جميع الـ u_i والـ v_i حتّى ينتج حلّ أساسيّ ممكن.

(3) في الجدول الذي أمامك معطى حلّ أساسيّ ممكن لمسألة نقل، ومعطاة قيَم $u_1 ، u_2 ، u_3 ، v_1 ، v_2 ، v_3$.

المصادر	الغايات			العرض	u_i
	1	2	3		
1	34	15 (15)	17 (3)	18	0
2	10 (10)	8 (0)	4	10	-7
3	25	18	18 (10)	10	1
الطلب	10	15	13		
v_j	17	15	17		

هل الحلّ هو أمثل؟ علّل إجابتك.

/ يتبع في صفحة 18 /

موديلات حسابيّة

إذا تعلّمتَ هذا المسار، أجب عن سؤال واحد من السؤالين 11-12 (25 درجة) .

11. في هذا السؤال بندان "أ–ب" . لا علاقة بينهما . أجب عن البندين .

أ. أمامك تعريف : **بادئة** الكلمة x هي كلّ كلمة تنتج بواسطة حذف عدد معيّن من الرموز من آخر الكلمة x ، بما في ذلك الكلمة الفارغة والكلمة x نفسها .

مثال : بالنسبة للكلمة $x = abcbad$ جميع بادئات الكلمة x هي :

$\varepsilon, a, ab, abc, abcb, abcba, abcbad$

أمامك اللغة L فوق الأبجديّة $\Sigma = \{a,b,c,d\}$.

L هي مجموعة الكلمات التي في كلّ واحدة منها بالنسبة **لكلّ بادئة** في الكلمة – الفرق بين عدد مرّات ظهور الرمز c وبين عدد مرّات ظهور الرمز d هو أكبر من 0 أو يساوي 0 ، وأصغر من 3 أو يساوي 3 :

$$0 \leq \#_c(w) - \#_d(w) \leq 3$$

$\#_c(w)$ يشير إلى عدد مرّات ظهور c في الكلمة w .

$\#_d(w)$ يشير إلى عدد مرّات ظهور d في الكلمة w .

أمثلة لكلمات تتبع للّغة L :

accbdcacab , bacaabdbcb , abba , cdcdcd , abcbadb

أمثلة لكلمات لا تتبع للّغة L :

daac – لأنّه توجد البادئة d ، التي فيها : $\#_c(w) - \#_d(w) = -1 < 0$

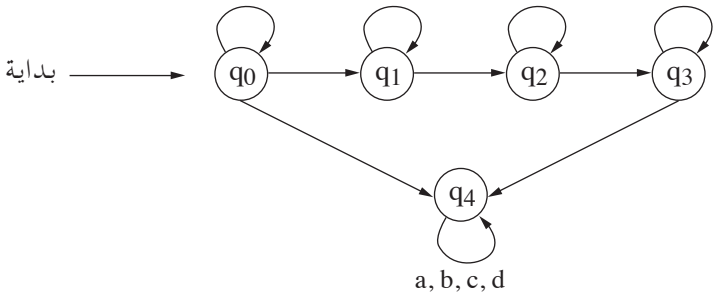
cddc – لأنّه توجد البادئة cdd ، التي فيها : $\#_c(w) - \#_d(w) = -1 < 0$

accbdcacacd – لأنّه توجد البادئة accbdcacac ، التي فيها : $\#_c(w) - \#_d(w) = 4 > 3$

(انتبه : تكملة السؤال في الصفحة التالية .)

/ يتبع في صفحة 19 /

أمامك رسم جزئيّ لأوتومات نهائيّ **محدود** يتلقّى اللغة L . هناك انتقالات وإشارات إدخال وحالات متلقّية ناقصة في الرسم. يشمل الرسم جميع حالات الأوتومات.



انسخ الرسم إلى دفترك، وأكمله بحيث يكون الأوتومات **محدودًا** ويتلقّى اللغة L .
عليك إكمال الانتقالات الناقصة وإشارات الإدخال الناقصة، والإشارة إلى **جميع الحالات المتلقّية**.
انتبه: لا تُضِف حالات إلى الأوتومات، ولا تُنقص منه حالات وانتقالات.

ب. (لا علاقة له مع البند "أ".)

Σ^* هي مجموعة كلّ الكلمات فوق الأبجديّة Σ ، بما في ذلك الكلمة الفارغة.

معطاة اللغتان L_2 وَ L_1 فوق الأبجديّة Σ .

$L_1 = \Sigma^*$ وَ L_2 هي لغة غير نظاميّة.

نُعرِّف: $L_3 = L_2 \cap \overline{L_1}$.

(1) ما هي اللغة $\overline{L_1}$ ؟

(2) هل اللغة L_3 هي نظاميّة؟ علّل إجابتك.

12. في هذا السؤال يندان، "أ–ب". لا علاقة بينهما. أجب عن اليندين.

أ. معطاة الأبجديّة $\Sigma = \{a,b\}$. ابنِ أوتومات راصّة بالنسبة للّغة:

$$L = \{a^2 b^k a^n \mid n > 0 \quad , \quad k > 0 \quad , \quad n < k\}$$

ب. (لا علاقة له مع البند "أ".)

ابنِ آلة تيورينج تتلقّى كمُدخَل عددًا x بتمثيل أونواريّ، وتحسب قيمة الدالّة التي أمامك.

$$f(x) = \begin{cases} x+1 & x < 2 \\ x-1 & x \geq 2 \end{cases}$$

/ يتبع في صفحة 20 /

برمجة موجّهة كائنات

إذا تعلّمتَ هذا المسار وتكتب بلغة Java، أجب عن سؤال واحد من السؤالين 13-14 (25 درجة).

13. أمامك مشروع عُرِّفت فيه الفئات التالية: A ، B ، C ، D ، E . طُبِّقت في المشروع عمليّة f() في فئتين.
قطعة الكود التي أمامك **سليمة**.

```
A a1 = new A();
A e1 = new E();
E c1 = new C();
C b1 = new B();
C d1 = new D();
```

معطى أنّ:
الأمر B d2=new D(); يؤدّي إلى خطأ تجميع (مسح أوامر).
الأمر a1.f(); يؤدّي إلى خطأ تجميع (مسح أوامر).
الأمر ((E)e1).f(); سليم ويطبع "bye-bye" .
الأمر ((B)b1).f(); سليم ويطبع "hello" .
الأمر ((D)b1).f(); يؤدّي إلى خطأ في زمن التشغيل.

أ. ارسم شجرة وراثة لجميع الفئات، واذكر في أيّة فئتين طُبِّقت العمليّة f() .

ب. معطاة الفئة Z :

```
public class Z
{
    public void g(){}
}
```

(1) أَضِف الفئة Z إلى شجرة الوراثة التي رسمتَها في البند "أ"، بحيث يكون الأمر الذي أمامك سليمًا:

```
Z x = new A();
```

(2) قطعة الكود التي أمامك **سليمة**.

```
A a2 = new A();
Z z1 = new Z();
Z a3 = new A();
```

بالنسبة لكلّ واحد من الأوامر v-i التي أمامك، اذكر إذا كان سليمًا أم غير سليم.
إذا كان الأمر غير سليم، علّل لماذا.
البنود v-i **لا** تتعلّق ببعضها البعض.

```
i     a2 = z1;
ii    a3 = z1;
iii   ((A)z1).g();
iv    a2.g();
v     ((A)a3).g();
```

/ يتبع في صفحة 21 /

جـ. معطاة الفئة Y :

```
public class Y
{
    public void m(){}
}
```

(1) أَضِف الفئة Y إلى شجرة الوراثة التي رسمتَها في البند "أ"، بحيث يكون الأمر الذي أمامك سليمًا (بدون التطرّق إلى المعطيات التي في البند "ب").

```
C f = new Y();
```

(2) قطعة الكود التي أمامك **سليمة**.

```
A a2 = new A();
Y y1 = new Y();
C y2 = new Y();
```

بالنسبة لكلّ واحد من الأوامر v-i التي أمامك، اذكر إذا كان سليمًا أم غير سليم.
إذا كان الأمر غير سليم، علّل لماذا.
البنود v-i **لا** تتعلّق ببعضها البعض.

i `a2 = y1;`

ii `y2 = y1;`

iii `((A)y1).m();`

iv `y2.m();`

v `((A)y2).m();`

/ يتبع في صفحة 22 /

14. دفع ضرائب الأرنونا (Property tax) هو دفع للسلطات المحلّيّة مقابل مساحة السكن والأرض الإضافيّة التي يملكها السكّان. سعر الأرنونا هو 10 شواكل للمتر المربّع للمساحة المعدّة للسكن وَ 0.5 شيكل للمتر المربّع للأرض الإضافيّة. مثال، الساكن الذي يملك مساحة للسكن بِكبَر 100 متر مربّع ومساحة أرض بِكبَر 500 متر مربّع يدفع ضرائب أرنونا بمبلغ $100 \cdot 10 + 500 \cdot 0.5 = 1250$.

قُسِّم السكّان إلى ثلاث مجموعات:

- الساكن في المدينة يدفع ضرائب أرنونا مقابل مساحة السكن والأرض التي يملكها، ويحصل على منحة بمبلغ 250 شيكلاً لدفع الأرنونا.
- الساكن المسنّ في المدينة – ساكن في المدينة عمره 60 سنة أو أكثر — يدفع ضرائب أرنونا كما يدفع الساكن في المدينة، وبالإضافة إلى ذلك يستحقّ تخفيضًا بحسب عمره: 1% تخفيض مقابل كلّ سنة من عمره بعد عمر 60 سنة. يُحسَب التخفيض بعد إنقاص المنحة من المبلغ الأصليّ.
- الساكن في القرية يدفع أرنونا مقابل مساحة السكن والأرض التي يملكها ويستحقّ تخفيضًا نسبته 10%. **الساكن في القرية الذي عمره 60 سنة أو أكثر لا يستحقّ تخفيضًا يتعلّق بالعمر.**

إذا كان مبلغ دفع ضرائب الأرنونا بعد إنقاص التخفيض والمنحة هو عدد سالب، يكون الساكن معفيًّا من الدفع (يدفع 0 شيكل).

تقرّر تطوير برنامج حاسوب لحساب ضرائب الأرنونا وجبايتها من السكّان، ولهذا الغرض عُرِّفت واجهة التطبيق التي أمامك.

```
interface IData
{
    public String getName(); // تعيد اسم الساكن
    public double getPropertyTax(); // تعيد مبلغ ضريبة الأرنونا
}
```

في البندين "أ–ب" اللذين أمامك عليك تعريف الفئات حسب مبادئ البرمجة الموجّهة كائنات وفقًا للمتطلّبات:

- معطيات حساب ضريبة الأرنونا (الدفع مقابل مساحة الشقّة بالمتر المربّع، والدفع مقابل مساحة الأرض بالمتر المربّع، ونسبة التخفيض بالنسب المئويّة، ومبلغ المنحة بالشواكل) عرِّفها مرّة واحدة فقط كمعطيات ثابتة.
- جميع الصفات التي ليست ثابتة عرِّفها كخاصّة (private).

أ. عرِّف فئة Resident (تمثّل ساكنًا)، تطبّق واجهة التطبيق IData، واكتب فيها صفات الفئة (لا حاجة لكتابة عمليّة بنائيّة).

ب. عرِّف ثلاث فئات لكلّ واحدة من مجموعات السكّان: CityResident — ساكن في المدينة، SeniorCityResident — ساكن مسنّ في المدينة، VillageResident — ساكن في القرية.
اكتب صفات الفئات والعمليّة ()getPropertyTax التي تمكّن حساب مبلغ دفع ضريبة الأرنونا لكلّ واحدة من ثلاث مجموعات السكّان (لا حاجة لكتابة عمليّة بنائيّة).

/ يتبع في صفحة 23 /

جـ. معطى جسم عمليّة يتلقّى مصفوفة a لمعطيات السكّان، ويطبع بالنسبة لكلّ ساكن مسنّ في المدينة، اسمَهُ ومبلغ الأرنونا الذي عليه دفعه.

```
for (int i = 0;  i < a.length; i++)
{
    if (a[i] instanceof SeniorCityResident)
    {
        System.out.println(a[i].getName()+" "+a[i].getPropertyTax());
    }
}
```

بالنسبة لكلّ واحد من عناوين العمليّة (1)-(3) التي أمامك، اذكر إذا كانت العمليّة سليمة أم غير سليمة.
إذا كانت العمليّة غير سليمة – علّل لماذا، وصحِّح العمليّة، وانسخ العمليّة المصحَّحة إلى دفترك.

(1) public static void print(Object [] a)

(2) public static void print(IData [] a)

(3) public static void print(Resident [] a)

/ يتبع في صفحة 24 /

برمجة موجّهة كائنات

إذا تعلّمتَ هذا المسار وتكتب بلغة C#، أجب عن سؤال واحد من السؤالين 15-16 (25 درجة) .

15. أمامك مشروع عُرِّفت فيه الفئات التالية: A ، B ، C ، D ، E . طُبِّقت في المشروع عمليّة F() في فئتين.
قطعة الكود التي أمامك **سليمة**.

```
A a1 = new A();
A e1 = new E();
E c1 = new C();
C b1 = new B();
C d1=  new D();
```

معطى أنّ:
الأمر ;()B d2=new D يؤدّي إلى خطأ تجميع (مسح أوامر) .
الأمر ;()a1.F يؤدّي إلى خطأ تجميع (مسح أوامر) .
الأمر ;()((E)e1).F سليم ويطبع "bye-bye" .
الأمر ;()((B)b1).F سليم ويطبع "hello" .
الأمر ;()((D)b1).F يؤدّي إلى خطأ في زمن التشغيل.

أ. ارسم شجرة وراثة لجميع الفئات، واذكر في أيّة فئتين طُبِّقت العمليّة F() .

ب. معطاة الفئة Z :

```
public class Z
{
        public void G(){}
}
```

(1) أَضِف الفئة Z إلى شجرة الوراثة التي رسمتَها في البند "أ"، بحيث يكون الأمر الذي أمامك سليمًا:

```
Z x = new A();
```

(2) قطعة الكود التي أمامك **سليمة**.

```
A a2 = new A();
Z z1 = new Z();
Z a3 = new A();
```

بالنسبة لكلّ واحد من الأوامر v-i التي أمامك، اذكر إذا كان سليمًا أم غير سليم.
إذا كان الأمر غير سليم، علّل لماذا.
البنود v-i **لا** تتعلّق ببعضها البعض.

```
i     a2 = z1;
ii    a3 = z1;
iii   ((A)z1).G();
iv    a2.G();
v     ((A)a3).G();
```

/ يتبع في صفحة 25 /

جـ. معطاة الفئة Y :

```
public class Y
{
    public void M(){}
}
```

(1) أَضِف الفئة Y إلى شجرة الوراثة التي رسمتَها في البند "أ"، بحيث يكون الأمر الذي أمامك سليمًا (بدون التطرّق إلى المعطيات التي في البند "ب").

```
C f = new Y();
```

(2) قطعة الكود التي أمامك **سليمة**.

```
A a2 = new A();
Y y1 = new Y();
C y2 = new Y();
```

بالنسبة لكلّ واحد من الأوامر v-i التي أمامك، اذكر إذا كان سليمًا أم غير سليم.
إذا كان الأمر غير سليم، علّل لماذا.
البنود v-i **لا** تتعلّق ببعضها البعض.

```
i     a2 = y1;
ii    y2 = y1;
iii   ((A)y1).M();
iv    y2.M();
v     ((A)y2).M();
```

/ يتبع في صفحة 26 /

16. دفع ضرائب الأرنونا (Property tax) هو دفع للسلطات المحلّيّة مقابل مساحة السكن والأرض الإضافيّة التي يملكها السكّان. سعر الأرنونا هو 10 شواكل للمتر المربّع للمساحة المعدّة للسكن وَ 0.5 شيكل للمتر المربّع للأرض الإضافيّة. مثال، الساكن الذي يملك مساحة للسكن بِكبَر 100 متر مربّع ومساحة أرض بِكبَر 500 متر مربّع يدفع ضرائب أرنونا بمبلغ $100 \cdot 10 + 500 \cdot 0.5 = 1250$.

قُسِّم السكّان إلى ثلاث مجموعات:

- الساكن في المدينة يدفع ضرائب أرنونا مقابل مساحة السكن والأرض التي يملكها، ويحصل على منحة بمبلغ 250 شيكلاً لدفع الأرنونا.
- الساكن المسنّ في المدينة – ساكن في المدينة عمره 60 سنة أو أكثر — يدفع ضرائب أرنونا كما يدفع الساكن في المدينة، وبالإضافة إلى ذلك يستحقّ تخفيضًا بحسب عمره: 1% تخفيض مقابل كلّ سنة من عمره بعد عمر 60 سنة. يُحسَب التخفيض بعد إنقاص المنحة من المبلغ الأصليّ.
- الساكن في القرية يدفع أرنونا مقابل مساحة السكن والأرض التي يملكها ويستحقّ تخفيضًا نسبته 10%. **الساكن في القرية الذي عمره 60 سنة أو أكثر لا يستحقّ تخفيضًا يتعلّق بالعمر.**

إذا كان مبلغ دفع ضرائب الأرنونا بعد إنقاص التخفيض والمنحة هو عدد سالب، يكون الساكن معفيًّا من الدفع (يدفع 0 شيكل).

تقرّر تطوير برنامج حاسوب لحساب ضرائب الأرنونا وجبايتها من السكّان، ولهذا الغرض عُرِّفت واجهة التطبيق التي أمامك.

```
interface IData
{
    string GetName(); // تعيد اسم الساكن
    double GetPropertyTax(); // تعيد مبلغ ضريبة الأرنونا
}
```

في البندين "أ–ب" اللذين أمامك عليك تعريف الفئات حسب مبادئ البرمجة الموجّهة كائنات وفقًا للمتطلّبات:

- معطيات حساب ضريبة الأرنونا (الدفع مقابل مساحة الشقّة بالمتر المربّع، والدفع مقابل مساحة الأرض بالمتر المربّع، ونسبة التخفيض بالنسب المئويّة، ومبلغ المنحة بالشواكل) عرِّفها مرّة واحدة فقط كمعطيات ثابتة.
- جميع الصفات التي ليست ثابتة عرِّفها كخاصّة (private).

أ. عرِّف فئة Resident (تمثّل ساكنًا)، تطبّق واجهة التطبيق IData، واكتب فيها صفات الفئة (لا حاجة لكتابة عمليّة بنائيّة).

ب. عرِّف ثلاث فئات لكلّ واحدة من مجموعات السكّان: CityResident — ساكن في المدينة، SeniorCityResident — ساكن مسنّ في المدينة، VillageResident — ساكن في القرية.
اكتب صفات الفئات والعمليّة ()GetPropertyTax التي تمكّن حساب مبلغ دفع ضريبة الأرنونا لكلّ واحدة من ثلاث مجموعات السكّان (لا حاجة لكتابة عمليّة بنائيّة).

/ يتبع في صفحة 27 /

جـ. معطى جسم عمليّة يتلقّى مصفوفة a لمعطيات السكّان، ويطبع بالنسبة لكلّ ساكن مسنّ في المدينة، اسمَهُ ومبلغ الأرنونا الذي عليه دفعه.

```
for ( int i = 0; i < a.Length; i++)
{
    if (a[i] is SeniorCityResident)
    {
        Console.WriteLine (a[i].GetName()+" "+a[i].GetPropertyTax());
    }
}
```

بالنسبة لكلّ واحد من عناوين العمليّة (1)-(3) التي أمامك، اذكر إذا كانت العمليّة سليمة أم غير سليمة.
إذا كانت العمليّة غير سليمة – علّل لماذا، وصحِّح العمليّة، وانسخ العمليّة المصحَّحة إلى دفترك.

(1) public static void Print(Object [] a)

(2) public static void Print(IData [] a)

(3) public static void Print(Resident [] a)

בהצלחה!
نتمنّى لك النّجاح!

זכות היוצרים שמורה למדינת ישראל.
אין להעתיק או לפרסם אלא ברשות משרד החינוך.
حقوق الطّبع محفوظة لدولة إسرائيل.
النّسخ أو النّشر ممنوعان إلّا بإذن من وزارة التّربية والتّعليم.