מדינת ישראל
משרד החינוך

סוג הבחינה: בגרות לבתי"ס על־יסודיים
מועד הבחינה: קיץ תשע"ו, **מועד ב**
מספר השאלון: **899381**
תרגום לערבית (2)

מדעי המחשב

על פי תכנית הרפורמה ללמידה משמעותית

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

دولة إسرائيل
وزارة التّربية والتّعليم

نوع الامتحان: بجروت للمدارس الثّانويّة
موعد الامتحان: صيف 2016، **الموعد "ب"**
رقم النّموذج: **899381**
ترجمة إلى العربيّة (2)

علم الحاسوب

حسب خطّة الإصلاح: التّعلّم ذي المعنى

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 عمليّة رئيسيّة في الفئة **Stam**. تمّ في العمليّة الرئيسيّة تعريف مصفوفة أحاديّة الأبعاد arr من نمط صحيح.
اكتب بلغة Java أو بلغة C# ، في العمليّة الرئيسيّة، قطعة برنامج تستقبل عددًا صحيحًا num ، وتطبع مجموع كلّ الحدود في المصفوفة، التي قيمتها أصغر من num.

/ يتبع في صفحة 3 /

أجب عن أحد السؤالين 2-3 (15 درجة) .

2. عَرِّفوا الفئة **A** التي لها صفتان : n1 — من نمط صحيح، وَ n2 — من نمط صحيح .
تمّ في الفئة تعريف : عمليّتين بنائيّتين،
وعمليّات get وَ set بلغة Java وَ Get وَ Set بلغة C# لكلّ صفة .
في هذه الفئة، العمليّة toString بلغة Java وَ ToString بلغة C# تُعيد نصًّا هو عبارة عن تمرين جمع قيمتَي n1 وَ n2، ونتيجته .
مثال :
إذا كانت القيمة التي في n1 هي العدد 3 والقيمة التي في n2 هي العدد 9 ، تُعيد العمليّة النصّ : "3 + 9 = 12" .
كما أنّ العمليّة void add(int k) بلغة Java وَ void Add(int k) بلغة C# تُضيف القيمة k إلى قيمة كلّ واحدة من الصفتين n1 وَ n2 .
طباعة كائن تستدعي بلغة Java العمليّة toString التابعة للفئة، وبلغة C# العمليّة ToString التابعة للفئة .
أمامك العمليّة الرئيسيّة في الفئة Program مكتوبة بلغة Java وبلغة C# .
تَستعمل العمليّة الفئة **A** .

C#

```
public class Program
{
   public static void Main()
   {
      A a = new A();
      a.SetN1(4);
      a.SetN2(5);
      A a1 = new A(a.GetN1(), a.GetN2());
      Console.WriteLine(a);
      a1.Add(4);
      Console.WriteLine(a);
      Console.WriteLine(a1);
   }
}
```

Java

```
public class Program
{
   public static void main(String[] args)
   {
      A a = new A();
      a.setN1(4);
      a.setN2(5);
      A a1 = new A(a.getN1(), a.getN2());
      System.out.println (a);
      a1.add(4);
      System.out.println(a);
      System.out.println(a1);
   }
}
```

(انتبه: تكملة السؤال في الصفحة التالية.) /يتبع في صفحة 4/

أ. عليك تعريف الفئة **A** حسب البنود الفرعيّة i-v بحيث تُنفَّذ العمليّة الرئيسيّة بدون أخطاء في التجميع والتحويل للغة الآلة وَ/أو التشغيل.

i اكتب بلغة Java أو بلغة C# عنوان الفئة وصفاتها.

ii اكتب بلغة Java أو بلغة C# عنوانَي العمليّتين البنائيّتين للفئة **A** .

iii طبِّق بلغة Java جميع عمليّات get وَ set ، أو طبِّق بلغة C# جميع عمليّات Get وَ Set، الضروريّة لتشغيل العمليّة الرئيسيّة.

iv طبِّق بلغة Java العمليّة add أو بلغة C# العمليّة Add .

v طبِّق بلغة Java العمليّة toString أو بلغة C# العمليّة ToString .

ب. اعرض الكائنات التي تتكوّن في أعقاب تفعيل العمليّة الرئيسيّة. بالنسبة لكلّ كائن، اكتب أسماء جميع صفاته وقيَم جميع الصفات.

كما واعرض التغيّرات في الكائنات في أعقاب الأمر add(4) بلغة Java أو Add(4) بلغة C#.
اكتب مُخرَج العمليّة الرئيسيّة.

/ يتبع في صفحة 5 /

3. أمامك واجهة تطبيق الفئة **Button** التي تمثّل زِرًّا. للزرّ ثلاث صفات:
رقم مكتوب عليه — num، وكِبَر الزرّ — size، ولون الزرّ — color.

وصف العمليّة	عنوان العمليّة بلغة Java
عمليّة تبني زرًّا مكتوبًا عليه الرقم num ، بِكِبَر 5 ، بلون "black" .	public Button(int num)
عمليّة تبني زرًّا مكتوبًا عليه الرقم num، بِكِبَر size ، بلون color .	public Button(int num , int size , String color)
عمليّة تُعيد كِبَر الزرّ.	public int getSize()
عمليّة تُكبِّر كِبَر الزرّ بـ x .	public void addToSize(int x)
عمليّة تُعيد true إذا كان الزرّ other مطابقًا بِكِبَره للزرّ الحاليّ، خلاف ذلك – تُعيد العمليّة false .	public boolean isSameSize(Button other)

وصف العمليّة	عنوان العمليّة بلغة C#
عمليّة تبني زرًّا مكتوبًا عليه الرقم num ، بِكِبَر 5 ، بلون "black" .	public Button(int num)
عمليّة تبني زرًّا مكتوبًا عليه الرقم num، بِكِبَر size ، بلون color .	public Button(int num , int size , string color)
عمليّة تُعيد كِبَر الزرّ.	public int GetSize()
عمليّة تُكبِّر كِبَر الزرّ بـ x .	public void AddToSize(int x)
عمليّة تُعيد true إذا كان الزرّ other مطابقًا بِكِبَره للزرّ الحاليّ، خلاف ذلك – تُعيد العمليّة false .	public bool IsSameSize(Button other)

أ. طبِّق في الفئة **Button** ، بلغة Java أو بلغة C# ، العمليّة البنائيّة Button(int num) .

ب. طبِّق في الفئة **Button** ، بلغة Java أو بلغة C# ، العمليّة التي تُكبِّر كِبَر الزرّ بـ x .

جـ. طبِّق في الفئة **Button** ، بلغة Java أو بلغة C# ، العمليّة التي تفحص إذا كان كِبَر الزرّ other مطابقًا لكِبَر الزرّ الحاليّ.

(انتبه: البند "د" للسؤال في الصفحة التالية.)

/يتبع في صفحة 6/

د. أمامك قطعة من عمليّة رئيسيّة في الفئة Program مكتوبة بلغة Java وبلغة C# .
تتبّع تنفيذ قطعة العمليّة، واكتب المُخرَج الذي ينتج. لكلّ كائن، اكتب أسماء كلّ صفاته وقيَم جميع الصفات والتغيّرات التي كانت فيها.

Java

```
Button a1 = new Button(1, 12, "red");
Button a2 = new Button(2, 12, "green");
Button a3 = new Button(3, 12, "blue");
Button a4 = new Button(4);
a1.addToSize(2);
a4.addToSize(9);
if (a1.isSameSize(a3))
        System.out.println("***");
if (a4.isSameSize(a1))
        System.out.println("$$$");
if (a2.isSameSize(a3))
        System.out.println("###");
```

C#

```
Button a1 = new Button(1, 12, "red");
Button a2 = new Button(2, 12, "green");
Button a3 = new Button(3, 12, "blue");
Button a4 = new Button(4);
a1.AddToSize(2);
a4.AddToSize(9);
if (a1.IsSameSize(a3))
        Console.WriteLine("***");
if (a4.IsSameSize(a1))
        Console.WriteLine("$$$");
if (a2.IsSameSize(a3))
        Console.WriteLine("###");
```

/ يتبع في صفحة 7 /

الفصل الثاني (50 درجة)

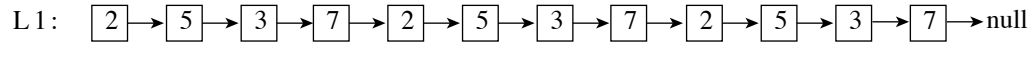
انتبه: في كلّ سؤال يُطلب فيه تطبيق، بإمكانك استعمال عمليّات الفئات دور وراصّة وشجرة بينارية وحلقة رابطة، بدون تطبيقها.
إذا استعملتَ عمليّات إضافيّة، عليك تطبيقها.

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

4. تسمّى القائمة L **مثلّثيّة** إذا كانت تحقّق الشروط التالية:

* القائمة ليست فارغة.
* عدد الحدود فيها يقسم على 3 بدون باقٍ.
* الحدود في الثلث الأوّل من القائمة تحوي نفس القيَم التي تحويها الحدود في الثلث الثاني من القائمة ونفس القيَم التي تحويها الحدود في الثلث الثالث من القائمة. القيَم مرتَّبة بنفس الترتيب في كلّ واحد من الأثلاث.

مثلاً: القائمة L1 التي أمامك هي قائمة **مثلّثيّة** بطول 12 .



اكتب بلغة C# أو بلغة Java عملية خارجيّة تتلقّى قائمة L حدودها من نمط صحيح.
إذا كانت L قائمة **مثلّثيّة**، تعيد العمليّة true.
خلاف ذلك – تعيد العمليّة false.

/ يتبع في صفحة 8 /

5. **انتبه:** لهذا السؤال صيغتان: الأولى بلغة Java (في الصفحتين 8-9)، والثانية بلغة C# (في الصفحتين 10-11). اعمل حسب اللغة التي تعلّمتَها.

للّذين يحلّون بلغة Java

أمامك العمليّتان sod وَ what اللتان تتلقّيان المصفوفة a التي حدودها من نمط صحيح والمرتَّبة بترتيب تصاعديّ، وتتلقّيان أيضًا عددًا صحيحًا k .

للعمليّتين نفس ادّعاء الخروج.

```
public static boolean sod(int[] a , int k)
{
    for (int i = 0; i < a.length−1; i++)
    {
        int j = i+1;
        while (j < a.length)
        {
            if (a[i] + a[j] == k)
                return true;
            j++;
        }
    }
    return false;
}

public static boolean what(int[] a , int k)
{
    int left = 0 , right = a.length−1;
    while (left < right)
    {
        if (a[left] + a[right] == k)
            return true;
        if (a[left] + a[right] < k)
            left++;
        else
            right−−;
    }
    return false;
}
```

/ يتبع في صفحة 9 /

معطاة المصفوفة a :

a	2	4	7	12	18

أ. تتبّع بواسطة جدول متابعة تنفيذ العمليّة sod بالنسبة للمصفوفة المعطاة a ، والعدد k = 11 . اكتب القيمة التي تُعاد.
يجب أن يشمل جدول المتابعة أعمدة بالنسبة لـ : i, j, a[i], a[j] ، وعمودًا إضافيًّا يُذكر فيه إذا كان الشرط الذي في الأمر if يتحقّق أم لا يتحقّق.

ب. تتبّع بواسطة جدول متابعة تنفيذ العمليّة sod بالنسبة للمصفوفة المعطاة a ، والعدد k = 10 . اكتب القيمة التي تُعاد.
يجب أن يشمل جدول المتابعة الأعمدة التي تمّ تفصيلها في البند "أ".

جـ. ما هو ادّعاء خروج العمليّة sod ؟

د. ما هي تعقيدات زمن تشغيل العمليّة sod ؟ علّل إجابتك.

هـ. تتبّع بواسطة جدول متابعة تنفيذ العمليّة what بالنسبة للمصفوفة المعطاة a ، والعدد k = 11 . اكتب القيمة التي تُعاد.
يجب أن يشمل جدول المتابعة أعمدة بالنسبة لـ : left, right, a[left], a[right] ، وعمودًا إضافيًّا لكلّ واحد من الأمرين if ، بحيث يُذكَر في كلّ واحد من العمودين إذا كان الشرط الذي في الأمر if يتحقّق أم لا يتحقّق.

و. ما هي تعقيدات زمن تشغيل العمليّة what ؟ علّل إجابتك.

ز. أيّ من العمليّتين – sod أم what – أكثر نجاعة؟ علّل إجابتك.

حـ. تمّ تغيير ادّعاء دخول العمليّتين sod وَ what بحيث يمكن نقل مصفوفة a غير مرتَّبة إليهما.

(1) هل يتغيّر ادّعاء خروج العمليّة sod ؟ علّل إجابتك.

(2) هل يتغيّر ادّعاء خروج العمليّة what ؟ علّل إجابتك.

/ يتبع في صفحة 10 /

للّذين يحلّون بلغة C#

أمامك العمليّتان Sod وَ What اللتان تتلقّيان المصفوفة a التي حدودها من نمط صحيح والمرتَّبة بترتيب تصاعديّ، وتتلقّيان أيضًا عددًا صحيحًا k .
للعمليّتين نفس ادّعاء الخروج.

```
public static bool Sod(int[] a , int k)
{
    for (int i = 0; i < a.Length−1; i++)
    {
        int j = i+1;
        while (j < a.Length)
        {
            if (a[i] + a[j] == k)
                return true;
            j++;
        }
    }
    return false;
}

public static bool What(int[] a , int k)
{
    int left = 0 , right = a.Length−1;
    while (left < right)
    {
        if (a[left] + a[right] == k)
            return true;
        if (a[left] + a[right] < k)
            left++;
        else
            right−−;
    }
    return false;
}
```

/ يتبع في صفحة 11 /

معطاة المصفوفة a :

a	2	4	7	12	18

أ. تتبّع بواسطة جدول متابعة تنفيذ العمليّة Sod بالنسبة للمصفوفة المعطاة a ، والعدد k = 11 . اكتب القيمة التي تُعاد.
يجب أن يشمل جدول المتابعة أعمدة بالنسبة لـ : i, j, a[i], a[j] ، وعمودًا إضافيًّا يُذكر فيه إذا كان الشرط الذي في الأمر if يتحقّق أم لا يتحقّق.

ب. تتبّع بواسطة جدول متابعة تنفيذ العمليّة Sod بالنسبة للمصفوفة المعطاة a ، والعدد k = 10 . اكتب القيمة التي تُعاد.
يجب أن يشمل جدول المتابعة الأعمدة التي تمّ تفصيلها في البند "أ".

جـ. ما هو ادّعاء خروج العمليّة Sod ؟

د. ما هي تعقيدات زمن تشغيل العمليّة Sod ؟ علّل إجابتك.

هـ. تتبّع بواسطة جدول متابعة تنفيذ العمليّة What بالنسبة للمصفوفة المعطاة a ، والعدد k = 11 . اكتب القيمة التي تُعاد.
يجب أن يشمل جدول المتابعة أعمدة بالنسبة لـ : left, right, a[left], a[right] ، وعمودًا إضافيًّا لكلّ واحد من الأمرين if ، بحيث يُذكَر في كلّ واحد من العمودين إذا كان الشرط الذي في الأمر if يتحقّق أم لا يتحقّق.

و. ما هي تعقيدات زمن تشغيل العمليّة What ؟ علّل إجابتك.

ز. أيّ من العمليّتين – Sod أم What – أكثر نجاعة؟ علّل إجابتك.

حـ. تمّ تغيير ادّعاء دخول العمليّتين Sod وَ What بحيث يمكن نقل مصفوفة a غير مرتَّبة إليهما.

(1) هل يتغيّر ادّعاء خروج العمليّة Sod ؟ علّل إجابتك.

(2) هل يتغيّر ادّعاء خروج العمليّة What ؟ علّل إجابتك.

/ يتبع في صفحة 12 /

6. **انتبه:** لهذا السؤال صيغتان: الأولى بلغة Java (في الصفحتين 12-13)، والثانية بلغة C# (في الصفحتين 14-15). اعمل حسب اللغة التي تعلّمتَها.

للّذين يحلّون بلغة Java

اللعبة "إلعب من فضلك" هي لعبة يلعبها لاعب واحد في مرحلتين. يحصل اللاعب على 52 ورقة شدّة مخلوطة ومختلفة. يوجد على كلّ ورقة عدد بين 1 وَ 13، وإحدى صور من أربع صور. الصور ممثَّلة بواسطة الأعداد 1 حتّى 4.

في المرحلة الأولى يوزّع اللاعب الأوراق إلى أربع كومات حسب صورة الورقة، بحيث تكون في كلّ كومة أوراق عليها نفس الصورة. الأوراق موضوعة الواحدة فوق الأخرى.

في المرحلة الثانية يُجري اللاعب قرعة لعدد بين 1 وَ 4 يمثّل صورة ورقة. يتوجّه اللاعب إلى الكومة التي على جميع أوراقها تتواجد الصورة الممثَّلة بواسطة العدد الذي خرج في القرعة. يأخذ اللاعب الورقة التي في رأس الكومة، وينقلها إلى كومة خامسة.

يتمّ تنفيذ هذه المرحلة حتّى تنتهي اللعبة.

تنتهي اللعبة عندما يخرج في القرعة عدد يمثّل كومة لا توجد فيها أوراق.

الفوز هو حالة يكون فيها مجموع الأعداد التي على الأوراق في الكومة الخامسة يقسم على 100 بدون باقٍ.

بعد أن تنتهي اللعبة يفحصون إذا تحقّق فوز.

لغرض تطبيق اللعبة عُرّفت الفئات: **Card** التي تمثّل ورقة واحدة، وَ **Deck** التي تمثّل الكومات الخمس اللازمة في اللعبة، وَ **Test** التي تُدير اللعبة.

كما وكُتبت في الفئة **Test** العمليّة public static boolean game(Card[] cards)، التي تتلقّى مصفوفة من 52 ورقة مختلفة مرتَّبة عشوائيًّا. العمليّة تُدير اللعبة حتّى نهايتها. تُعيد العمليّة true إذا انتهت اللعبة بفوز، وخلاف ذلك – تُعيد false.

أمامك الفئة **Card** التي تمثّل ورقة عليها عدد (value) بين 1 وَ 13 وصورة (shape) بين 1 وَ 4.

```
public class Card
{
    private int value;
    private int shape;

    public Card (int value, int shape)
    {
        this.value = value;
        this.shape = shape;
    }
    public int getValue () {   return this.value;    }
    public int getShape () {   return this.shape;    }
}
```

/ يتبع في صفحة 13 /

أ. i اكتب بلغة Java عنوان الفئة **Deck** وصفاتها.

ii اكتب بلغة Java في الفئة **Deck** عناوين العمليّات التالية:

- عمليّة بنائيّة تعرّف بأن تكون الكومات الخمس فارغة.
- عمليّة insert تتلقّى ورقة وتُضيفها إلى الكومة الصحيحة حسب المرحلة الأولى في اللعبة.
- عمليّة move تُجري قرعة لعدد يمثّل صورة. إذا كانت كومة الأوراق التي خرجت صورتها في القرعة فارغة، تُعيد العمليّة false. خلاف ذلك – تنقل العمليّة إلى الكومة الخامسة الورقة التي في رأس الكومة التي خرج عددها في القرعة وتُعيد true.
- عمليّة sum تُعيد المجموع الحاليّ للأعداد التي على الأوراق في الكومة الخامسة.

انتبه: في هذا البند الفرعيّ، لا حاجة لتطبيق العمليّات.

ب. طبِّق بلغة Java العمليّة البنائيّة في الفئة **Deck**.

جـ. طبِّق بلغة Java العمليّة sum في الفئة **Deck**.

د. طبِّق بلغة Java العمليّة game في الفئة **Test**.

بإمكانك استعمال العمليّتين insert وَ move للفئة Deck، بدون تطبيقهما.
إذا استعملتَ عمليّات أخرى، عليك تطبيقها.

/ يتبع في صفحة 14 /

للّذين يحلّون بلغة C#

اللعبة "إلعب من فضلك" هي لعبة يلعبها لاعب واحد في مرحلتين. يحصل اللاعب على 52 ورقة شدّة مخلوطة ومختلفة. يوجد على كلّ ورقة عدد بين 1 وَ 13 ، وإحدى صور من أربع صور. الصور ممثَّلة بواسطة الأعداد 1 حتّى 4 .

في المرحلة الأولى يوزّع اللاعب الأوراق إلى أربع كومات حسب صورة الورقة، بحيث تكون في كلّ كومة أوراق عليها نفس الصورة. الأوراق موضوعة الواحدة فوق الأخرى.

في المرحلة الثانية يُجري اللاعب قرعة لعدد بين 1 وَ 4 يمثّل صورة ورقة. يتوجّه اللاعب إلى الكومة التي على جميع أوراقها تتواجد الصورة الممثَّلة بواسطة العدد الذي خرج في القرعة. يأخذ اللاعب الورقة التي في رأس الكومة، وينقلها إلى كومة خامسة.

يتمّ تنفيذ هذه المرحلة حتّى تنتهي اللعبة.

تنتهي اللعبة عندما يخرج في القرعة عدد يمثّل كومة لا توجد فيها أوراق.

الفوز هو حالة يكون فيها مجموع الأعداد التي على الأوراق في الكومة الخامسة يقسم على 100 بدون باقٍ.

بعد أن تنتهي اللعبة يفحصون إذا تحقّق فوز.

لغرض تطبيق اللعبة عُرّفت الفئات: **Card** التي تمثّل ورقة واحدة، وَ **Deck** التي تمثّل الكومات الخمس اللازمة في اللعبة، وَ **Test** التي تُدير اللعبة.

كما وكُتبت في الفئة **Test** العمليّة public static bool Game(Card[] cards) ، التي تتلقّى مصفوفة من 52 ورقة مختلفة مرتَّبة عشوائيًّا. العمليّة تُدير اللعبة حتّى نهايتها. تُعيد العمليّة true إذا انتهت اللعبة بفوز، وخلاف ذلك – تُعيد false.

أمامك الفئة **Card** التي تمثّل ورقة عليها عدد (value) بين 1 وَ 13 وصورة (shape) بين 1 وَ 4.

```
public class Card
{
    private int value;
    private int shape;

    public Card (int value, int shape)
    {
        this.value = value;
        this.shape = shape;
    }
    public int GetValue () {    return this.value;    }
    public int GetShape () {    return this.shape;    }
}
```

/ يتبع في صفحة 15 /

أ. i اكتب بلغة C# عنوان الفئة **Deck** وصفاتها.

ii اكتب بلغة C# في الفئة **Deck** عناوين العمليّات التالية:

- عمليّة بنائيّة تعرّف بأن تكون الكومات الخمس فارغة.
- عمليّة Insert تتلقّى ورقة وتُضيفها إلى الكومة الصحيحة حسب المرحلة الأولى في اللعبة.
- عمليّة Move تُجري قرعة لعدد يمثّل صورة. إذا كانت كومة الأوراق التي خرجت صورتها في القرعة فارغة، تُعيد العمليّة false. خلاف ذلك – تنقل العمليّة إلى الكومة الخامسة الورقة التي في رأس الكومة التي خرج عددها في القرعة وتُعيد true.
- عمليّة Sum تُعيد المجموع الحاليّ للأعداد التي على الأوراق في الكومة الخامسة.

انتبه: في هذا البند الفرعيّ، لا حاجة لتطبيق العمليّات.

ب. طبِّق بلغة C# العمليّة البنائيّة في الفئة **Deck**.

جـ. طبِّق بلغة C# العمليّة Sum في الفئة **Deck**.

د. طبِّق بلغة C# العمليّة Game في الفئة **Test**.

بإمكانك استعمال العمليّتين Insert وَ Move للفئة Deck، بدون تطبيقهما.
إذا استعملتَ عمليّات أخرى، عليك تطبيقها.

/ يتبع في صفحة 16 /

الفصل الثالث (25 درجة)

في هذا الفصل أسئلة في أربعة مسارات:
أنظمة حاسوب وأسمبلي، ص 16-17.
مقدّمة لبحث الأداءات، ص 18-24.
موديلات حسابيّة، ص 25-26.
برمجة موجّهة كائنات بلغة Java، ص 27-31؛ برمجة موجّهة كائنات بلغة #C ، ص 32-36.

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 توجد علاقة بينهما. أجب عن البندين.

أ. أمامك قطعة برنامج بلغة أسمبلي:

انتبه: المعطيات هي بأساس سداسيّ عشريّ.

```
START:  MOV     AX , C83BH
        MOV     BX , A89CH
        SHL     AX , 1
        OR      AL , 33H
        NOT     BL
        ADD     AX , BX
```

تتبّع بواسطة جدول متابعة تنفيذ قطعة البرنامج.
فصّل في جدول المتابعة محتوى CF ، SF ، ZF ، BX ، AX في كلّ مرحلة.

/ يتبع في صفحة 17 /

ب. (لا علاقة له مع البند "أ" .)

في مقطع المعطيات عُرِّفت المصفوفة:

```
ARR  DB  5  DUP (?)
```

أمامك 4 قطع iv-i بلغة أسمبلي هدفها ابتداء جميع خلايا المصفوفة ARR بـِ 0 .

تتبّع بواسطة جدول متابعة تنفيذ كلّ واحدة من القطع iv-i التي أمامك، وحدّد إذا كانت تنفّذ المطلوب أم لا تنفّذ المطلوب.

i

```
        MOV     SI , 0
        MOV     CX , 4
A1:     MOV     ARR[SI] , 0
        INC     SI
        LOOP    A1
```

---

ii

```
        MOV     CX , 5
        LEA     BX , ARR
        MOV     AL , 0
A1:     MOV     [BX] , AL
        INC     BX
        LOOP    A1
```

---

iii

```
        MOV     BX , 5
        DEC     BX
A1:     MOV     ARR[BX] , 0
        DEC     BX
        JNZ     A1
```

---

iv

```
        MOV     DI , 0
A1:     MOV     ARR[DI] , 0
        INC     DI
        CMP     DI , 5
        JC      A1
```

8. خُزِّن في الخازن AX عدد بينارِيّ. اكتب قطعة برنامج بلغة أسمبلي، تُعوِّض في الخازن BL عدد المرّات التي يظهر فيها التسلسل 1011 في العدد البينارِيّ الذي في الخازن AX.

مثال: إذا كان في الخازن AX مخزونًا العدد البينارِيّ: 0110110010110110 ،

يُعوَّض في الخازن BL العدد 3.

/ يتبع في صفحة 18 /

مقدّمة لبحث الأداءات

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

9. معطاة مسألة تخطيط خطّيّ :

$$\text{Max}\{z = 5x_1 - x_2\}$$

تخضع للاضطرارات التالية :

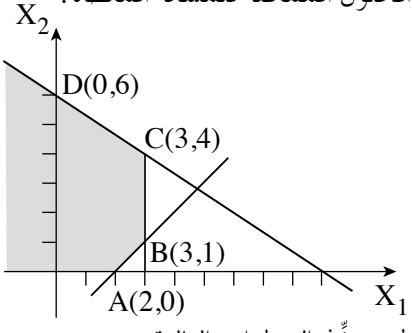
$$2x_1 + 3x_2 \le 18$$
$$x_1 - x_2 \le 2$$
$$x_1 \le 3$$
$$x_2 \ge 0$$

x_1 ليس مقيّدًا بإشارة .

معطى أيضًا أنّ الحلّ $(3,1)$ هو حلّ أمثل للمسألة .

يتطرّق كلّ واحد من البنود "أ–هـ" التي أمامك إلى مسألة التخطيط الخطّيّ المعطاة .
البنود لا تتعلّق ببعضها البعض . أجب عن جميع البنود .

أ. أمامك رسم لمجال الحلول الممكنة للمسألة المعطاة .



انسخ الرسم إلى دفترك، ونفِّذ الخطوات التالية :

الخطوة 1 : أضِف إلى الرسم الذي في دفترك مسقط ارتفاع دالّة الهدف بالنسبة لـ $z = 5$.
احسب إحداثيّات نقاط تقاطع هذا المسقط مع المحورين x_1 وَ x_2 ، وأشر إليها في الرسم .

الخطوة 2 : أضِف إلى الرسم الذي في دفترك مسقط ارتفاع دالّة الهدف بالنسبة لـ $z = 15$.
احسب إحداثيّات نقاط تقاطع هذا المسقط مع المحورين x_1 وَ x_2 ، وأشِر إليها في الرسم .

الخطوة 3 : بواسطة سهم، أشِر في الرسم الذي في دفترك، إلى اتّجاه تصاعد دالّة الهدف .

هل تُبيِّن الخطوات 1-3 أنّ الحلّ $(1, 3)$ هو الحلّ الأمثل الوحيد؟ علّل إجابتك .

/يتبع في صفحة 19 /

ب. نضيف إلى المسألة المعطاة في بداية السؤال الاضطرار:

$$x_1 - x_2 \geq 2$$

ارسم في دفترك هيئة محاور جديدة، وأشِر فيها فقط إلى مجال الحلول الممكنة الذي ينتج بعد إضافة هذا الاضطرار.

جـ. نضيف إلى المسألة المعطاة في بداية السؤال اضطرارًا جديدًا معيّنًا، بحيث يكون مجال الحلول الممكنة بعد إضافته هو القطعة التي بين النقطتين (4 , 3) وَ (1 , 3) .

هل يتغيّر الحلّ الأمثل المعطى (1 , 3) ؟ علّل إجابتك.

د. نضيف إلى المسألة المعطاة في بداية السؤال اضطرارًا جديدًا:

$$3x_1 + 2x_2 \leq 14$$

ادّعى أحد الطلّاب أنّ الحلّ الأمثل المعطى (1 , 3) لن يتغيّر.

هل هذا الطالب على حقّ؟ علّل إجابتك بدون أن ترسم من جديد مجال الحلول الممكنة بعد إضافة هذا الاضطرار.

هـ. نغيّر فقط دالّة هدف المسألة التي في بداية السؤال إلى $z = 3x_1$.

أمامك الأقوال i-iv ، واحد منها فقط صحيح. انسخ القول الصحيح إلى دفترك، وعلّل اختيارك.

i الحلّ الأمثل الوحيد سيكون (1 , 3).

ii الحلّ الأمثل الوحيد سيكون (4 , 3).

iii الحلّ الأمثل الجديد سيكون:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \lambda \begin{pmatrix} 3 \\ 1 \end{pmatrix} + (1-\lambda) \begin{pmatrix} 3 \\ 4 \end{pmatrix}$$

عندما $0 \leq \lambda \leq 1$

iv الحلّ الأمثل لن يكون محصورًا.

/ يتبع في صفحة 20 /

10. في هذا السؤال ستّة بنود، "أ – و"، لا علاقة بينها. أجب عن جميع البنود.

أ. أمامك مسألة نقل فيها مصدران وَ 4 غايات. تكاليف الوحدة من كلّ مصدر إلى كلّ غاية معطاة في الجدول الذي أمامك.

المصادر	الغايات				العرض
	1	2	3	4	
1	8	9	4	6	100
2	14	12	13	8	80
الطلب	10	40	30	70	

حسب الجدول المعطى لا يمكن تفعيل طريقة الزاوية الشماليّة – الغربيّة لإيجاد حلّ ممكن لمسألة النقل. فسّر لماذا.

ب. في الجدول الذي أمامك معطى جزء من حلّ أساسيّ ممكن لمسألة نقل:

$x_{12}=2$ ، $x_{11}=10$.

المصادر	الغايات			العرض
	1	2	3	
1	4 (10)	5 (2)	10	12
2	6	3	6	12
3	6	2	6	6
الطلب	10	10	10	

انسخ الجدول إلى دفترك، وأكمله حسب طريقة الزاوية الشماليّة – الغربيّة.

/ يتبع في صفحة 21 /

جـ. في الجدول الذي أمامك معطى حلّ أساسيّ ممكن لمسألة نقل. القيمة التي حُدِّدت لِـ u_1 هي 0. انسخ الجدول إلى دفترك، واحسب قيمة v_1 ، v_2 ، v_3 ، u_2 ، u_3 ، واكتب القيَم التي حسبتَها في الأماكن الملائمة في الجدول.

المصادر	الغايات			العرض	u_i
	1	2	3		
1	4 (10)	5 (2)	10	12	0
2	6	3 (2)	6 (10)	12	
3	6	2 (6)	6	6	
الطلب	10	10	10		
v_j					

(انتبه: تكملة السؤال في الصفحة التالية.) / يتبع في صفحة 22 /

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

المصادر	**الغايات**			**العرض**	u_i
	1	2	3		
1	4 (20)	5 (4)	10	24	0
2	2	6 (16)	3 (4)	20	1
3	6	2	6 (8)	8	4
الطلب	20	20	12		
v_j	4	5	2		

هل الحلّ هو أمثل؟ علّل إجابتك.

/ يتبع في صفحة 23 /

هـ. في الجدول الذي أمامك معطى حلّ لمسألة نقل،
ومعطاة قيَم v_1 ، v_2 ، v_3 ، u_1 ، u_2 ، u_3 ، التي تلائم هذ الحلّ.

المصادر	الغايات			العرض	u_i
	1	2	3		
1	10 (20)	25	30	20	10
2	10 (30)	22	14 (20)	50	10
3	16	20 (40)	20 (20)	60	16
الطلب	50	40	40		
v_j	0	4	4		

أمامك الأقوال i-iv ، واحد منها فقط صحيح. انسخ القول الصحيح إلى دفترك، وعلّل اختيارك.

i الحلّ المعطى ليس حلّاً ممكنًا.

ii الحلّ المعطى هو حلّ أساسيّ ممكن لكنّه ليس أمثل.

iii الحلّ المعطى هو حلّ أمثل وحيد.

iv الحلّ المعطى هو حلّ أمثل لكنّه ليس حلّاً أمثل وحيدًا.

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 24 /

و. في الجدول الذي أمامك معطى جزء من حلّ أساسيّ ممكن لمسألة نقل تمّ حسابه حسب طريقة الزاوية الشماليّة – الغربيّة، ومعطاة قيَم v_1 ، v_2 ، v_3 ، u_1 ، u_2 ، u_3 التي تمّ حسابها حسب هذا الحلّ.

المصادر	**الغايات**			**العرض**	u_i
	1	2	3		
1	12 (20)	15	17	20	2
2	10	18 (10)	14	10	0
3	20	10 (5)	18 (10)	15	−8
الطلب	20	15	10		
v_j	10	18	26		

ما الذي يجب إضافته إلى الجدول للحصول على حلّ أساسيّ ممكن يلائم جميع المعطيات التي في الجدول؟

انسخ الإجابة الصحيحة إلى دفترك من بين الإمكانيّات i-iv، وعلّل إجابتك.

i $X_{12} = 0$.

ii $X_{21} = 0$.

iii $X_{13} = 0$.

iv $X_{31} = 0$.

/ يتبع في صفحة 25 /

موديلات حسابيّة

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

11. أمامك اللغة L فوق الأبجديّة $\{\$, 0\}$:

$$L = \left\{ 0^3 \$ 0^{i_1} \$ 0^{i_2} \$ \ldots 0^{i_k} \$ \;\middle|\; \begin{array}{r} k \geq 1 \text{ ,} \\ \text{لكلّ } m \text{ بين } 1 \text{ وَ } k : i_m \geq 0 \\ i_m \text{ وَ } \text{ يقسم على } 3 \text{ بدون باقٍ} \end{array} \right\}$$

أ. اكتب أقصر كلمة في اللغة L .

ب. ابنِ أوتوماتًا نهائيًّا محدودًا يتلقّى اللغة L .

/ يتبع في صفحة 26 /

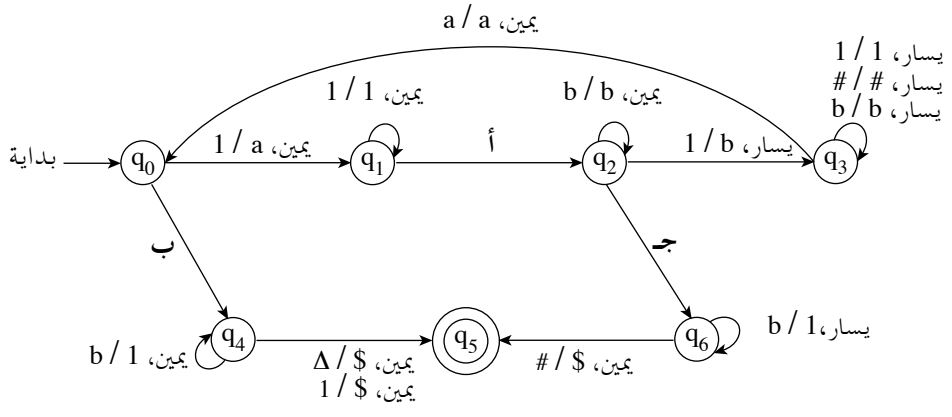
12. بُنيت آلة تيورينج تحسب الدالّة $f(m,n)$. تجد هذه الآلة أصغر عدد من العددين m وَ n .
تتلقّى الآلة كمُدخَل عددين m وَ n صحيحين وأكبر من 0 . العددان مسجّلان على الشريط كعددين أونارّيين (العدد الأونارّيّ m هو عدد ممثَّل بواسطة m رموز من 1) وبين العددين مسجَّلة الإشارة # .
مُخرَج الآلة هو أصغر عدد من العددين، ويُسجَّل على الشريط كعدد أونارّيّ بين إشارتَي $.
الإشارة $ اليسرى تُسجَّل بدلاً من الإشارة # ، والعدد يُسجَّل عن يمينها.
خلال العمليّة تستطيع الآلة الاستعانة بالإشارتين a ، b .
مثال: بالنسبة للمُدخلَين $m=3$ وَ $n=5$ ، قبل الحساب يبدو شريط الذاكرة على النحو التالي:

⊢	1	1	1	#	1	1	1	1	1	Δ	Δ	Δ	…

بعد إنهاء الحساب يبدو الشريط على النحو التالي:

⊢	…	…	…	$	1	1	1	$	…

ملاحظة: الخلايا المشار إليها بِـ… هي خلايا لا أهمّيّة لمحتواها.
أمامك رسم جزئيّ للآلة.



أ. توجد في الشريط ثلاثة انتقالات أُشير إليها بالأحرف أ–جـ.
تنقص في هذه الانتقالات إشارات الإدخال والأوامر.
انسخ الرسم إلى دفترك، وأكمل فيه الانتقالات الثلاثة الناقصة بحيث تحسب الآلة الدالّة $f(m,n)$.

ب. بيّن عمليّة الحساب في الآلة بالنسبة للمُدخلَين $m=1$ ، $n=1$.
في كلّ مرحلة، اكتب حالة الشريط، وأشر أين يتواجد رأس الآلة، واكتب في أيّ من الحالات q_0-q_6 تتواجد الآلة.

/ يتبع في صفحة 27 /

برمجة موجّهة كائنات

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

13. أمامك جزء من مشروع يتناول وسائل مواصلات ويشمل الفئات التالية:

الفئة Vehicle	تمثّل وسيلة مواصلات
الفئة Train	تمثّل قطارًا الذي هو وسيلة مواصلات
الفئة Boat	تمثّل قاربًا الذي هو وسيلة مواصلات
الفئة Airplane	تمثّل طائرة التي هي وسيلة مواصلات
الفئة TransportationCompany	تمثّل شركة لديها وسائل مواصلات من أنواع مختلفة

```
public class Vehicle
{
    private String type;          //  الموقع ( أرض / مياه / جوّ )
    private String way;           //  نوع الطريق ( شارع / سكّة حديد / نهر / ... )
    private int maxSpeed;         //  السرعة القصوى

    public Vehicle(String type, String way, int maxSpeed)
    {
        this.type = type;
        this.way = way;
        this.maxSpeed = maxSpeed;
    }
}

public class Train extends Vehicle
{
    private int numOfCarriages;          //  عدد عربات القطار

    public Train(int maxSpeed, int numOfCarriages)
    {
        super("land", "tracks", maxSpeed);
        this.numOfCarriages = numOfCarriages;
    }

    public void incNumOfCarriages(int n)  //  تُكبِّر بـِ n عدد العربات في القطار
    {
        this.numOfCarriages = this.numOfCarriages + n;
    }
}
```

/ يتبع في صفحة 28 /

```
public class Boat extends Vehicle
{
    public Boat(String way, int maxSpeed)
    {
        super("water", way, maxSpeed);
    }
}

public class Airplane extends Vehicle
{
    private int maxHeight;                          // ارتفاع الطيران الأقصى
    public Airplane(int maxSpeed, int maxHeight)
    {
        super("sky" , "air" , maxSpeed);
        this.maxHeight = maxHeight;
    }
}

public class TransportationCompany
{
    private Vehicle[] vehicles = new Vehicle[50];   // مصفوفة وسائل المواصلات في الشركة
    private int counter = 0;                        // عدد وسائل المواصلات الموجودة فعليًّا
    public TransportationCompany()
    {
    }

    public void addVehicle (Vehicle v)  // تُضيف وسيلة مواصلات إلى مصفوفة وسائل
    {                                   // المواصلات في الشركة. افترض أنّه يوجد مكان
                                        // لإضافة وسيلة مواصلات.
        this.vehicles[counter] = v;
        this.counter++;
    }
}
```

(انتبه: بنود السؤال في الصفحة التالية.)

/ يتبع في صفحة 29 /

أ. طبِّق بلغة Java فئة رئيسيّة Program فيها عمليّة رئيسيّة، تنفّذ المهمّتين التاليتين:

i بناء كائن من نمط شركة وسائل مواصلات — **TransportationCompany**
يُسمّى company1 .

ii إضافة قارب واحد وقطار واحد إلى الشركة company1 .
اختر قيَمًا للصفات كما تشاء.

ب. في الفئة **TransportationCompany** عُرِّفت العمليّة:

```
public void display()
{
    for (int i=0; i<this.counter; i++)
    {
        System.out.println((i+1) + ":" + this.vehicles[i]);
    }
}
```

طبِّق بلغة Java عمليّات تُمكّن تنفيذًا سليمًا للعمليّة display() ، بحيث تُطبَع لكلّ وسيلة مواصلات جميع صفاتها. عرِّف العمليّات بالطريقة الأكثر ملاءمة لمبادئ البرمجة الموجّهة كائنات (تعليب — encapsulation ، توريث — inheritance ، تعدّد الأشكال — polymorphism) .
بالنسبة لكلّ عمليّة تطبّقها، اكتب إلى أيّ فئة تتبع.
لا يمكن تغيير العمليّة display() .

جـ. طبِّق بلغة Java عمليّة، تتلقّى عددًا صحيحًا n وتُضيف n عربات إلى كلّ القطارات التي تتبع للشركة التي لديها وسائل مواصلات من أنواع مختلفة. وثِّق العمليّة، واكتب في أيّ فئة يجب تعريفها. لا يمكن تغيير العمليّات الموجودة في المشروع.

/ يتبع في صفحة 30 /

14. أمامك الفئتان **AA** وَ **BB** :

```
public class AA
{
    private String st;

    public AA()                 {  this.st = "excellent";  }
    public AA(String st)        {  this.st = st;           }
    public String getSt()       {  return this.st;         }
    public void setSt(String st){  this.st = st;           }
    public String toString()    {  return "st = " + this.st; }
}

public class BB extends AA
{
    private int num;

    public BB()                 {  super();  this.num = 1;  }
    public BB(int num , String st) {  super(st); this.num = Math.abs(num);  }
    public int getNum()         {  return this.num;        }
    public void setNum(int num) {  this.num = num;}
    public String toString()    {  return super.toString() + " num = " + this.num;}
}
```

أ. عرِّف في الفئة **AA** عمليّة بوليانيّة باسم isLike(Object obj)، تتلقّى كائنًا obj من نمط Object. إذا كان الكائن obj من نمط **AA** وكذلك محتوى النصّ st التابع لِـ obj مطابقًا لِمحتوى النصّ st التابع للكائن الحاليّ – تُعيد العمليّة true، خلاف ذلك – تُعيد false.

ب. عرِّف في الفئة **BB** عمليّة تدهس العمليّة التي عرّفتَها في البند "أ".
إذا كان الكائن obj من نمط **BB** وكذلك قيمة صفته num مساوية لقيمة الصفة num التابعة للكائن الحاليّ – تُعيد العمليّة true ، خلاف ذلك – تُعيد false.

/ يتبع في صفحة 31 /

جـ. أمامك قطعة من عمليّة رئيسيّة:

```
AA  a = new AA("excellent");
BB  b = new BB();
a = b;
if (a.isLike(b))  System.out.println(a);
```

هل قطعة البرنامج سليمة؟

إذا كانت سليمة – ماذا يكون مُخرَج القطعة؟ اكتب أيّة صيغة للعمليّة isLike تُفعَّل – الصيغة التابعة لِـ AA أم الصيغة التابعة لِـ BB.

إذا كانت غير سليمة – اشرح ما هو الخطأ ومتى سيظهر: أثناء التجميع والتحويل للغة الآلة أم أثناء التشغيل.

د. أمامك قطعة من عمليّة رئيسيّة:

```
AA  aa = new AA();
BB  bb = new BB(2 , "excellent");
bb = aa;
if (bb.isLike(aa))  System.out.println(bb);
```

هل قطعة البرنامج سليمة؟

إذا كانت سليمة – ماذا يكون مُخرَج القطعة؟ اكتب أيّة صيغة للعمليّة isLike تُفعَّل – الصيغة التابعة لِـ AA أم الصيغة التابعة لِـ BB.

إذا كانت غير سليمة – اشرح ما هو الخطأ ومتى سيظهر: أثناء التجميع والتحويل للغة الآلة أم أثناء التشغيل.

هـ. اكتب عمليّة خارجيّة باسم longString تتلقّى مصفوفة لكائنات من نمط Object .

تُعيد العمليّة نصًّا مركَّبًا من دمج الصفة st لكائنات من نمط AA في المصفوفة، على النحو التالي:

- إذا كانت للكائن الصفة st فقط، النصّ الذي في الصفة st يُدمَج مرّة واحدة.
- إذا كانت للكائن الصفة num أيضًا، النصّ الذي في الصفة st يُدمَج num مرّات.
- إذا لم يكن في المصفوفة أيّ كائن من نمط AA ، يُعاد نصّ فارغ.

/ يتبع في صفحة 32 /

برمجة موجّهة كائنات

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

15. أمامك جزء من مشروع يتناول وسائل مواصلات ويشمل الفئات التالية:

الفئة Vehicle	تمثّل وسيلة مواصلات
الفئة Train	تمثّل قطارًا الذي هو وسيلة مواصلات
الفئة Boat	تمثّل قاربًا الذي هو وسيلة مواصلات
الفئة Airplane	تمثّل طائرة التي هي وسيلة مواصلات
الفئة TransportationCompany	تمثّل شركة لديها وسائل مواصلات من أنواع مختلفة

```
public class Vehicle
{
    private string type;          //  الموقع ( أرض / مياه / جوّ )
    private string way;           //  نوع الطريق ( شارع / سكّة حديد / نهر / ... )
    private int maxSpeed;         //  السرعة القصوى

    public Vehicle(string type, string way, int maxSpeed)
    {
        this.type = type;
        this.way = way;
        this.maxSpeed = maxSpeed;
    }
}

public class Train : Vehicle
{
    private int numOfCarriages;          //  عدد عربات القطار
    public Train(int maxSpeed, int numOfCarriages) : base("land", "tracks",maxSpeed)
    {
        this.numOfCarriages = numOfCarriages;
    }
    public void incNumOfCarriages(int n)  //  تُكبِّر بـِ n عدد العربات في القطار
    {
        this.numOfCarriages = this.numOfCarriages + n;
    }
}
```

/ يتبع في صفحة 33 /

```
public class Boat : Vehicle
{
    public Boat(string way, int maxSpeed) : base("water", way, maxSpeed)
    {

    }
}

public class Airplane : Vehicle
{
    private int maxHeight;                          //          ارتفاع الطيران الأقصى
    public Airplane(int maxSpeed, int maxHeight) : base("sky", "air", maxSpeed)
    {
        this.maxHeight = maxHeight;
    }
}

public class TransportationCompany
{
    private Vehicle[] vehicles = new Vehicle[50];   // مصفوفة وسائل المواصلات في الشركة
    private int counter = 0;                        //  عدد وسائل المواصلات الموجودة فعليًا
    public TransportationCompany()
    {
    }

    public void AddVehicle (Vehicle v) //        تُضيف وسيلة مواصلات إلى مصفوفة وسائل
    {                                           المواصلات في الشركة. افترض أنّه يوجد مكان
                                        //      لإضافة وسيلة مواصلات
        this.vehicles[counter] = v;
        this.counter++;
    }
}
```

(انتبه: بنود السؤال في الصفحة التالية.)

/ يتبع في صفحة 34 /

أ. طبِّق بلغة C# فئة رئيسيّة Program فيها عمليّة رئيسيّة، تنفّذ المهمّتين التاليتين:

i بناء كائن من نمط شركة وسائل مواصلات — **TransportationCompany** يُسمّى company1 .

ii إضافة قارب واحد وقطار واحد إلى الشركة company1.
اختر قيَمًا للصفات كما تشاء.

ب. في الفئة **TransportationCompany** عُرِّفت العمليّة:

```
public void Display()
{
    for (int i=0; i<this.counter; i++)
    {
        Console.WriteLine((i+1) + ":" + this.vehicles[i]);
    }
}
```

طبِّق بلغة C# عمليّات تُمكّن تنفيذًا سليمًا للعمليّة Display() ، بحيث تُطبَع لكلّ وسيلة مواصلات جميع صفاتها. عرِّف العمليّات بالطريقة الأكثر ملاءمة لمبادئ البرمجة الموجّهة كائنات (تعليب — encapsulation ، توريث — inheritance ، تعدّد الأشكال — polymorphism).
بالنسبة لكلّ عمليّة تطبّقها، اكتب إلى أيّ فئة تتبع.
لا يمكن تغيير العمليّة Display().

جـ. طبِّق بلغة C# عمليّة، تتلقّى عددًا صحيحًا n وتُضيف n عربات إلى كلّ القطارات التي تتبع للشركة التي لديها وسائل مواصلات من أنواع مختلفة. وثِّق العمليّة، واكتب في أيّ فئة يجب تعريفها. لا يمكن تغيير العمليّات الموجودة في المشروع.

/ يتبع في صفحة 35 /

16. أمامك الفئتان **AA** وَ **BB** :

```
public class AA
{
    private string st;

    public AA()                          { this.st = "excellent";    }
    public AA(string st)                 { this.st = st;             }
    public string GetSt()                { return this.st;           }
    public void SetSt (string st)        { this.st = st;             }
    public override string ToString()    { return "st = " + this.st; }
}

public class BB : AA
{
    private int num;

    public BB() : base()                          { this.num = 1;    }
    public BB(int num, string st) : base(st)      { this.num = Math.Abs(num);      }
    public int GetNum()                           { return this.num; }
    public void SetNum(int num)                   { this.num = num;}
    public override string ToString()             { return base.ToString() + " num = "
                                                    + this.num;}
}
```

أ. عرِّف في الفئة **AA** عمليّة بوليانيّة باسم IsLike (Object obj)، تتلقّى كائنًا obj من نمط Object. إذا كان الكائن obj من نمط **AA** وكذلك محتوى النصّ st التابع لِـ obj مطابقًا لِمحتوى النصّ st التابع للكائن الحاليّ – تُعيد العمليّة true، خلاف ذلك – تُعيد false.

ب. عرِّف في الفئة **BB** عمليّة تدهس العمليّة التي عرّفتَها في البند "أ".
إذا كان الكائن obj من نمط **BB** وكذلك قيمة صفته num مساوية لقيمة الصفة num التابعة للكائن الحاليّ – تُعيد العمليّة true ، خلاف ذلك – تُعيد false.

(انتبه: تكملة السؤال في الصفحة التالية.)

/ يتبع في صفحة 36 /

جـ. أمامك قطعة من عمليّة رئيسيّة:

```
AA a = new AA("excellent");
BB b = new BB();
a = b;
if (a.IsLike(b)) Console.WriteLine(a);
```

هل قطعة البرنامج سليمة؟

إذا كانت سليمة – ماذا يكون مُخرَج القطعة؟ اكتب أيّة صيغة للعمليّة IsLike تُفعِّل – الصيغة التابعة لِـ AA أم الصيغة التابعة لِـ BB.

إذا كانت غير سليمة – اشرح ما هو الخطأ ومتى سيظهر: أثناء التجميع والتحويل للغة الآلة أم أثناء التشغيل.

د. أمامك قطعة من عمليّة رئيسيّة:

```
AA  aa = new AA();
BB  bb = new BB(2 , "excellent");
bb = aa;
if (bb.IsLike(aa)) Console.WriteLine(bb);
```

هل قطعة البرنامج سليمة؟

إذا كانت سليمة – ماذا يكون مُخرَج القطعة؟ اكتب أيّة صيغة للعمليّة IsLike تُفعِّل – الصيغة التابعة لِـ AA أم الصيغة التابعة لِـ BB.

إذا كانت غير سليمة – اشرح ما هو الخطأ ومتى سيظهر: أثناء التجميع والتحويل للغة الآلة أم أثناء التشغيل.

هـ. اكتب عمليّة خارجيّة باسم LongString تتلقّى مصفوفة لكائنات من نمط Object .

تُعيد العمليّة نصًّا مركَّبًا من دمج الصفة st لكائنات من نمط AA في المصفوفة، على النحو التالي:

- إذا كانت للكائن الصفة st فقط، النصّ الذي في الصفة st يُدمَج مرّة واحدة.
- إذا كانت للكائن الصفة num أيضًا، النصّ الذي في الصفة st يُدمَج num مرّات.
- إذا لم يكن في المصفوفة أيّ كائن من نمط AA ، يُعاد نصّ فارغ.

בהצלחה!
نتمنى لك النجاح!

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.