الفصل التاسع

تطوير علاقات المرافقة بين الكائنات
IMPLEMENTING ASSOCIATION RELATIONSHIPS

أهداف الفصل:

- التعرف على علاقات المرافقة لكائنات برادشو مارينا.
- إقامة علاقة ترافق من النوع واحد إلى واحد لأصناف VB .NET.
- إضافة مهام إضافية للصنف Boat.
- إقامة علاقة ترافق من النوع واحد إلى مجموعة بين كائنات الصنف Dock وكائنات الصنف Slip.
- إضافة الصنف Boat والصنف Customer لمثال الصنف Slip.
- إنشاء أصناف المرافقة للصنف Lease واستخدامها.

لقد تعلمنا في الفصل الثامن كيف نطور أشجار التعميم/التخصيص التي تمثل علاقات التوارث بين الأصناف الرئيسة والأصناف الفرعية. يوضح نموذج Class Diagram علاقات التوريث، كما يوضح نوعاً آخر من العلاقات والتي نطلق عليها علاقات المرافقة (Association Relationships) بين الأصناف والتي تمثل درجة كبيرة من الأهمية في تطوير نظم المعلومات. تمثل علاقة المرافقة بين الأصناف على شكل خط في نموذج Class Diagram وتوصف بجملة فعلية. فعلى سبيل المثال يحتوي نموذج Class Diagram لشركة برادشو مارينا على العديد من العلاقات المرافقة وهي: العميل (الصنف Customer) يملك مركباً (الصنف Boat) والمركب يرسى في مرسى (الصنف Slip). إن أهمية هذه المعلومات لدى مديري شركة برادشو مارينا لا تقل أهمية عن معرفة اسم العميل وعنوانه ومصنّع المركب وطوله.

سوف تتعلم كيف تطور علاقات المرافقة بين الأصناف مستخدماً لغة VB .NET. وعندما تطور علاقة واحد إلى واحد (One to One) سوف تتنقل من كائن إلى كائن. فعلى سبيل المثال، يمكنك الانتقال بين الكائنات للحصول على معلومات حول العميل الذي يملك مركباً ما. كذلك سوف تتعلم أيضاً تطوير علاقة واحد إلى مجموعة ومن ثم

يمكنك الحصول على معلومات مراسي رصيف ما وذلك بالانتقال من كائن الصنف Dock إلى كائنات الصنف Slip. وأخيراً سوف نتعلم كيف ننشئ صنف مرافقة (Association Class) وهو الصنف Lease الذي قدمناه في الفصل الثامن.

التعرف على علاقات المرافقة بين أصناف نظام شركة برادشو مارينا

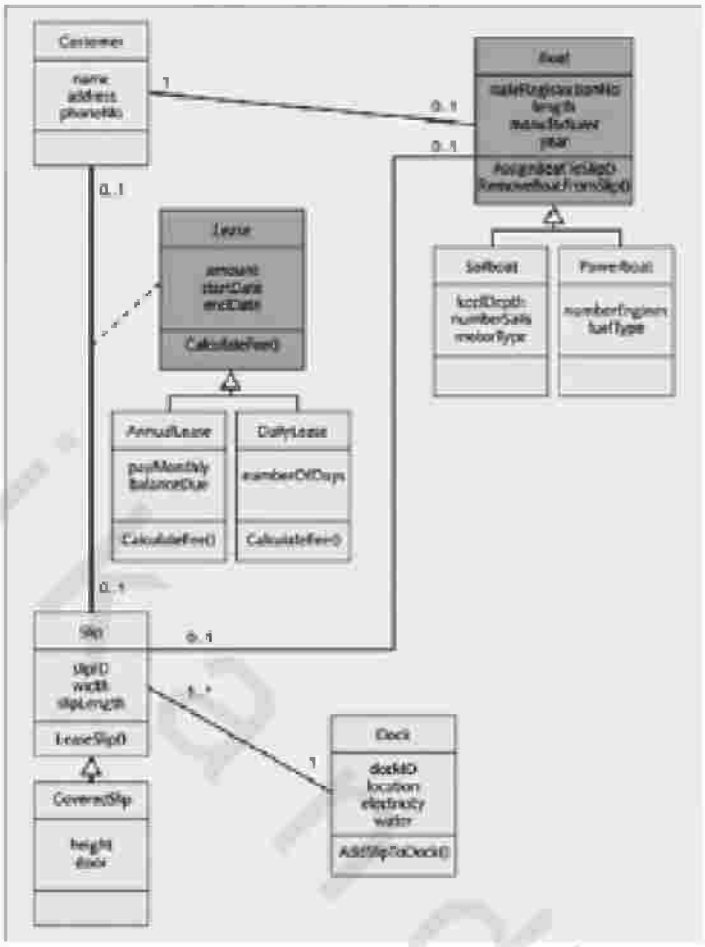
Identifying Association Relationships on Bradshaw Marina's Class Diagram

لقد تعرفنا على متطلبات نظام شركة برادشو خلال الفصل الخامس ومن ثم قدمنا نموذج Class Diagram الذي يوصف أصناف مجال المشكلة التي يحتاجها النظام (يوضح الشكل رقم ٩.١ هذا النموذج). وقد طورنا حتى الآن الصنف Customer والصنف Slip والصنف Boat (بالإضافة إلى أصنافه الفرعية) والصنف Lease (بالإضافة إلى أصنافه الفرعية). يوضح نموذج Class Diagram كيف تترافق (أو كيف ترتبط) كائنات الأصناف سوياً. فعلى سبيل المثال فإنه من الضروري معرفة أي مركب يرسى على مرسى محدد، وأي عميل يملك مركباً محدداً. تشير علاقات المرافقة على نموذج Class Diagram إلى أن النظام يتطلب معلومات حول هذه العلاقات.

تتمثل علاقات المرافقة على النموذج Class Diagram بخطوط تربط كائنات صنف ما بصنف آخر. ويشير الرقم المدون على نهاية الخط بعدد الكائنات التي تدخل في هذه العلاقة (تذكر أن ١..١ تعني واحد وواحد فقط وأن *..١ تعني واحد أو أكثر). باستخدام علاقات المرافقة، يمكنك الانتقال من كائن إلى آخر. فعلى سبيل المثال، إذا كان لديك كائن الصنف Customer فبإمكانك الحصول على كائن الصنف Boatالمرتبط به (المملوك له). وبالمثل لو كان لديك كائن الصنف Boat فيمكنك الحصول على كائن الصنف Slip (المرسى المخصص لهذا المركب).

عادة ما تحتوي نظم المعلومات على العديد من علاقات المرافقة التي تمثل درجة عالية من الأهمية لاحتياجات العمل لأنها تشير إلى العلاقات التي يجب تذكرها (حفظها بواسطة النظام). يوضح نموذج Class Diagram العديد من علاقات المرافقة لنظام برادشو والتي تشير إلى معلومات يجب الاحتفاظ بها:

- العميل يملك مركباً.
- المركب مملوك للعميل.
- المركب خُصِّص له مرسى.
- المرسى يحتوي على مركب.
- الرصيف يحتوي على مراسي.
- المرسى يمثل جزءاً من الرصيف.
- المرسى يؤجر إلى العميل (من خلال صنف الترافق Lease).
- العميل يؤجر مرسى (من خلال صنف الترافق Lease).



الشكل رقم (٩,١). نموذج Class Diagram لشركة برادشو.

وأحياناً تأخذ علاقة المرافقة شكلاً أكثر قوة بين الكائنات، حيث يوجد نوعان من العلاقات وهما: علاقة تجميعية (Aggregation Relationship) وعلاقات تكوينية (Composition Relationship) والتي نرمز لهما بخط ينتهي بشكل معين. تتسم العلاقات التجميعية بالترافق القوي بين الكائنات، حيث يحتوي كائن ما على كائن آخر، مثل المدينة تحتوي على مراكز تجارية. وأيضاً فإن علاقات التكوين تتسم بعلاقة ترافق قوية بين الكائنات، حيث يتكون كائن ما من كائنات أخرى مثل أن الحائط "جزء من" المبنى (أو المبنى يتكون من عدة حوائط). عزيزي القارئ، إن تطوير كل من علاقات التجميع وعلاقات التكوين بين الكائنات مثل تطوير علاقات الترافق والتي سوف نتعلم كيفية تطويرها خلال هذا الفصل. أحياناً لا يذكر بعض محللي النظم هاتين العلاقتين أثناء مرحلة التحليل مطلقاً. ويجب أن تلاحظ أن نموذج Class Diagram الخاص بنظام شركة برادشو مارينا لا يحتوي على هذا النوع من العلاقات. مع أنه ربما تختلف نظرة محلل نظم آخر إلى أن العلاقة بين الرصيف والمراسي تنتمي إلى علاقات التكوين حيث يتكون الرصيف من عدة مراسي.

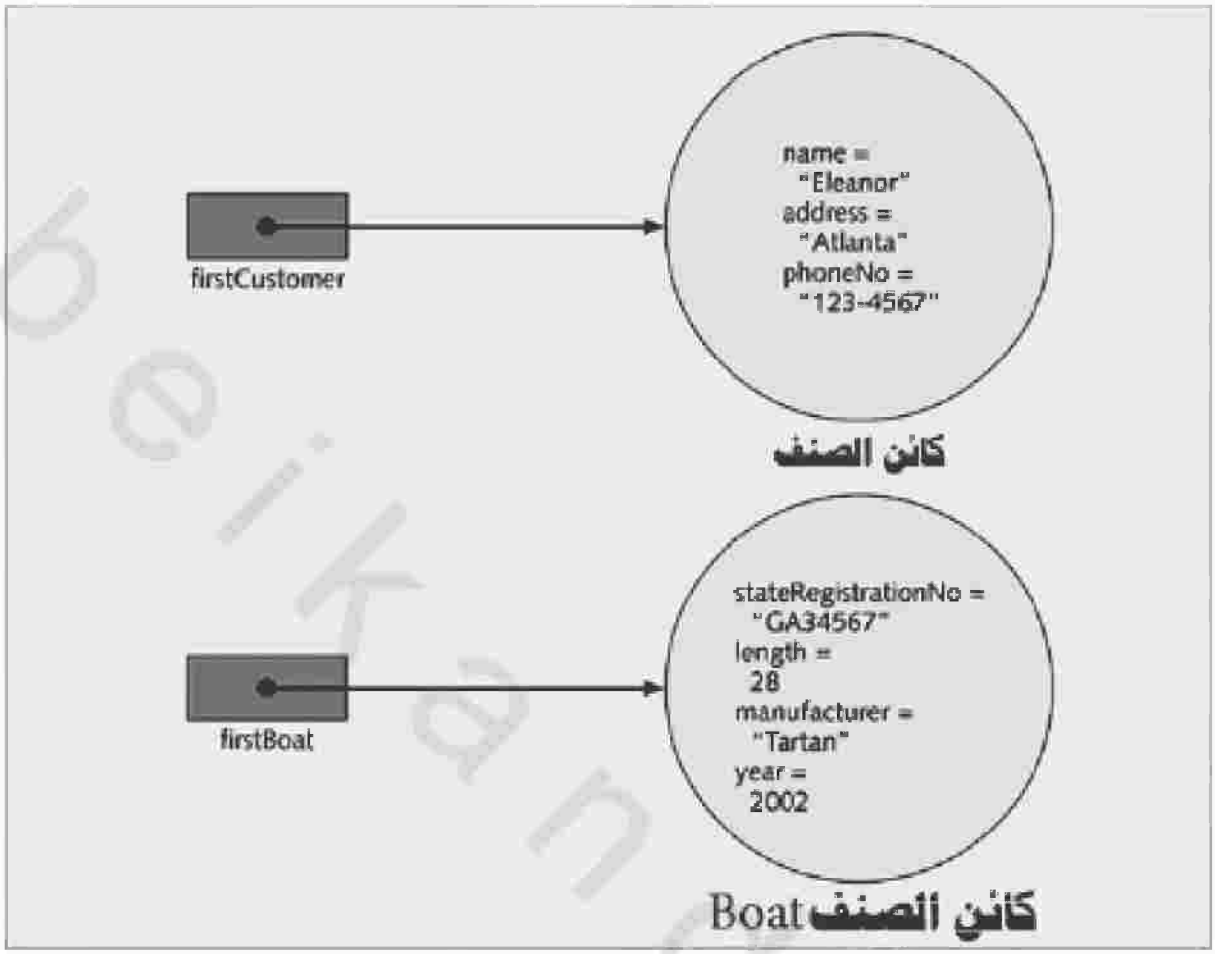
يمثل الصنف Lease صنف ترافق ، حيث ارتبط وجوده بوجود علاقة ترافق بين الصنف Customer والصنف Slip (كما أوضحنا في الفصل الخامس). يتم ربط الصنف Lease بخط متقطع مع علاقة الترافق بين كل من الصنف Customer والصنف Slip. وسوف يطور صنف الترافق بلغة VB .NET مثل أي صنف آخر ثم يضاف إليه علاقة الترافق مع الأصناف الأخرى.

ربط أصناف VB .NET بعلاقة ترافق من النوع واحد إلى واحد
Associating VB .NET Classes in One-to-One Relationship

لقد قدمنا في الفصل السادس الصنف Customer وقدمنا في الفصل الثامن الصنف Boat مستقلاً عن الأصناف الفرعية Sailboat وPowerboat، وكما هو واضح من نموذج Class Diagram فإن هناك علاقة ترافق بين الصنف Customer والصنف Boat، حيث إن العميل يملك مركباً.

عزيزي القارئ لاحظ أن علاقة الترافق بين العميل والمركب، والممثلة في نموذج Class Diagram بخط ، هي علاقة ثنائية الاتجاه. الأول: العميل يملك مركباً، والثاني: المركب مملوك للعميل ؛ ولذلك يجب أن يأخذ مطورو النظام ذلك في الاعتبار، حيث يتم معالجة كل اتجاه على حده. فعلى سبيل المثال، ربما تختلف اتجاهات العلاقة والتي يمكن أن يكون أحدها إجبارياً (Mandatory) والآخر اختيارياً (Optional) ؛ ولذلك إذا رجعنا لعلاقة العميل والمركب نجد أن المركب يجب أن يكون مملوكاً للعميل (علاقة إجبارية)، أما العميل الذي لا يملك مركباً فربما تخزن بياناته داخل النظام (علاقة اختيارية). أما عدد أطراف العلاقة (Multiplicity) بين الطرفين ربما يكون مختلفاً. فعلى سبيل المثال ، الرصيف يحتوي على العديد من المراسي أما المرسى فهو جزء من رصيف واحد.

تطور علاقات الترابط في لغة VB .NET بواسطة تعريف متغير إشارة (Reference Variable) كصفة داخل صنف ما ، وبذلك يستطيع أن يشير إلى كائن من صنف آخر. يوضح الشكل رقم (٩.٢) كيف يشير متغير إشارة إلى كائن من صنف ما، حيث يشير متغير إشارة من النوع Customer إلى كائن من الصنف Customer ويشير متغير إشارة من النوع Boat إلى كائن من الصنف Boat ؛ ولذلك لو تم إضافة متغير إشارة من النوع Customer كصفة داخل الصنف Boat يستطيع كل كائن من الصنف Boat أن يشير إلى كائن من الصنف Customer ومن ثم يستدعي إجراءاته. وأيضاً إذا أضفنا متغير إشارة من الصنف Boat كصفة للصنف Customer يستطيع كل كائن من الصنف Customer أن يشير إلى كائن من الصنف Boat ومن ثم يستدعي إجراءاته. وبهذا نكون قد أنشأنا علاقة ترافق بين الصنف Customer والصنف Boat في كلا الاتجاهين مستخدمين لغة VB .NET.



الشكل رقم (٩,٢). كيف تشير متغيرات الإشارة إلى كائن من صنف ما.

لقد أنشأنا متغيرات إشارة تشير إلى كائنات من أصناف خلال برامج الاختبار السابقة، ثم تستدعي إجراءاتها لاختبار صحة عملها. كما أنشأنا متغيرات إشارة لكلٍّ من الصنف String والصنف DateTime كصفات داخل أصناف نظام شركة برادشو مارينا، حيث أنشأنا متغير إشارة لكائن من الصنف String كصفة لتخزين اسم مصنِّع المركب داخل الصنف Boat. وأنشأنا متغير إشارة لكائن من الصنف DateTime كصفة داخل الصنف Lease لتخزين تاريخ بداية العقد وتاريخ نهايته.

لا يقدم هذا الفصل جديداً عن استخدام متغيرات الإشارة كصفات داخل الأصناف، ولكن الجديد هو استخدام هذه الصفات في تطوير علاقات الترافق بين أصناف مجال المشكلة. وسوف تكون أمثلة هذا الفصل سهلة بالنسبة لك إذا فهمت الفرق بين متغير الإشارة وبين الكائن.

تعديل الصنف Customer
Modifying the Customer Class

تذكر أن الصنف Customer يحتوي على ثلاث صفات وهي: الصفة name لتخزين اسم العميل، والصفة address لتخزين عنوان العميل، والصفة phone لتخزين هاتف العميل. ولتطوير علاقة الترافق واحد إلى واحد مع الصنف Boat، يجب إضافة صفة داخل الصنف Customer التي ستحتوي على متغير إشارة لكائن من الصنف Boat. يوضح الشكل رقم (٩,٣) تعريف الصنف Customer بعد إضافة علاقة الترافق مع الصنف Boat، حيث يحتوي الصنف Customer الآن على أربع صفات.

```
'attributes
Private name As String
Private address As String
Private phoneNo As String
'reference variable for Boat instance
Private theBoat As Boat
```

```
' Customer class from Chapter 6
' Modified to Associate with Boat

Public Class Customer

    'attributes
    Private name As String
    Private address As String
    Private phoneNo As String
    'reference variable for Boat instance
    Private theBoat As Boat

    'constructor (3 parameters)
    Public Sub New(ByVal aName As String, ByVal anAddress As String, _
      ByVal aPhoneNo As String)
        SetName(aName)
        SetAddress(anAddress)
        SetPhoneNo(aPhoneNo)
        SetBoat(Nothing)
    End Sub

    'TellAboutSelf method
    Public Function TellAboutSelf() As String
        Return name & ", " & address & ", " & phoneNo
    End Function

    'get accessor methods (including GetBoat)
    Public Function GetName() As String
        Return name
    End Function
    Public Function GetAddress() As String
        Return address
    End Function
    Public Function GetPhoneNo() As String
        Return phoneNo
    End Function
```

الشكل رقم (٩,٣). تعريف الصنف Customer المعدل.

```
    Public Function GetBoat() As Boat
        Return theBoat
    End Function

    'set accessor methods [including SetBoat]
    Public Sub SetName(ByVal aName As String)
        name = aName
    End Sub
    Public Sub SetAddress(ByVal anAddress As String)
        address = anAddress
    End Sub
    Public Sub SetPhoneNo(ByVal aPhoneNo As String)
        phoneNo = aPhoneNo
    End Sub
    Public Sub SetBoat(ByVal aBoat As Boat)
        theBoat = aBoat
    End Sub

End Class
```

تابع الشكل رقم (٩,٣).

عزيزي القارئ يجب أن تلاحظ أن متغير الإشارة Boat يشير إلى كائن من الصنف Boat ولا يحتوي على قيمه تمثل مفتاح رئيساً (أو رقماً تعريفياً) للمركب، حيث إن هذه القيم تستخدم لتطوير العلاقات بين الجداول في قواعد البيانات العلاقية (Relational Database). ومن ثم يمكن أن يرتبط جدول "أ" بجدول "ب" بإضافة مفتاح أجنبي (Foreign Key) داخل الجدول "أ" الذي يقابل المفتاح الرئيس في الجدول "ب". على الجانب الآخر متغير الإشارة في الصنف Customer يشير مباشرة إلى كائن من النوع Boat.

وكما تعودنا أن إضافة صفة جديدة تتطلب إضافة إجراء وصول Setter لإسناد متغير الإشارة وإجراء وصول Getter لاسترجاع متغير الإشارة. يستقبل إجراء الوصول Setter متغير إشارة لإسناده داخل الصفة theBoat هكذا:

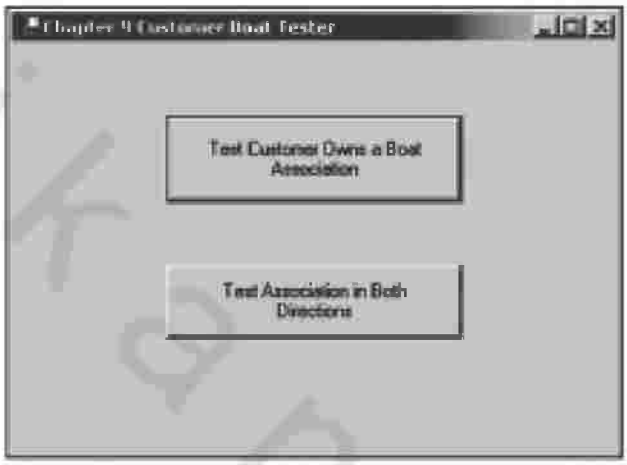
```
Public Sub SetBoat(ByVal aBoat As Boat)
    theBoat = aBoat
End Sub
```

يعيد إجراء الوصول Getter الصفة theBoat هكذا (لاحظ أن نوع البيانات الذي سيرجع هو Boat):

```
Public Function GetBoat() As Boat
    Return theBoat
End Function
```

بداية يسند إجراء الإنشاء للصنف Customer الصفة theBoat بالقيمة Nothing مستخدماً الإجراء SetBoat، وبعد إنشاء كائنات الصنف Customer يمكن إسناد كائن الصنف Boat لاحقاً.

يوضح الشكل رقم (٩.٤) نموذجاً بعنوان "Chapter 9 Customer Boat Tester" والذي يحتوي على زرّين. يختبر إجراء معالجة الزر الأول الصنف Customer بعد تعديله وذلك بإنشاء علاقة ترافق بين الصنف Customer والصنف Boat. لاحظ أن تعريف الصنف Boat لم يتغير إلى الآن ؛ ولذلك لم يطلب من كائن الصنف Boat أن ينشئ العلاقة من طرفه، بل سيتم إنشاؤها من طرف كائن الصنف Customer فقط..يوضح الشكل رقم (٩.٥) محتويات هذا الإجراء.

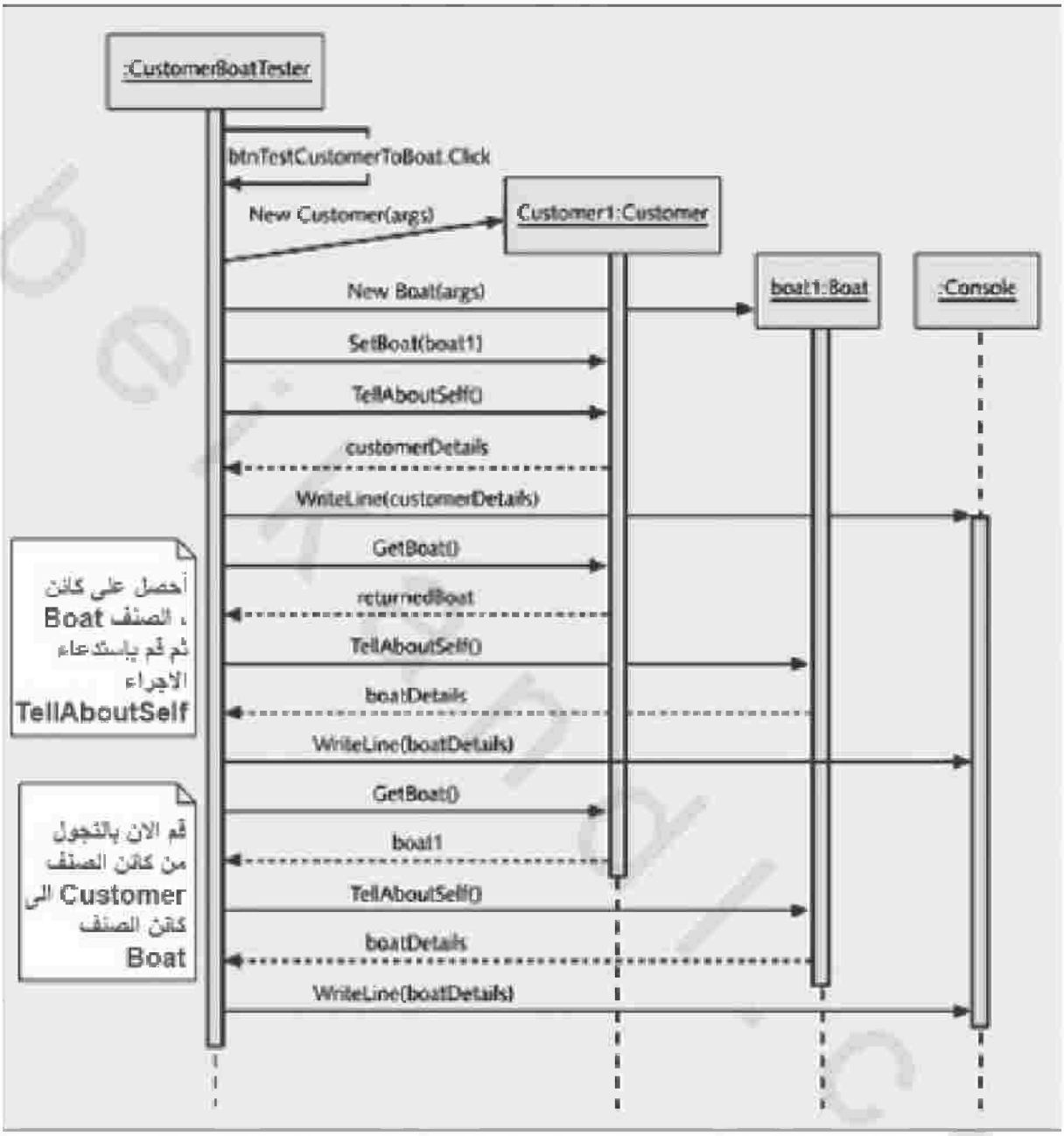


الشكل رقم (٩.٤). نموذج اختبار علاقة الصنف Customer بالصنف Boat.

```
Private Sub btnTestCustomerToBoat_Click(ByVal sender As _
System.Object, ByVal e As System.EventArgs) Handles _
btnTestCustomerToBoat.Click

    'create customer instance
    Dim customer1 As New Customer("Eleanor", "Atlanta", "123-4567")
    'create boat instance
    Dim boat1 As New Boat("GA34567", 28, "Tartan", 2002)
    'ask customer1 to set its boat reference to boat1
    customer1.SetBoat(boat1)
    'get customer information
    Console.WriteLine("Customer is " & customer1.TellAboutSelf())
    'get boat reference from customer
    Dim returnedBoat As Boat = customer1.GetBoat()
    'get the information on the boat
    Console.WriteLine("Boat info is " _
     & returnedBoat.TellAboutSelf())
    'navigate directly from customer to boat
    Console.WriteLine("Again boat info is " _
     & customer1.GetBoat().TellAboutSelf())
```

الشكل رقم (٩.٥). شفرة إجراء اختبار علاقة الصنف Customer بالصنف Boat.

يوضح الشكل رقم (٩.٦) جزءاً من نموذج التفاعل بين الكائنات الناتج من الإجراء السابق، كما يوضح الشكل رقم (٩.٧) مخرجات هذا الإجراء.



الشكل رقم (٩,٦). نموذج Sequence Diagram يوضح تفاعل الصنف Customer مع الصنف Boat.

```
Customer is Eleanor, Atlanta, 123-4567
Boat info is GA34567, 28, Tartan, 2002
Again boat info is GA34567, 28, Tartan, 2002
```

الشكل رقم (٩,٧). مخرجات إجراء اختبار علاقة الصنف Customer بالصنف Boat.

أولاً، سيتم إنشاء كائن من الصنف Customer وكائن من الصنف Boat هكذا:

```
'create customer instance
Dim customer1 As New Customer("Eleanor", "Atlanta", _ "123-4567")
'create boat instance
Dim boat1 As New Boat("GA34567", 28, "Tartan", 2002)
```

ثم يستدعى الإجراء SetBoat من كائن الصنف Customer ممرراً إليه متغير الإشارة boat1 لإقامة العلاقة بينها من طرف كائن الصنف Customer هكذا:

```
'ask customer1 to set its boat reference to boat1
customer1.SetBoat(boat1)
```

ثم يستدعى الإجراء TellAboutSelf من كائن الصنف Customer للتحقق من صحة بيانات العميل. والآن يمكن استرجاع متغير الإشارة الذي يشير إلى كائن الصنف Boat من كائن الصنف Customer ومن ثم استدعاء الإجراء TellAboutSelf من كائن الصنف Boatلعرض بيانات المركب المملوك لهذا العميل هكذا:

```
'get boat reference from customer
Dim returnedBoat As Boat = customer1.GetBoat()
'get the information on the boat
Console.WriteLine("Boat info is " _
 & returnedBoat.TellAboutSelf())
```

إن إمكانية التجول من كائن لكائن من خلال علاقة الترافق التي تربط بينهما تعد من أحسن إمكانيات البرمجة المعتمدة على الكائنات حيث يمكن استدعاء سلسلة من الإجراءات في أمر واحد هكذا:

```
customer1.GetBoat().GetStateRegistrationNo()
```

يجب قراءة الأمر السابق من اليسار إلى اليمين حيث يستقبل الكائن Customer1 رسالة GetBoat التي تعيد متغير إشارة لكائن الصنف Boat. عندئذ يستبدل هذا المتغير الجزء الأيسر من الأمر، وبذلك يستقبل رسالة GetStateRegistrationNo التي تعيد رقم تسجيل حالة المركب على شكل نص (كائن من الصنف String). إن أي إجراء معرف داخل الصنف Boat يمكن استدعاؤه بالتجول من كائن الصنف Customer إلى كائن من الصنف Boat وذلك بفضل وجود علاقة ترافق

بينهما. ومن ثم يمكن معرفة معلومات المركب الذي يملكه عميل ما عن طريق التجول من كائن الصنف Customer إلى كائن الصنف Boat ثم استدعاء الإجراء TellAboutSelf المعرف لدى الصنف Boat هكذا:

```
'navigate directly from customer to boat
Console.WriteLine("Again boat info is " _
 & customer1.GetBoat().TellAboutSelf())
```

تعديل الصنف Boat
Modifying the Boat Class

لم نعدل حتى الآن تعريف الصنف Boat لتطوير علاقة الترافق بينه وبين الصنف Customer؛ ولذلك توجد الآن علاقة الترافق في اتجاه واحد، حيث لا يعلم كائن الصنف Boat أي معلومات عن كائن الصنف Customer الذي يملكه. ويمكن تعديل تعريف الصنف Boat لإقامة علاقة الترافق في الاتجاه الآخر وذلك بإضافة متغير إشارة للصنف Customer كصفة لدى الصنف Boat، ثم تعريف إجراءات الوصول لإسناد/استرجاع هذه الصفة. وبهذا نكون قد أقمنا علاقة الترافق في الاتجاه الآخر (المركب مملوك للعميل). يوضح الشكل رقم (٩.٨) تعريف الصنف Boat بعد التعديل.

يوضح الشكل رقم (٩.٩) أوامر إجراء اختبار العلاقة في كلا الاتجاهين، كما يوضح الشكل رقم (٩.١٠) مخرجات هذا الإجراء.

يحتوي إجراء الاختبار على كائنين من النوع Customer وكائنين من النوع Boat، ثم أقيمت العلاقة بينها في كلا الاتجاهين هكذا:

```
'establish association for first customer and boat
customer1.SetBoat(boat1)
boat1.SetCustomer(customer1)
'establish association for second customer and boat
customer2.SetBoat(boat2)
boat2.SetCustomer(customer2)
```

ثم طلبنا من كائنات الصنف Customer معلومات عن المراكب التي تمتلكها وذلك بالتجول من كائن إلى آخر. وبالمثل طلبنا من كائنات الصنف Boat معلومات عن العملاء الذي تمتلكها وذلك بالتجول المباشر من كائن إلى آخر. على سبيل المثال توضح الأوامر التالية ذلك مع كل من كائن الصنف Customer وكائن الصنف Boat:

```
'navigate directly from customer1 to boat1
Console.WriteLine("Boat info is " _
& customer1.GetBoat().TellAboutSelf())
'navigate directly from to boat1 to customer1
Console.WriteLine("Customer info is " _
& boat1.GetCustomer().TellAboutSelf())
```

```
' Boat class from Chapter 8 used as concrete class
' Modified to associate with Customer

Public Class Boat

    'attributes
    Private stateRegistrationNo As String
    Private length As Single
    Private manufacturer As String
    Private year As Integer
    'reference variable for Customer instance
    Private theCustomer As Customer

    'constructor (four parameters)
    Public Sub New(ByVal sStateRegistrationNo As String, _
       ByVal sLength As Single, _
       ByVal sManufacturer As String, ByVal sYear As Integer)
       'set values of attributes using accessor methods
       SetStateRegistrationNo(sStateRegistrationNo)
       SetLength(sLength)
       SetManufacturer(sManufacturer)
       SetYear(sYear)
       SetCustomer(Nothing)
    End Sub

    'TellAboutSelf method
    Public Overridable Function TellAboutSelf() As String
       Return (stateRegistrationNo & ", " & length & ", " & _
         manufacturer & ", " & year)
    End Function

    'Get accessor methods (including GetCustomer)
    Public Function GetStateRegistrationNo() As String
       Return stateRegistrationNo
    End Function
    Public Function GetLength() As Single
       Return length
    End Function
    Public Function GetManufacturer() As String
       Return manufacturer
    End Function
    Public Function GetYear() As Integer
       Return year
    End Function
    Public Function GetCustomer() As Customer
       Return theCustomer
    End Function
   'Set accessor methods (including SetCustomer)
   Public Sub SetStateRegistrationNo(ByVal sStateRegistrationNo _
      As String)
      stateRegistrationNo = sStateRegistrationNo
   End Sub
   Public Sub SetLength(ByVal sLength As Single)
      length = sLength
   End Sub
   Public Sub SetManufacturer(ByVal sManufacturer As String)
      manufacturer = sManufacturer
   End Sub
   Public Sub SetYear(ByVal sYear As Integer)
      year = sYear
   End Sub
   Public Sub SetCustomer(ByVal sCustomer As Customer)
      theCustomer = sCustomer
   End Sub

End Class
```

الشكل رقم (٩,٨). تعريف الصنف Boat المعدل.

```
Private Sub btnTestBothDirections_Click(ByVal sender As _
   System.Object, ByVal e As System.EventArgs) Handles _
   btnTestBothDirections.Click

      'create two customer instances
      Dim customer1 As New Customer("Eleanor", "Atlanta", "123-4567")
      Dim customer2 As New Customer("JoAnn", "St Louis", "987-6543")
      'create two boat instances
      Dim boat1 As New Boat("GA34567", 28, "Tartan", 2002)
      Dim boat2 As New Boat("MO12345", 32, "Catalina", 2003)
      'establish association for first customer and boat
      customer1.SetBoat(boat1)
      boat1.SetCustomer(customer1)
      'establish association for second customer and boat
      customer2.SetBoat(boat2)
      boat2.SetCustomer(customer2)

      'navigate directly from customer1 to boat1
      Console.WriteLine("Boat info is " _
      & customer1.GetBoat().TellAboutSelf())
      'navigate directly from boat1 to customer1
      Console.WriteLine("Customer info is " _
      & boat1.GetCustomer().TellAboutSelf())

      'do the same for second customer
      Console.WriteLine("Boat info is " _
      & customer2.GetBoat().TellAboutSelf())
      'navigate directly from boat2 to customer2
      Console.WriteLine("Customer info is " _
      & boat2.GetCustomer().TellAboutSelf())

End Sub
```

الشكل رقم (٩,٩). شفرة إجراء اختبار علاقة الصنف Customer بالصنف Boat في الاتجاهين.

```
Boat info is GA34567, 28, Tartan, 2002
Customer info is Eleanor, Atlanta, 123-4567
Boat info is MO12345, 32, Catalina, 2003
Customer info is JoAnn, St Louis, 987-6543
```

الشكل رقم (٩,١٠). مخرجات إجراء اختبار علاقة الصنف Customer بالصنف Boat في الاتجاهين.

إضافة مهام أخرى للصنف Boat

Adding Functionality to the Boat Class

لقد عرفنا خلال الفقرة السابقة كيف نطور علاقات الترافق بين كل من الصنف Customer والصنف Boat في كلا الاتجاهين. كما يمكن إضافة بعض المهام الهامة الأخرى إلى الأصناف التي يوجد بديلها علاقات ترافق لجعلها أكثر إفادة وتطبيقاً. فعلى سبيل المثال، يمكن إضافة إجراء خاص داخل الصنف Boat لإقامة علاقة الترافق بين الصنف Boat والصنف Customer في كلا الاتجاهين وذلك في خطوة واحدة. ولأن شركة برادشو ماربنا لا تريد أن تحتفظ ببيانات المركب إلا إذا كان مملوكاً بالفعل للعميل، إذن يمكن جعل علاقة الترافق بين الصنف Boat والصنف Customer من النوع الإجباري (Mandatory) وذلك بإضافة متغير إشارة لكائن من الصنف Customer في إجراء إنشاء الصنف Boat. ومن ثم لا يمكن أن ننشئ كائناً من الصنف Boat إلا إذا كان يوجد بالفعل كائن من الصنف Customer. وأخيراً، يمكن تعديل الإجراء TellAboutSelf المعرف لدى الصنف Boat لكي نعيد كلاً من معلومات المركب والعميل معاً حيث لا يوجد مركب ليس مملوكاً للعميل. سوف نصف هذه التحسينات من خلال المشروع Example2 التالي. يوضح الشكل رقم (٩.١١) تعريف الصنف Boat بعد إضافة هذه الإمكانات إليه.

إن الإجراء الخاص AssignBoatToCustomer، الذي أضيف إلى الصنف Boat، يقيم العلاقة بين كلا الصنفين في خطوة واحدة، حيث يستدعي الإجراء SetCustomer لإقامة العلاقة من جهة المركب. ثم يستدعي الإجراء SetBoat من كائن الصنف Customer طالباً منه إقامة العلاقة من جهته (لاحظ استخدام الكلمة Me والتي تعني كائن الصنف Boat الحالي). وبمعنى آخر، إن كائن الصنف Boat يرسل رسالة إلى كائن الصنف Customer لإقامة علاقة ترافق بينهما من جهة العميل. توضح الأوامر التالية تعريف الإجراء AssignBoatToCustomer:

```
'custom method to establish associations in both directions
Public Sub AssignBoatToCustomer(ByVal aCustomer As Customer)
    SetCustomer(aCustomer)      'set customer reference for boat
    aCustomer.SetBoat(Me)       'ask the customer to set its boat
End Sub
```

يمكن استدعاء هذا الإجراء من برنامج الاختبار مستخدماً الأوامر التالية:

```
boat1.AssignBoatToCustomer(customer1)
```

```
' Boat class with additional capabilities

Public Class Boat

    'attributes
    Private stateRegistrationNo As String
    Private length As Single
    Private manufacturer As String
    Private year As Integer

    'reference variable for Customer instance
    Private theCustomer As Customer

    'constructor (four parameters plus customer reference)
    Public Sub New(ByVal aStateRegistrationNo As String, _
       ByVal aLength As Single, _
       ByVal aManufacturer As String, ByVal aYear As Integer, _
       ByVal aCustomer As Customer)
       'set values of attributes using accessor methods
       SetStateRegistrationNo(aStateRegistrationNo)
       SetLength(aLength)
       SetManufacturer(aManufacturer)
       SetYear(aYear)
       'because boat must have a customer, assign it in constructor
       AssignBoatToCustomer(aCustomer)
    End Sub

    'custom method to establish associations in both directions
    Public Sub AssignBoatToCustomer(ByVal aCustomer As Customer)
       SetCustomer(aCustomer)      'set customer reference for boat
       aCustomer.SetBoat(Me)       'ask customer to set its boat
    End Sub

    'TellAboutSelf method that gets customer info also
    Public Overridable Function TellAboutSelf() As String
       Return "boat number " & stateRegistrationNo & ", " _
       & length & ", " & manufacturer & ", " & year _
       & " customer for boat is " & theCustomer.TellAboutSelf()
    End Function

    (Continue with Get and Set accessor methods)
    . . .
    . . .

End Class
```

الشكل رقم (٩,١١). تعريف الصنف Boat بعد الإمكانيات الإضافية.

عند استدعاء الإجراء AssignBoatToCustomer، يستدعي كائن الصنف Boat الإجراء SetCustomer (ممرراً إليه متغير إشارة لكائن الصنف Customer) المعرف داخله. ثم يستدعي الإجراء SetBoat المعرف لدى الصنف Customer ممرراً إليه متغير إشارة لكائن الصنف Boat (الكلمة Me). لاحظ أن الإجراء AssignBoatToCustomer يمكن أن يستخدم لتعديل علاقة ترافق ما، حيث يمكن استبدال المالك القديم للمركب بمالك جديد من خلال تعديل العلاقة في الاتجاهين وفي خطوة واحدة.

حدث تحسين آخر على الصنف Boat وهو أن شركة برادشو مارينا لا تريد أن تحتفظ بمعلومات مركب لا يملكه عميل. ومن ثم تم تعديل إجراء إنشاء الصنف Boat ليستقبل متغير إشارة لكائن الصنف Customer الذي سيملك كائن الصنف Boat المراد إنشاؤه. وبهذا تصبح العلاقة بين الصنف Customer والصنف Boat علاقة إجبارية بدلاً من أن تكون علاقة اختيارية، حيث لا تستطيع إنشاء كائن من الصنف Boat إلا إذا مررت كائن من الصنف Customer إلى إجراء إنشائه. سيصبح رأس إجراء الإنشاء الخاص بالصنف Boat هكذا:

```
'constructor (four parameters plus customer reference)
Public Sub New(ByVal aStateRegistrationNo As String, _
    ByVal aLength As Single, _
    ByVal aManufacturer As String, ByVal aYear As Integer, _
    ByVal aCustomer As Customer)
```

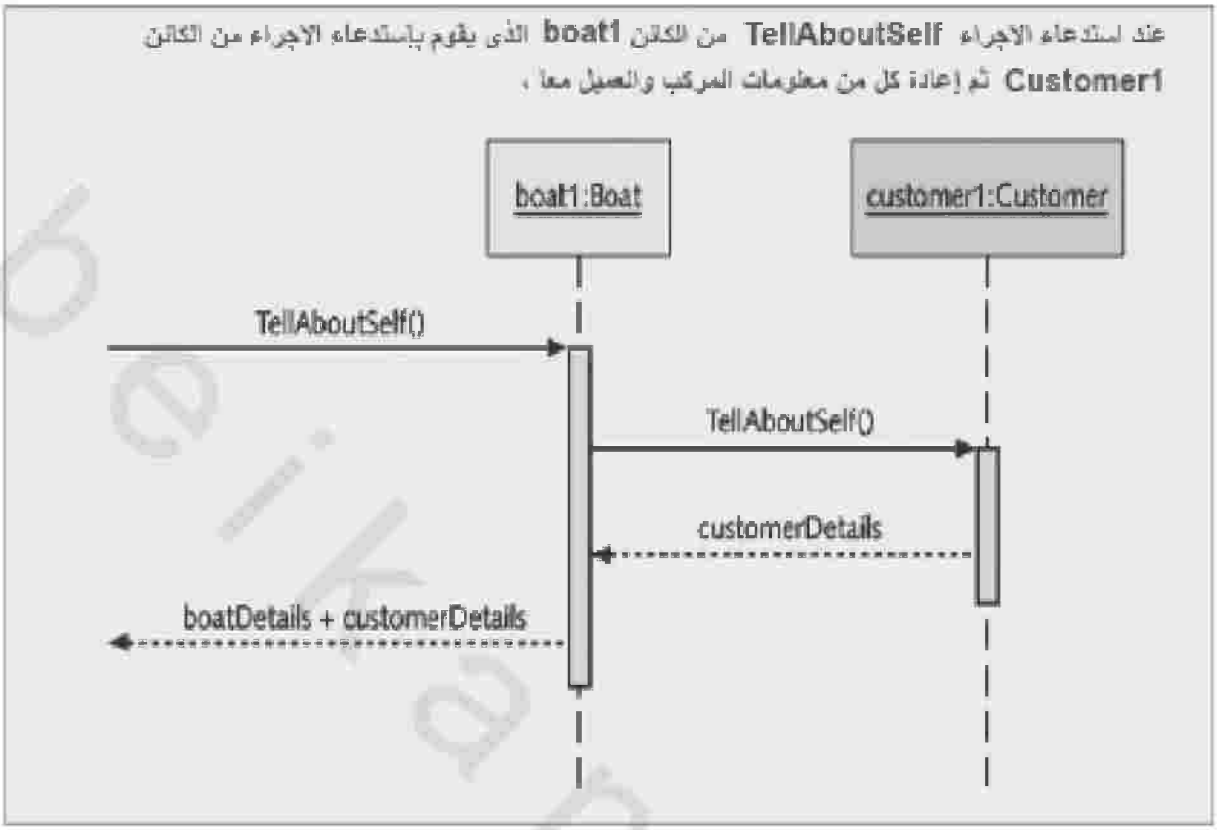
يسند إجراء الإنشاء جميع صفات كائن الصنف Boat ثم يستدعي الإجراء AssignBoatToCustomer لكي يسند صفة العميل لدى المركب ثم يسند صفة المركب لدى العميل؛ وبذلك، عندما ينشئ كائن من الصنف Boat ينشئ بشكل تلقائي علاقة الترافق بينه وبين كائن الصنف Customer في كلا الاتجاهين:

```
'set values of attributes using accessor methods
SetStateRegistrationNo(aStateRegistrationNo)
SetLength(aLength)
SetManufacturer(aManufacturer)
SetYear(aYear)
'because boat must have a customer, assign it in 'constructor
AssignBoatToCustomer(aCustomer)
```

وأخيراً، حدث تعديل على إجراء TellAboutSelf المعرف لدى الصنف Boat، حيث أصبح يعيد كلاً من معلومات المركب ومعلومات العميل الذي يملكه وذلك لأن العلاقة بين الصنف Boat وبين الصنف Customer أصبحت علاقة إجبارية. بمعنى أنه لا يوجد مركب (كائن من الصنف Boat) ليس له مالك (كائن من الصنف Customer). سيستدعي الإجراء TellAboutSelf المعروف لدى الصنف Boat الإجراء TellAboutSelf المعروف لدى الصنف Customer ليستقبل معلومات العميل داخل نص (String). ثم يعيد نصاً يحتوي على النص العائد من العميل ومعلومات المركب هكذا:

```
'TellAboutSelf method that gets customer info also
Public Overridable Function TellAboutSelf() As String
    Return "boat number " & stateRegistrationNo & ", " _
    & length & ", " & manufacturer & ", " & year _
    &"customer for boat is" & theCustomer.TellAboutSelf()
End Function
```

يوضح الشكل رقم (٩.١٢) جزءاً من نموذج Sequence Diagram الذي يمثل التفاعل بين الكائنات عند استدعاء الإجراء TellAboutSelf من أحد كائنات الصنف Boat.



الشكل رقم (٩,١٢). نموذج Sequence Diagram يوضح إجراء TellAboutSelf الخاص بالصنف Boat.

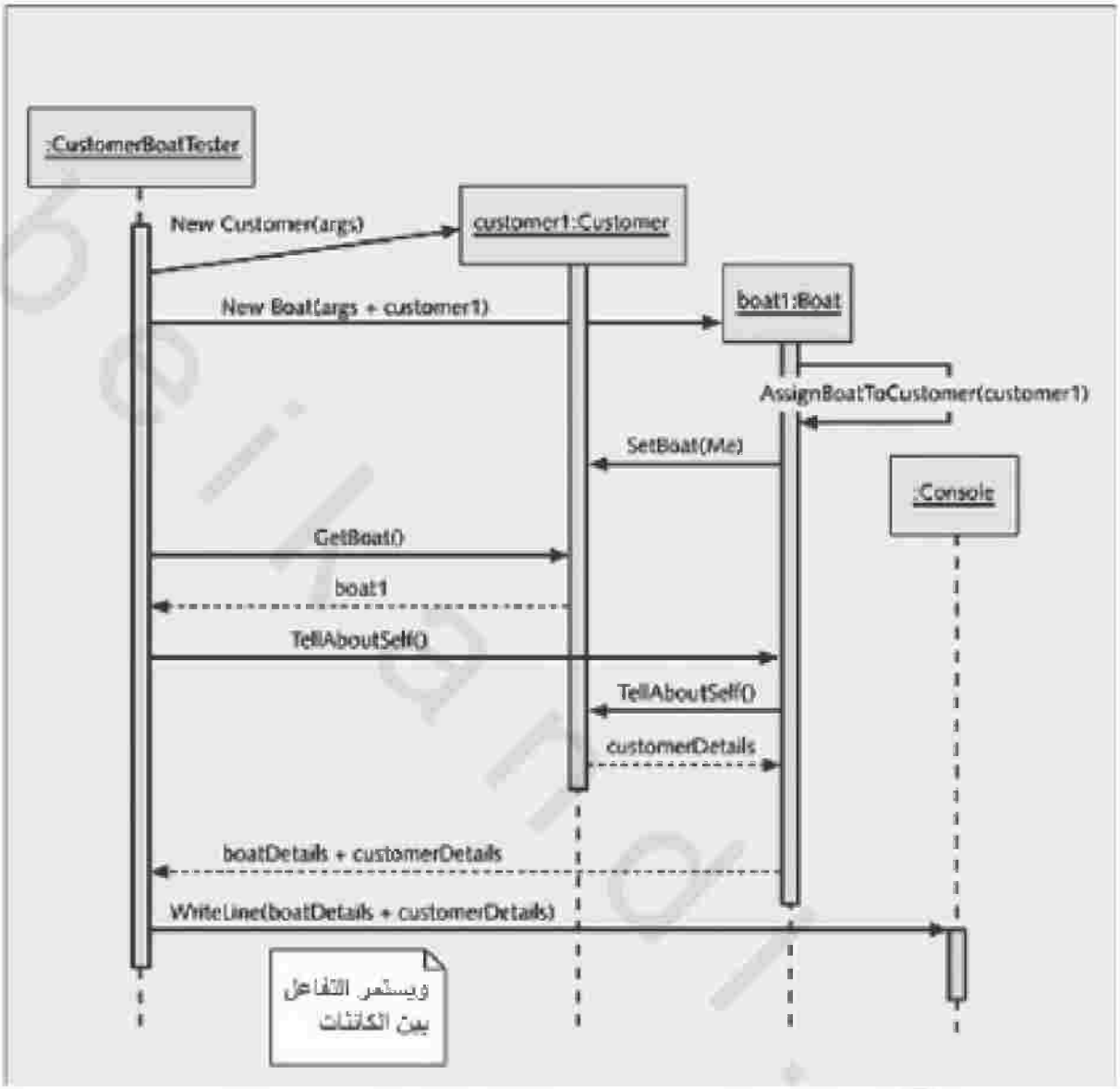
يوضح الشكل رقم (٩,١٣) إجراء اختبار قدرات الصنف Boat الجديد. كما يوضح الشكل رقم (٩,١٤) مخرجات هذا الإجراء. ومن الملاحظ أن أوامر هذا الإجراء بسيطة لأن إجراء إنشاء الصنف Boat يتطلب متغير إشارة لكائن الصنف Customer ثم تنشأ علاقة الترافق بين الصنف Boat والصنف Customer بواسطة استدعاء الإجراء TellAboutSelf داخل إجراء الإنشاء. وما يزيد الأمر بساطة هو أن الإجراء TellAboutSelf المعرف لدى الصنف Boat يعيد كلاً من معلومات المركب ومعلومات العميل. يعرض الشكل رقم (٩,١٥) نموذج Sequence Diagram الذي يوضح التفاعل بين الكائنات لتنفيذ هذا الإجراء. لاحظ أن هذا الإجراء يختبر علاقة الترافق في كلا الاتجاهين. ولاحظ أيضاً أن الإجراء TellAboutSelf المعرف لدى الصنف Customer لا يعيد معلومات المركب وذلك لأنه ليس كل عميل يملك مركباً ولكن كل مركب يملكه عميل.

```
Private Sub btnTestBothDirections_Click(ByVal sender _
As System.Object, ByVal e As System.EventArgs) _
Handles btnTestBothDirections.Click

    'create two customer instances
    Dim customer1 As New Customer("Eleanor", "Atlanta", "123-4567")
    Dim customer2 As New Customer("JoAnn", "St Louis", "987-6543")

    'create two boat instances passing customer references
    Dim boat1 As New Boat("GA34567", 28, "Tartan", 2002, customer1)
    Dim boat2 As New Boat("MO12345", 32, "Catalina", 2003, customer2)

    'navigate directly from customer1 to boat1
    Console.WriteLine("Boat info is " _
    & customer1.GetBoat().TellAboutSelf())
    'navigate directly from boat1 to customer1
    Console.WriteLine("Customer info is " _
    & boat1.GetCustomer().TellAboutSelf())

    'do the same for second customer
    Console.WriteLine("Boat info is " _
    & customer2.GetBoat().TellAboutSelf())

    'navigate directly from boat2 to customer2
    Console.WriteLine("Customer info is " _
    & boat2.GetCustomer().TellAboutSelf())

End Sub
```

الشكل رقم (٩,١٣). شفرة إجراء اختبار الصنف Boat بعد الإمكانيات الإضافية.

```
Boat info is boat number GA34567, 28, Tartan, 2002 customer for boat is Eleanor, Atlanta, 123-4567
Customer info is Eleanor, Atlanta, 123-4567
Boat info is boat number MO12345, 32, Catalina, 2003 customer for boat is JoAnn, St Louis, 987-6543
Customer info is JoAnn, St Louis, 987-6543
```

الشكل رقم (٩,١٤). مخرجات إجراء اختبار الصنف Boat بعد الإمكانيات الإضافية.



الشكل رقم (٩,١٥). نموذج Sequence Diagram لإجراء اختبار الصنف Customer والصنف Boat.

إنشاء علاقة ترافق واحد إلى مجموعة بين الصنف Dock والصنف Slip

Associating Dock and Slips: A One-to-Many Association Relationship

لقد قدمنا الصنف Slip (المرسى) خلال الفصل السابع. هذا الصنف تربطه علاقة ترافق بالصنف Dock (رصيف). كما واضح في الشكل رقم (٩,١) تتكون هذه العلاقة من اتجاهين وهما: يلحق المرسى برصيف ويتكون الرصيف من عدة مراسي. ينتمي الشق الأول من العلاقة (من المرسى إلى الرصيف) إلى النوع واحد إلى واحد (One-to-One) والتي تتشابه مع نفس العلاقة التي تربط العميل بالمركب، وبذلك يتم تطويرها بنفس

الأسلوب المتبع في الفقرة السابقة. ولكن الشق الآخر من العلاقة (من الرصيف إلى المرسى) ينتمي إلى النوع واحد إلى مجموعة (One-to-Many) والتي تدون بالرمز *..١ على نموذج Class Diagram. ويتطلب هذا الشق من العلاقة أسلوباً آخر لتطويره.

يتطلب تطوير هذا الشق من العلاقة تعريف العديد من متغيرات الإشارة لكائنات الصنف Slip داخل الصنف Dock، حيث يمكن استخدام الصنف ArrayList (قدمناه في الفصل الرابع) لإنشاء حاوية تخزين متغيرات إشارة لكائنات الصنف Slip.

تقديم الصنف Dock
Introducing the Dock Class

يوضح الشكل رقم (٩.١٦) التعريف الأول للصنف Dock الذي يحتوي على أربع صفات وهي: الصفة dockId (رقم تعريف الرصيف)، والصفة location (موقع الرصيف)، والصفة electricity من النوع Boolean (تعبر عن وجود كهرباء أم لا)، والصفة water من النوع Boolean (تعبر عن وجود مياه أم لا).

ولتطوير علاقة الترافق واحد إلى مجموعة نضيف صفة خاصة باسم slips من النوع ArrayList هكذا:

```
'references to slips associated with dock (1 to many)
Private slips As ArrayList
```

يسند إجراء الإنشاء قيم الصفات الأربعة الأولى، ثم يسند كائن جديد من النوع ArrayList إلى الصفة slips هكذا:

```
slips = New ArrayList()
```

سوف نضيف إجراء وصول Getter لإعادة الصفة slips عند استدعائه مثل أي إجراء Getter آخر (لاحظ نوع البيانات العائد من هذا الإجراء) هكذا:

```
'return the ArrayList containing slip references
Public Function GetSlips() As ArrayList
    Return slips
End Function
```

```
' Dock class (new to Chapter 9)
' each dock has many slips

Public Class Dock

    'attributes
    Private dockId As String
    Private location As String
    Private electricity As Boolean
    Private water As Boolean

    'references to slips associated with dock (1 to many)
    Private slips As ArrayList

    'constructor (four parameters)
    Public Sub New(ByVal aDockID As String, _
    ByVal aLocation As String, _
    ByVal anElectricity As Boolean, ByVal aWater As Boolean)
        'invoke setter methods to populate attributes
        SetDockId(aDockID)
        SetLocation(aLocation)
        SetElectricity(anElectricity)
        SetWater(aWater)
        slips = New ArrayList()
    End Sub

  'add the slip reference to the ArrayList instance
  Public Sub AddSlip(ByVal aSlip As Slip)
     slips.Add(aSlip)
  End Sub

  'return the ArrayList containing slip references
  Public Function GetSlips() As ArrayList
     Return slips
  End Function

  'TellAboutSelf method
  Public Function TellAboutSelf() As String
     Dim dockDetails As String
     dockDetails = "Dock " & dockId _
      & " location " & location & " has electricity " _
      & electricity & " has water " & water
     Return dockDetails
  End Function

  'get accessor methods
  Public Function GetDockId() As String
     Return dockId
  End Function
  Public Function GetLocation() As String
     Return location
  End Function
  Public Function GetElectricity() As Boolean
     Return electricity
  End Function
  Public Function GetWater() As Boolean
     Return water
  End Function

  'set accessor methods
  Public Sub SetDockId(ByVal aDockId As String)
     dockId = aDockId
  End Sub
  Public Sub SetLocation(ByVal aLocation As String)
     location = aLocation
  End Sub
  Public Sub SetElectricity(ByVal anElectricity As Boolean)
     electricity = anElectricity
  End Sub
  Public Sub SetWater(ByVal aWater As Boolean)
     water = aWater
  End Sub

End Class
```

الشكل رقم (٩,١٦)- تعريف الصنف Dock.

ولإضافة مرسى (كائن من الصنف Slip) إلى رصيف (كائن من الصنف Dock)، يجب تعريف إجراء خاص يسمي AddSlip والذي سيتم استدعاؤه بواسطة الإجراء AddSlipToDock المعرف لدى الصنف Slip وذلك لإضافة كائن من الصنف Slip إلى كائن من الصنف Dock، كما هو واضح في الفقرة التالية. يحتوي الإجراء AddSlip على الأمر

slips.Add(aSlip) الذي يضيف متغير إشارة إلى كائن من الصنف Slip (aSlip) إلى كائن من الصنف ArrayList (المشار إليه بالمتغير slips) والمعرف لدى الصنف Dock. ومن ثم سيحتوي كائن الصنف Dock على العديد من كائنات الصنف Slip.

```
'add the slip reference to the ArrayList instance
Public Sub AddSlip(ByVal aSlip As Slip)
    slips.Add(aSlip)
End Sub
```

ربط الصنف Slip بالصنف Dock

Associating the Slip Class with Dock

قدمنا الصنف Slip خلال الفصل السابق، ثم أدخلنا عليه العديد من التحسينات مثل احتوائه على ثوابت (Constants) واستثناءات (Exceptions) لاختبار صحة البيانات. ولذلك سوف نستخدم نسخة مبسطة من الصنف Slip في هذا المثال. يوضح الشكل رقم (٩،١٧) تعريف الصنف Slip الأولي الذي سنربطه بالصنف Dock.

أصبح الصنف Boat بسيطاً مثل الصنف Dock. ولتطوير علاقة ترابط إجبارية (واحد إلى واحد) بينه وبين الصنف Dock، سنعرّف أولاً متغير إشارة إلى كائن من الصنف Dock كصفة في الصنف Slip، ثم نضيف إجراءي وصول لإسناد هذه الصفة داخل الصنف Slip واسترجاعها. بعد ذلك تعدل إجراء إنشاء الصنف Slip لكي يتوقع متغير إشارة لكائن من الصنف Dock. ومن ثم، عند إنشاء كائن من الصنف Slip، يجب ربطه بكائن من الصنف Dock (الذي يجب أن يكون موجوداً مسبقاً). كما يحتوي إجراء الإنشاء على أمر استدعاء الإجراء AddSlipToDock المعرف لدى الصنف Slip، والذي يقيم العلاقة في كلا الاتجاهين وذلك بإسناد الصفة theDock واستدعاء الإجراء AddSlip (المعرف لدى الصنف Dock) ممرراً إليه الكلمة Me هكذا:

```
'assign slip to dock (both directions of association)
Public Sub AddSlipToDock(ByVal aDock As Dock)
    theDock = aDock       ' assign dock reference to attribute
    aDock.AddSlip(Me)     ' ask dock to add Me as slip
End Sub
```

كما يحتوي الصنف Slip على الإجراء TellAboutSelf الذي يعيد معلومات حول كل من المرسى والرصيف المرتبط به هكذا:

```
'custom TellAboutSelf gets slip and dock information
Public Function TellAboutSelf() As String
    Dim slipDetails As String
    slipDetails = "Slip " & slipId _
     & " width " & slipWidth & " length " & slipLength _
     & " attached to " & theDock.TellAboutSelf()
    Return slipDetails
End Function
```

```
' Slip class -
' Modified from Chapter 7 to associate with Dock

Public Class Slip

    'attributes
    Private slipId As Integer
    Private slipWidth As Integer
    Private slipLength As Integer

    'reference variable points to one dock
    Private theDock As Dock

    'constructor (three parameters plus a Dock reference)
    Public Sub New(ByVal aSlipId As Integer, _
    ByVal aSlipWidth As Integer, ByVal aSlipLength As Integer, _
    ByVal aDock As Dock)
        'invoke setter methods to populate attributes
        SetSlipId(aSlipId)
        SetSlipWidth(aSlipWidth)
        SetSlipLength(aSlipLength)
        'invoke custom method for Dock association
        AddSlipToDock(aDock)
    End Sub

    'assign slip to dock (both directions of association)
    Public Sub AddSlipToDock(ByVal aDock As Dock)
        theDock = aDock       ' assign dock reference to attribute
        aDock.AddSlip(Me)     ' ask dock to add Me as slip
    End Sub

    'custom TellAboutSelf gets slip and dock information
    Public Function TellAboutSelf() As String
        Dim slipDetails As String
        slipDetails = "Slip " & slipId _
         & " width " & slipWidth & " length " & slipLength _
         & " attached to " & theDock.TellAboutSelf()
        Return slipDetails
    End Function

    'get accessor methods
    Public Function GetSlipId() As Integer
        Return slipId
    End Function
    Public Function GetSlipWidth() As Integer
        Return slipWidth
    End Function
    Public Function GetSlipLength() As Integer
        Return slipLength
    End Function
    Public Function GetDock() As Dock
        Return theDock
    End Function

    'set accessor methods
    Public Sub SetSlipId(ByVal aSlipId As Integer)
        slipId = aSlipId
    End Sub
    Public Sub SetSlipWidth(ByVal aSlipWidth As Integer)
        slipWidth = aSlipWidth
    End Sub
    Public Sub SetSlipLength(ByVal aSlipLength As Integer)
        slipLength = aSlipLength
    End Sub

End Class
```

الشكل رقم (٩,١٧). تعريف الصنف Slip المبدلي.

اختبار علاقة الترابط بين الصنف Dock والصنف Slip

Testing the Dock and Slip Association Relationship

يوضح الشكل رقم (٩.١٨) إجراء اختبار العلاقة بين الصنف Dock والصنف Slip، كما يوضح الشكل رقم (٩.١٩) مخرجات هذا الإجراء.

```
    Private Sub btnDockHasSlips_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnDockHasSlips.Click

        'declare dock and slip references
        Dim dock1 As Dock
        Dim slip1 As Slip
        Dim slip2 As Slip
        Dim slip3 As Slip

        'create the dock
        dock1 = New Dock(1, "Main Cove", True, False)

        'create three slip instances for the dock
        slip1 = New Slip(1, 10, 20, dock1)
        slip2 = New Slip(2, 12, 25, dock1)
        slip3 = New Slip(3, 14, 25, dock1)
        'verify Dock to Slip association (1 to many)
        'first get ArrayList of slips from dock
        Dim slips As ArrayList = dock1.GetSlips()
        'get dock information and count of slips
        Console.WriteLine("Dock " & dock1.GetDockId() & " has " _
         & slips.Count() & " slips")
        'get information about each slip from slip's ArrayList
        Console.WriteLine("Slips for Dock " & dock1.GetDockId())
        Dim i As Integer
        For i = 0 To slips.Count() - 1
           Console.WriteLine("   " & slips.Item(i).TellAboutSelf())
        Next

        'get dock information from a slip navigating directly (1 to 1)
        Console.WriteLine("The Dock for Slip " & slip1.GetSlipId())
        Console.WriteLine("   " & slip1.GetDock().TellAboutSelf())

        'get dock information from a slip using TellAboutSelf (1 to 1)
        Console.WriteLine("Slip information (including dock) for Slip " _
         & slip1.GetSlipId())
        Console.WriteLine("   " & slip1.TellAboutSelf())

    End Sub

End Class
```

الشكل رقم (٩.١٨)- شفرة إجراء اختبار علاقة الترابط بين الصنف Dock والصنف Slip.

```
Dock 1 has 3 slips
Slips for Dock 1
 Slip 1 width 10 length 20 attached to Dock 1 location Main Cove has electricity True has
water False
 Slip 2 width 12 length 25 attached to Dock 1 location Main Cove has electricity True has
water False
 Slip 3 width 14 length 25 attached to Dock 1 location Main Cove has electricity True has
water False
The Dock for Slip 1
  Dock 1 location Main Cove has electricity True has water False
Slip information (including dock) for Slip 1
 Slip 1 width 10 length 20 attached to Dock 1 location Main Cove has electricity True has
water False
```

الشكل رقم (٩,١٩). مخرجات إجراء اختبار علاقة الترابط بين الصنف Dock الصنف Slip.

وصفنا أولاً متغير إشارة (dock1) من النوع Dock وثلاثة متغيرات إشارة (slip1, slip2, slip3) من النوع Slip. ثم عرَّفنا كائناً من الصنف Dock وثلاثة كائنات من الصنف Slip. ولقد مررنا متغير إشارة كائن الصنف Dock (dock1) إلى إجراء إنشاء كائنات الصنف Slip؛ ولذلك ترتبط كائنات الصنف Slip الثلاثة بكائن الصنف Dock. وأيضاً يرتبط كائن الصنف Dock بكائنات الصنف Slip الثلاثة تلقائياً.

والآن يمكن اختبار علاقة الترابط بين الصنفين في الاتجاهين. أولاً، نختبر الاتجاه واحد إلى مجموعة بواسطة استرجاع كائن الصنف ArrayList من المتغير dock1 مستخدمين GetSlips. كما يمكن التحقق من عدد كائنات الصنف Slip باستدعاء الصفة Count المعرفة داخل الصنف ArrayList هكذا:

```
'first get ArrayList of slips from dock
Dim slips As ArrayList = dock1.GetSlips()
'get dock information and count of slips
Console.WriteLine("Dock " & dock1.GetDockId() & " has" _
 & slips.Count() & " slips")
```

والآن نستطيع التعامل مع كائنات الصنف Slip الثلاثة بالتجول داخل كائن المصفوفة ArrayList وذلك بكتابة التكرار For الذي يتكرر إلى أن تصل إلى نهاية المصفوفة. نستخدم الخاصية Item، المعرفة لدى الصنف ArrayList، لاسترجاع كائنات الصنف Slip واحد تلو الآخر. ونستخدم الإجراء TellAboutSelf الخاص بالصنف Slip لاسترجاع معلومات كل مرسى:

```
'get information about each slip from slips ArrayList
Console.WriteLine("Slips for Dock " & dock1.GetDockId())
Dim i As Integer
For i = 0 To slips.Count() - 1
    Console.WriteLine("  " & slips.Item(i).TellAboutSelf())
Next
```

يختبر الجزء الأخير من إجراء الاختبار علاقة الترابط بالاتجاه الآخر (واحد إلى واحد)، حيث نسترجع كائن الصنف Dock من أحد كائنات الصنف Slip. ثم نستدعي الإجراء TellAboutSelf الخاص بالصنف Dock لاسترجاع معلومات الرصيف. كذلك نستدعي الإجراء TellAboutSelf الخاص بالصنف Slip لاسترجاع معلومات كل من المرسى والرصيف الذي يحتويه:

```
'get dock information from a slip navigating directly (1 to 1)
Console.WriteLine("The Dock for Slip " & slip1.GetSlipId())
Console.WriteLine("   " & slip1.GetDock().TellAboutSelf())

'get dock information from a slip using TellAboutSelf (1 to 1)
Console.WriteLine("Slip information (including dock) for Slip " _
 & slip1.GetSlipId())
Console.WriteLine("   " & slip1.TellAboutSelf())
```

إضافة الصنف Boat والصنف Customer إلى مثال الصنف Slip
Adding Boat and Customer Classes to Slip Example

يجب تعديل الصنف Slip لتضيف إليه متغير إشارة من النوع Boat كصفة. كذلك إضافة إجراءي وصول لهذه الصفة (الإجراء SetBoat والإجراء GetBoat)؛ وذلك لوجود علاقة ترابط بين الصنف Slip والصنف Boat. ولأن علاقة الترابط بين الصنف Boat والصنف Slip علاقة اختيارية، فلا نحتاج إلى تعديل إجراء إنشاء الصنف Slip لكي يحتوي على متغير إشارة من النوع Boat.

نحتاج أيضاً إلى تعديل الصنف Boat لكي يتم ربطه مع الصنف Slip، حيث نضيف متغير إشارة من النوع Slip ثم نضيف إجراءي وصول لإسناد هذه الصفة داخل الصنف Boat واسترجاعها. ونضيف أيضاً الإجراء AssignBoatToCustomer المسؤول عن إقامة العلاقة بين الصنفين. كما لا نحتاج إلى تعديل إجراء الإنشاء حيث لا يرتبط كائن الصنف Boat في البداية بكائن الصنف Slip. بل تسند القيمة Nothing إلى متغير الإشارة (انظر إلى تعريف الصنف Boat الجزئي في الشكل رقم ٩,٢٠).

إن تعريف الصنف Boat يحتوي بالفعل على علاقة ربط بالصنف Customer؛ ولذلك يمكن أن يحتوي هذا المثال على الصنف Customer دون تعديل. والآن يتضح لنا أهمية علاقات الربط بين الأصناف. فإذا كان لدينا كائن من الصنف Customer (عميل ما)، نستطيع أن نتجول من كائن إلى كائن للحصول على المركب الذي يملكه هذا العميل. وإذا كان لدينا كائن من الصنف Dock (رصيف)، نستطيع أن نتجول للحصول على مراسي هذا الرصيف وكذلك العملاء الذين يؤجرون هذه المراسي بل الحصول على مراكب هؤلاء العملاء.

```
' Boat class associates with Customer and Slip

Public Class Boat

    'attributes
    Private stateRegistrationNo As String
    Private length As Single
    Private manufacturer As String
    Private year As Integer

    'reference variables for Customer and Slip instances
    Private theCustomer As Customer
    Private theSlip As Slip

    'constructor (four parameters plus customer reference)
    Public Sub New(ByVal aStateRegistrationNo As String, _
       ByVal aLength As Single, _
       ByVal aManufacturer As String, ByVal aYear As Integer, _
       ByVal aCustomer As Customer)
       'set values of attributes using accessor methods
       SetStateRegistrationNo(aStateRegistrationNo)
       SetLength(aLength)
       SetManufacturer(aManufacturer)
       SetYear(aYear)
       'because boat must have a customer, assign it here
       AssignBoatToCustomer(aCustomer)
    End Sub
    'custom method to associate with customer in both directions
    Public Sub AssignBoatToCustomer(ByVal aCustomer As Customer)
       SetCustomer(aCustomer)    'set customer reference for boat
       aCustomer.SetBoat(Me)     'ask customer to set its boat to Me
    End Sub

    'custom method to associate with slip in both directions
    Public Sub AssignBoatToSlip(ByVal aSlip As Slip)
       SetSlip(aSlip)     'set slip reference for boat
       aSlip.SetBoat(Me)  'ask slip to set its boat to Me
    End Sub

    'TellAboutSelf method that gets customer info also
    Public Overridable Function TellAboutSelf() As String
       Return stateRegistrationNo & ", " & length & ", " _
       & manufacturer & ", " & year _
       & " customer for boat is " & theCustomer.TellAboutSelf()
    End Function

    (Continue with Get and Set accessor methods)
    . . .
    . . .

End Class
```

الشكل رقم (٩,٢٠). جزء من تعريف الصنف Boat الخاص بالارتباط مع الصنفين Slip و Customer.

يوضح الشكل رقم (٩,٢١) إجراء اختبار شامل لعلاقات الترابط المقدمة في هذا المثال، كما يوضح الشكل رقم (٩,٢٢) مخرجات هذا الإجراء.

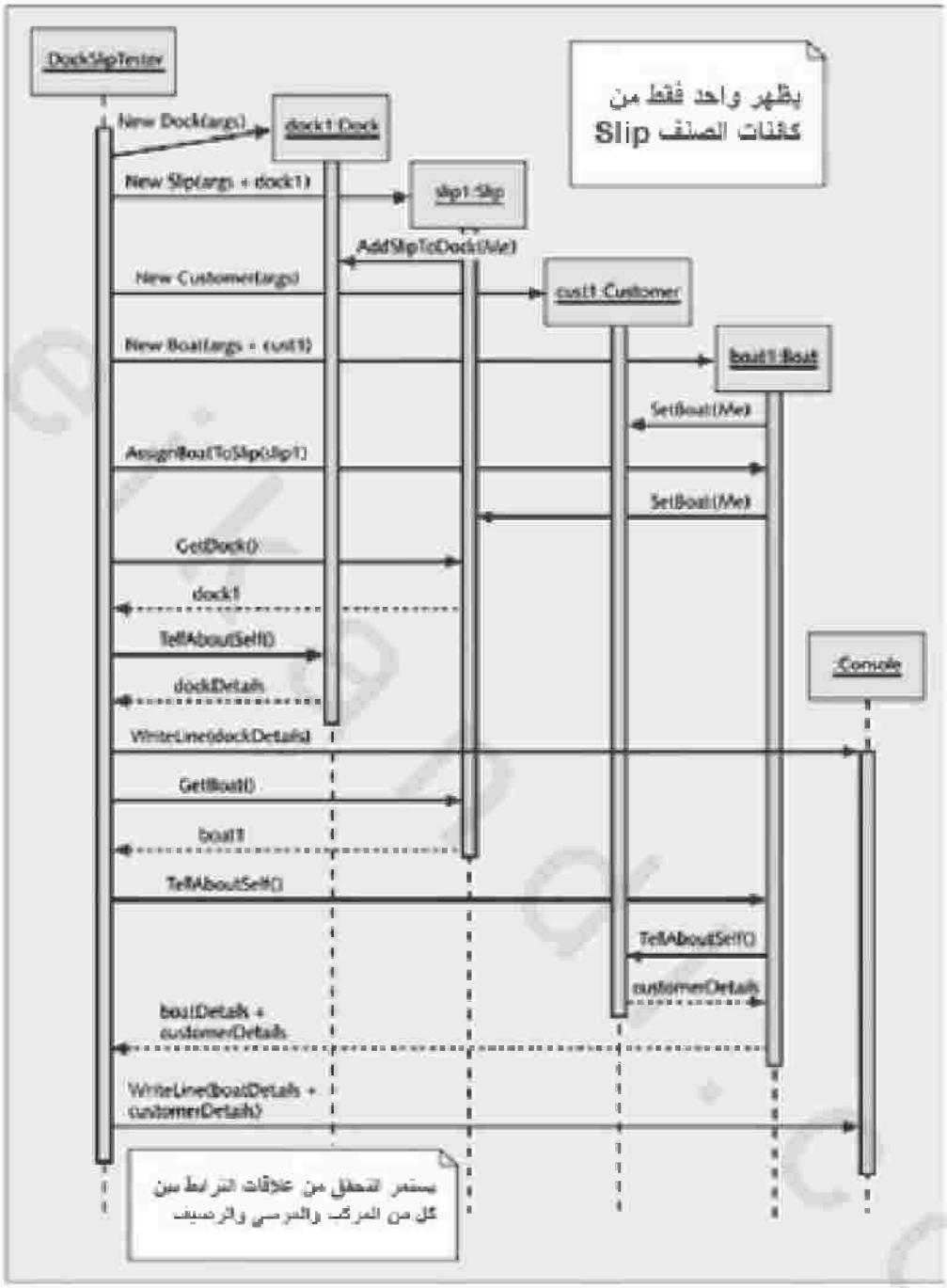
يوضح الشكل رقم (٩,٢٣) نموذج Sequence Diagram للتفاعلات التي تحدث بين الكائنات لتنفيذ إجراء الاختبار السابق. يوصف الإجراء متغير إشارة للصنف Dock ومتغيرا إشارة للصنف Slip. ثم أنشأنا كائناً للصنف Dock وكائنان للصنف Slip. ولقد أقيمت علاقة الترابط بينها في كلا الاتجاهين بواسطة تنفيذ إجراء إنشاء الصنف Slip. ثم أنشأنا كائنان من الصنف Customer وكائنان من الصنف Boat وأقيمت بينها علاقة الترابط بواسطة إجراء إنشاء الصنف Boat. وأخيراً، تم ربط كل مركب (كائن من الصنف Boat) إلى مرسى (كائن من الصنف Slip) بواسطة الإجراء AssignBoatToSlip الذي ينجز علاقة الربط الاختيارية بينها.

```
Private Sub btnAddBoatsAndCustomers_Click(ByVal sender As _
System.Object, ByVal e As System.EventArgs) _
Handles btnAddBoatsAndCustomers.Click

    'declare dock and slip references
    Dim dock1 As Dock
    Dim slip1 As Slip
    Dim slip2 As Slip

    'create customers
    Dim cust1 As New Customer("Eleanor", "Atlanta", "123-4567")
    Dim cust2 As New Customer("JoAnn", "St Louis", "987-6543")

    'create boats associated with customers
    Dim boat1 As New Boat("GA34567", 28, "Tartan", 2002, cust1)
    Dim boat2 As New Boat("MC98765", 32, "Catalina", 2003, cust2)

    'add boats to slips
    boat1.AssignBoatToSlip(slip1)
    boat2.AssignBoatToSlip(slip2)

    'get dock information from a slip navigating directly (1 to 1)
    Console.WriteLine("The Dock for Slip " & slip1.GetSlipId())
    Console.WriteLine("  " & slip1.GetDock().TellAboutSelf())

    'get boat and customer information for slip navigating
    'directly (1 to 1)
    Console.WriteLine("The Boat for Slip " & slip1.GetSlipId())
    Console.WriteLine("  " & slip1.GetBoat().TellAboutSelf())

    'get dock information navigating from customer
    Console.WriteLine("The Dock for Customer " & cust1.GetName())
    Console.WriteLine("  "
     & cust1.GetBoat().GetSlip().GetDock().TellAboutSelf())

    'navigate from dock to each slip, its boat and customer
    'First get ArrayList of slips from dock
    Dim slips As ArrayList = dock1.GetSlips()
    'get slip information using slip's ArrayList
    Console.WriteLine("Dock " & dock1.GetDockId() & " has " _
     & slips.Count() & " slips")
    Console.WriteLine("Slips and details for Dock " _
     & dock1.GetDockId())
    'iterate through each slip
    Dim i As Integer
    For i = 0 To slips.Count() - 1
       Console.WriteLine("  " & slips.Item(i).TellAboutSelf())
       Console.WriteLine("  boat "
        & slips.Item(i).GetBoat().TellAboutSelf())
    Next

End Sub
```

الشكل رقم (٩,٢١). شفرة إجراء اختبار علاقة الترابط بين الأصناف **Customer** و **Boat** و **Dock** و **Slip**.

```
The Dock for Slip 1
  Dock 1 location Main Cove has electricity True has water False
The Boat for Slip 1
  GA34567, 28, Tartan, 2002 customer for boat is Eleanor, Atlanta, 123-4567
The Dock for Customer Eleanor
  Dock 1 location Main Cove has electricity True has water False
Dock 1 has 2 slips
Slips and details for Dock 1
  Slip 1 width 10 length 20 attached to Dock 1 location Main Cove has electricity True has
water False
  boat GA34567, 28, Tartan, 2002 customer for boat is Eleanor, Atlanta, 123-4567
  Slip 2 width 12 length 25 attached to Dock 1 location Main Cove has electricity True has
water False
  boat MC98765, 32, Catalina, 2003 customer for boat is JoAnn, St Louis, 987-6543
```

الشكل رقم (٩,٢٢). مخرجات إجراء اختبار علاقة الترابط بين الأصناف **Customer** و **Boat** و **Dock** و **Slip**.



الشكل رقم (٩,٢٣). نموذج Sequence Diagram لإجراء اختبار علاقة الترابط بين الأصناف الأربعة.

يمكن استخدام كائن الصنف Slip للوصول إلى جميع إجراءات كائن الصنف Dock المرتبط به هكذا:

```
'get dock information from a slip navigating directly '(1:1)
Console.WriteLine("The Dock for Slip " & slip1.GetSlipId())
Console.WriteLine("   " & slip1.GetDock().TellAboutSelf())
```

ثم نستطيع التعامل مع إجراءات كائن الصنف Slip للوصول إلى معلومات كل من المركب الذي يرسي عليه والعميل المالك لهذا المركب هكذا:

```
'get boat and customer information for slip navigating
'directly (1:1)
Console.WriteLine("The Boat for Slip " & slip1.GetSlipId())
Console.WriteLine("   " & slip1.GetBoat().TellAboutSelf())
```

وأيضاً يمكن الوصول للمركب والمرسى عن طريق التجول من العميل إليها لعرض معلومات عنها وعن الرصيف الذي ينتمي إليه هذا المرسى هكذا:

```
'get dock information navigating from customer
Console.WriteLine("The Dock for Customer " & cust1.GetName())
Console.WriteLine("   " _
 & cust1.GetBoat().GetSlip().GetDock().TellAboutSelf())
```

اختبرنا أيضاً علاقة الترابط "واحد إلى مجموعة" التي تربط الرصيف بالمراسي الملحقة به، حيث تجولنا من الرصيف (كائن Dock) للحصول على المراسي التي تنتمي إليه (كائن من ArrayList). ثم تجولنا داخل هذه المراسي مستخدمين التكرار For حيث نستدعي إجراء الوصول GetBoat للحصول على المركب الذي يرسي على هذا المرسى. ثم نستدعي الإجراء TellAboutSelf من المركب لاسترجاع معلومات حول المركب وحول العميل الذي يملكه. وكل هذا حدث بفضل علاقات الترابط التي جعلتنا نصل إلى كل عميل يملك مركباً يرسي على مرسي ينتمي إلى رصيف (نقطة البداية) هكذا:

```
Dim slips As ArrayList = dock1.GetSlips()
'get slip information using slips ArrayList
Console.WriteLine("Dock " & dock1.GetDockId() & " has " _
 & slips.Count() & " slips")
Console.WriteLine("Slips and details for Dock " _
 & dock1.GetDockId())
'iterate through each slip
Dim i As Integer
For i = 0 To slips.Count() - 1
    Console.WriteLine("   " & slips.Item(i).TellAboutSelf())
    Console.WriteLine("   boat " _
     & slips.Item(i).GetBoat().TellAboutSelf())
Next
```

إنشاء الصنف المرافق (الصنف Lease) واستخدامه

Creating and Using an Association Class—Lease

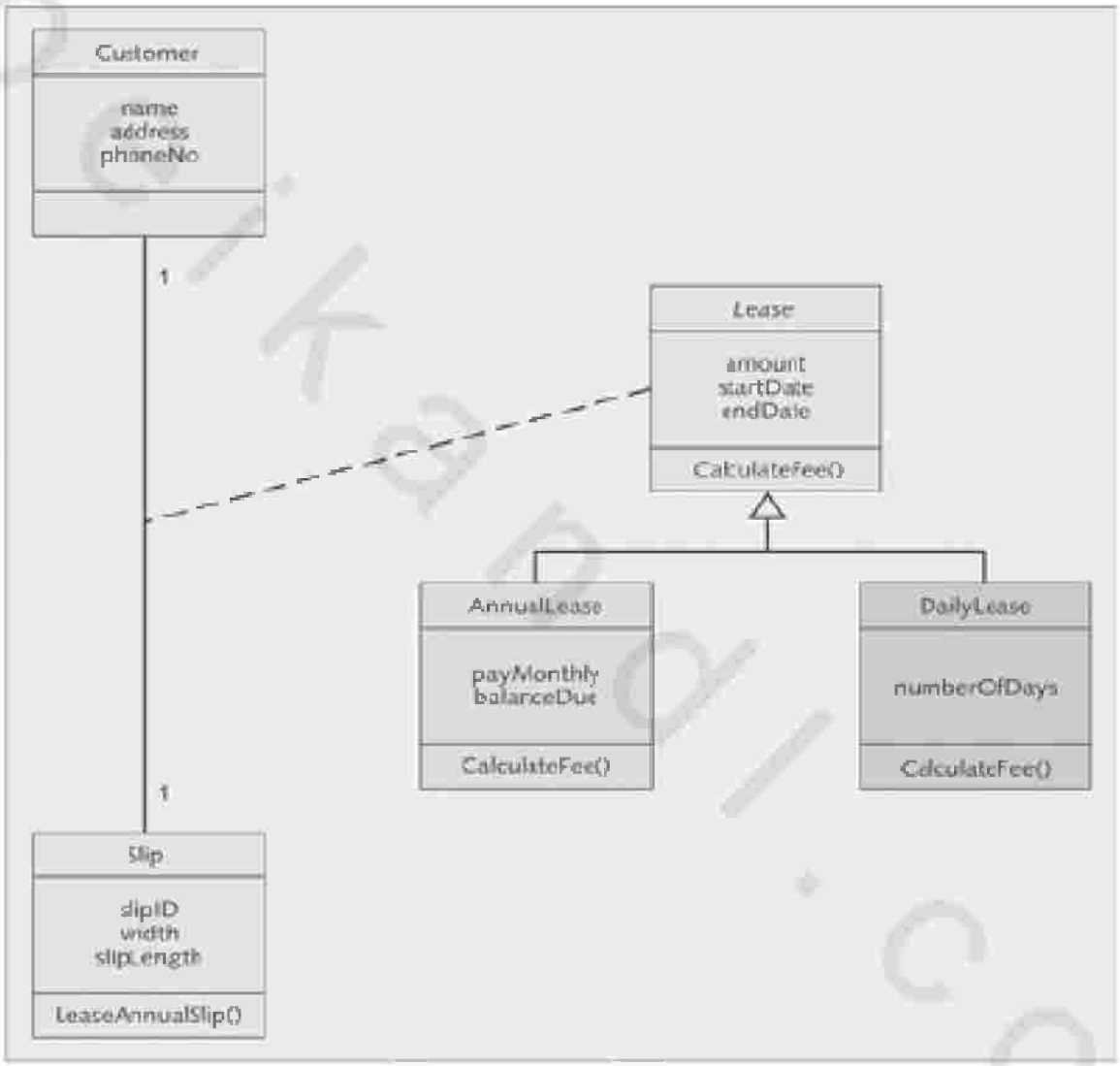
تعرفنا في الفصل الثامن على الصنف الرئيس Lease وعلى أصنافه الفرعية (الصنف AnnualLease والصنف DailyLease)، حيث يمثل الصنف Lease علاقة ترابط بين الصنف Customer والصنف Slip. ولكنها علاقة تحتوي على صفات (تاريخ بداية العقد وتاريخ نهاية العقد وقيمة العقد).يوضح الشكل رقم (٩,٢٤) جزءاً من نموذج Class Diagram لنظام شركة برادشو مارينا، حيث يرتبط الصنف Lease بخط متقطع مع علاقة الترافق التي تربط كلاً من الصنف Slip والصنف Customer. يرتبط الصنف Customer والصنف Slip بعلاقة "واحد إلى واحد" مما يعني أنه يتم إنشاء عقد جديد (Lease) لكل عميل (Customer) يؤجر مرسى (Slip). وغالباً يتواجد الصنف المرافق نتيجة علاقة ارتباط "مجموعة إلى مجموعة" (Many-to-Many) حيث يحق للعميل أن يؤجر العديد من المراسي ويمكن للمرسى أن يؤجر للعديد من العملاء مع مرور الوقت.

يجب أن يتم تعديل الصنف الرئيس Lease المقدم في الفصل الثامن لكي يشمل على متغير إشارة من النوع Slip ومتغير إشارة من النوع Customer (الشكل رقم ٩,٢٥). هذا بالإضافة إلى تعريف إجراءات وصول لإسناد هاتين الصفتين واسترجاعهما. ويمكن أن يبقى الصنف Lease صنفاً مجرداً (Abstract Class) كما هو.

عزيزي القارئ يجب أن تلاحظ أن الصنف الفرعي AnnualLease والصنف الفرعي DailyLease هما أصناف فرعية للصنف Lease؛ ولذلك سوف يرثا علاقة الترابط بينه وبين كل من الصنف Customer والصنف Slip. ومن ثم لا يتطلب ذلك إدخال أي تعديل إلى كل من الصنف AnnualLease والصنف DailyLease.

يستخدم هذا المثال الصنف Boat أيضاً (الذي قدمناه في الفقرة السابقة) والذي يرتبط بالصنف Customer والصنف Slip ولا يرتبط مباشرة بالصنف Lease؛ ولذلك لا نحتاج إلى تعديل تعريف الصنف Boat. ولكن يجب تعديل الصنف Customer ليحتوي على متغير إشارة من النوع Lease وذلك لإقامة العلاقة من جهة العميل. وبذلك نحتاج إلى إضافة إجراءات وصول لإنشاء هذه الصفة واسترجاعها.

أما الصنف Slip فيحتاج إلى تعديل أكثر حيث يجب إضافة إجراء خاص يكون مسؤولاً عن إنشاء كائن الصنف Lease (العقد) ثم يربطه مع كل من العميل والمرسى. ولكن أولاً نحتاج أن نضيف متغير إشارة من النوع Lease لإقامة العلاقة من جهة المرسى. ثم نضيف إجراءات وصول لإسناد هذه الصفة واسترجاعها. بعد ذلك نضيف الإجراء الخاص LeaseAnnualSlip لإنشاء عقد سنوي (كائن من الصنف AnnualLease). يوضح الشكل رقم (٩,٢٦) تعريف الصنف Slip بعد إضافة التعديلات.



الشكل رقم (٩,٢٤). جزء من نموذج Class Diagram يوضح علاقة ترافق الصنف Lease.

obeikandl.com

```
' Lease -- abstract association class
' between Slip and Customer

Public MustInherit Class Lease

   'attributes
   Private amount As Double
   Private startDate As DateTime
   Private endDate As DateTime

   'references to slip and customer
   Private theSlip As Slip
   Private theCustomer As Customer

   'constructor (1 parameter)
   Public Sub New(ByVal aStartDate As DateTime)
      SetStartDate(aStartDate)
      SetEndDate(Nothing)       ' endDate set by subclass
      SetAmount(0)
      'no customer or slip yet. set by subclass.
      SetCustomer(Nothing)
      SetSlip(Nothing)
   End Sub

   'TellAboutSelf method
   Public Overridable Function TellAboutSelf() As String
      Return (startDate & ", " & endDate & ", " & amount)
   End Function

   'custom method CalculateFee based on slip width
   'abstract method all subclasses must implement
   Public MustOverride Function CalculateFee(ByVal aWidth As Integer) _
   As Single

   'Get accessor methods
   Public Function GetStartDate() As DateTime
      Return startDate
   End Function
   Public Function GetEndDate() As DateTime
      Return endDate
   End Function
   Public Function GetAmount() As Double
      Return amount
   End Function
   Public Function GetCustomer() As Customer
      Return theCustomer
   End Function
   Public Function GetSlip() As Slip
      Return theSlip
   End Function

   'Set accessor methods
   Public Sub SetStartDate(ByVal aStartDate As DateTime)
      startDate = aStartDate
   End Sub
   Public Sub SetEndDate(ByVal anEndDate As DateTime)
      endDate = anEndDate
   End Sub
   Public Sub SetAmount(ByVal anAmount As Double)
      amount = anAmount
   End Sub
   Public Sub SetCustomer(ByVal aCustomer As Customer)
      theCustomer = aCustomer
   End Sub
   Public Sub SetSlip(ByVal aSlip As Slip)
      theSlip = aSlip
   End Sub

End Class
```

الشكل رقم (٩,٢٥). تعريف الصنف Lease المعدل.

```
' Slip class -
' Modified to associate with Dock and Boat
' Modified to associate with Lease
' With custom method LeaseAnnualSlip

Public Class Slip

   'attributes
   Private slipId As Integer
   Private slipWidth As Integer
   Private slipLength As Integer

   'reference variables point to one dock, one boat, and one lease
   Private theDock As Dock
   Private theBoat As Boat
   Private theLease As Lease

   'constructor (three parameters plus a Dock reference)
   Public Sub New(ByVal aSlipId As Integer, _
    ByVal aSlipWidth As Integer, _
    ByVal aSlipLength As Integer, ByVal aDock As Dock)
      'invoke setter methods to populate attributes
      SetSlipId(aSlipId)
      SetSlipWidth(aSlipWidth)
      SetSlipLength(aSlipLength)
    'invoke custom method for Dock association
    AddSlipToDock(aDock)
    'set boat and lease to nothing initially
    SetBoat(Nothing)
    SetLease(Nothing)
End Sub

'custom method creates an annual lease instance
'slip takes responsibility for much processing
Public Sub LeaseAnnualSlip(ByVal aCustomer As Customer, _
 ByVal aStartDate As DateTime, ByVal isPaymonthly As Boolean)
    'create AnnualLease instance and assign it to theLease
    theLease = _
     New AnnualLease(aStartDate, slipWidth, isPaymonthly)
    'tell lease to set its slip to me
    theLease.SetSlip(Me)
    'tell lease to set its customer
    theLease.SetCustomer(aCustomer)
    'tell customer to set its lease
    aCustomer.SetLease(theLease)
End Sub

'assign slip to dock (both directions of association)
Public Sub AddSlipToDock(ByVal aDock As Dock)
    theDock = aDock    ' assign the dock reference to attribute
    aDock.AddSlip(Me) ' ask dock to add Me as slip
End Sub

'custom TellAboutSelf gets slip and dock information
Public Function TellAboutSelf() As String
    Dim slipDetails As String
    slipDetails = "Slip " & slipId _
     & " width " & slipWidth & " length " & slipLength _
     & " attached to " & theDock.TellAboutSelf()
    Return slipDetails
End Function

'get accessor methods
Public Function GetSlipId() As Integer
    Return slipId
End Function
Public Function GetSlipWidth() As Integer
    Return slipWidth
End Function
Public Function GetSlipLength() As Integer
    Return slipLength
End Function
Public Function GetDock() As Dock
    Return theDock
End Function
Public Function GetBoat() As Boat
    Return theBoat
End Function
```

الشكل رقم (٩,٢٦). تعريف الصنف Slip المعدل.

```
    Public Function GetLease() As Lease
        Return theLease
    End Function

    'sub accessor methods
    Public Sub SetSlipId(ByVal aSlipId As Integer)
        slipId = aSlipId
    End Sub
    Public Sub SetSlipWidth(ByVal aSlipWidth As Integer)
        slipWidth = aSlipWidth
    End Sub
    Public Sub SetSlipLength(ByVal aSlipLength As Integer)
        slipLength = aSlipLength
    End Sub
    Public Sub SetBoat(ByVal aBoat As Boat)
        theBoat = aBoat
    End Sub
    Public Sub SetLease(ByVal aLease As Lease)
        theLease = aLease
    End Sub

End Class
```

تابع الشكل رقم (٩,٢٦).

ينشئ الإجراء LeaseAnnualSlip أولاً كائناً من الصنف AnnualLease الذي يتم إسناده إلى متغير الإشارة theLease المعرف لدى الصنف Slip وذلك لإضافة العلاقة بين المرسى والعقد من جهة المرسى (يوجد التعريف الكامل للصنف AnnualLease في الفصل الثامن). عزيزي القارئ هل لاحظت أنه من الممكن أن نسند متغير إشارة لصنف فرعي إلى متغير إشارة لصنف رئيس كما فعلنا في أوامر الإجراء LeaseAnnualSlip التالية:

```
'custom method creates an annual lease instance
'slip takes responsibility for much processing
Public Sub LeaseAnnualSlip(ByVal aCustomer As Customer, _
 ByVal aStartDate As DateTime, ByVal isPaymonthly As Boolean)
     'create AnnualLease instance and assign it to theLease
     theLease = _
      New AnnualLease(aStartDate, slipWidth, isPaymonthly)
     'tell lease to set its slip to this lease
     theLease.SetSlip(Me)
     'tell lease to set its customer
     theLease.SetCustomer(aCustomer)
     'tell customer to set its lease
     aCustomer.SetLease(theLease)
End Sub
```

وبالرجوع إلى إجراء إنشاء كائن الصنف AnnualLease، سنجد أننا قمنا بتمرير تاريخ بداية العقد (الذي يمرر إلى إجراء إنشاء الصنف Lease) ومن ثم سوف يتم استنتاج تاريخ نهاية العقد وإسناده. ومررنا عرض المرسى (الذي يستخدم لحساب قيمة العقد) وكذلك ما إذا كان العميل سيدفع قيمة العقد دفعة واحدة أو على أقساط شهرية. وبذلك سيتم حساب قيمة الأقساط المتبقية. أما الأمر التالي theLease.SetSlip(Me) فإنه يطلب من العقد أن يقيم العلاقة من جهته مع المرسى. ثم يطلب من العقد أيضاً أن يقيم العلاقة من جهته مع العميل

theLease.SetCustomer(aCustomer). وأخيراً نطلب من العميل أن يقيم العلاقة من جهته مع العقد aCustomer.SetLease(theLease).

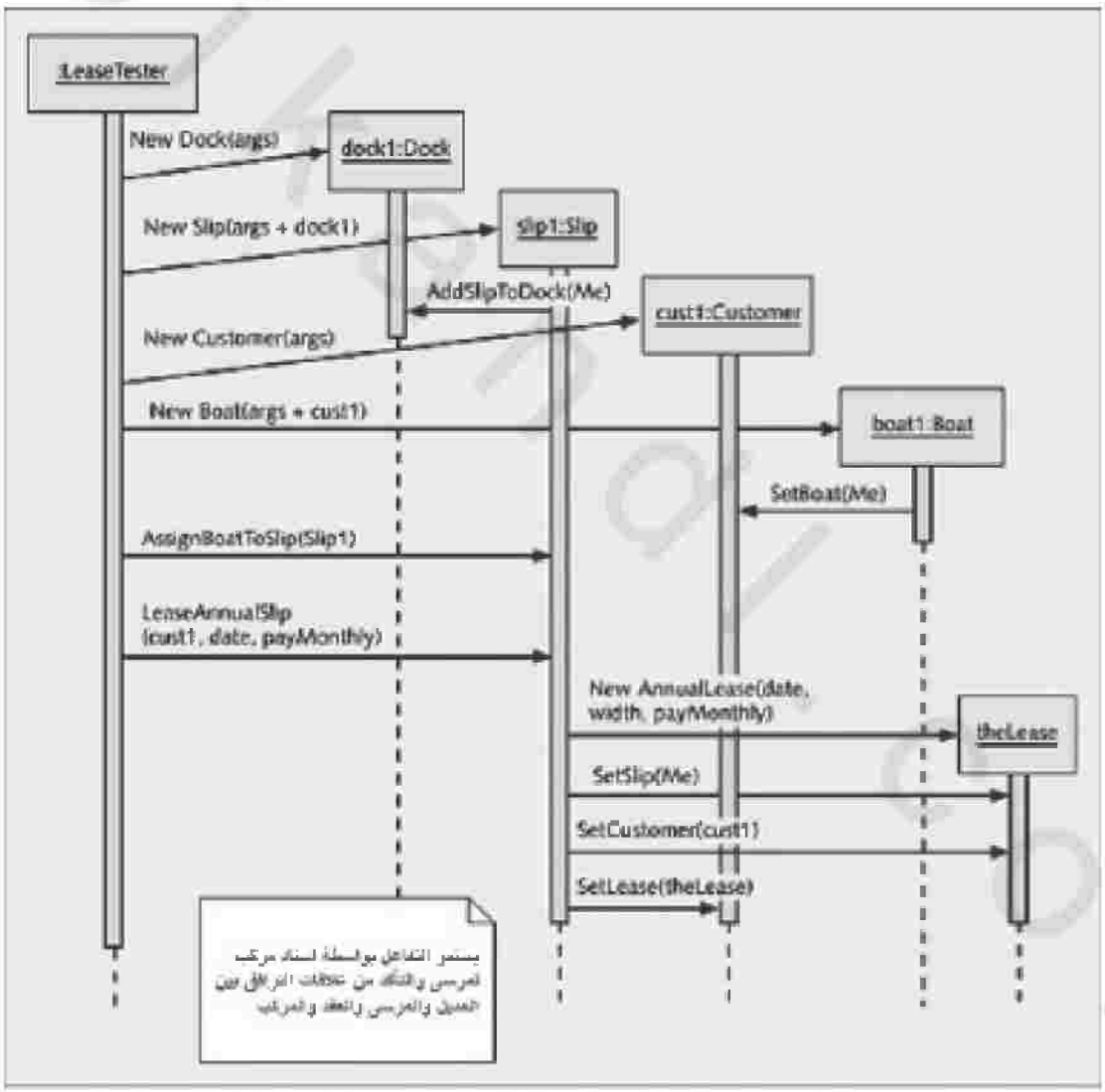
قمنا بتعديل تعريف كل من الصنف Slip والصنف Customer والصنف Lease، أما الصنف Boat والصنف AnnualLease فتم إضافتهما إلى المثال دون تعديل. والآن نستطيع أن نختبر الإجراء LeaseAnnualSlip بالإجراء الموضح في الشكل رقم (٩,٢٧). كما يوضح الشكل رقم (٩,٢٨) نموذج Sequence Diagram الذي يمثل التعاملات التي ستحدث بين كل من كائن الصنف Slip وكائن الصنف AnnualLease وكائن الصنف Customer. وأخيراً يوضح الشكل رقم (٩,٢٩) مخرجات هذا الإجراء.

```
Private Sub btnLeaseSlip_Click(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles btnLeaseSlip.Click

    'create the dock
    Dim dock1 = New Dock(1, "Main Cove", True, False)

    'create two slip instances for the dock
    Dim slip1 = New Slip(1, 10, 20, dock1)
    Dim slip2 = New Slip(2, 12, 25, dock1)

    'create customers
    Dim cust1 As New Customer("Eleanor", "Atlanta", "123-4567")
    Dim cust2 As New Customer("James", "St Louis", "987-6543")

    'create boats associated with customers
    Dim boat1 As New Boat("GA34567", 28, "Tartan", 2002, cust1)
    Dim boat2 As New Boat("MO98765", 32, "Catalina", 2003, cust2)

    'lease slips to customers
    slip1.LeaseAnnualSlip(cust1, New DateTime(2004, 3, 15), True)
    slip2.LeaseAnnualSlip(cust2, New DateTime(2004, 4, 20), False)

    'add boats to slips using lease references
    slip1.GetLease().GetCustomer().GetBoat().AssignBoatToSlip(slip1)
    slip2.GetLease().GetCustomer().GetBoat().AssignBoatToSlip(slip2)

    'get dock information from a slip navigating directly (1 to 1)
    Console.WriteLine("The Dock for Slip " & slip1.GetSlipId())
    Console.WriteLine("  is " & slip1.GetDock().TellAboutSelf())

    'get boat, customer and lease information for slip
    'navigating directly (1 to 1)
    Console.WriteLine("The Boat details for Slip " _
     & slip1.GetSlipId())
    Console.WriteLine("  is " & slip1.GetBoat().TellAboutSelf())
    Console.WriteLine("  Lease is " _
     & slip1.GetLease().TellAboutSelf())

    'navigate from dock to each slip, its boat, customer & lease
    'first get ArrayList of slips from dock
    Dim slips As ArrayList = dock1.GetSlips()
    'get slip information using slip's ArrayList
    Console.WriteLine("Dock " & dock1.GetDockId() & " has " _
     & slips.Count() & " slips")
    Console.WriteLine("Slips and details for Dock " _
     & dock1.GetDockId())
```

الشكل رقم (٩,٢٧). شفرة إجراء اختبار الإجراء LeaseAnnualSlip متضمنة الأصناف الأربعة.

```
'iterate through each slip
Dim i As Integer
For i = 0 To slips.Count() - 1
   Console.WriteLine("   " & slips.Item(i).TellAboutSelf())
   Console.WriteLine("  boat " _
    & slips.Item(i).GetBoat().TellAboutSelf())
   Console.WriteLine("  lease " _
    & slips.Item(i).GetLease().TellAboutSelf())
Next

'navigate from slip back to slip via lease, customer, and boat
Console.WriteLine("Example navigating from slip0 to slip1: " _
 & vbCrLf _
 & slip0.GetLease().GetCustomer().GetBoat().GetSlip().TellAboutSelf())

    End Sub
```

تابع الشكل رقم (٩,٢٧).



الشكل رقم (٩,٢٨). نموذج Sequence Diagram للإجراء LeaseAnnualSlip.

```
The Dock for Slip 1
  Is Dock 1 location Main Cove has electricity True has water False
The Boat details for Slip 1
  Is GA34567, 28, Tartan, 2002 customer for boat is Eleanor, Atlanta, 123-4567
  Lease is 3/15/2004, 3/15/2005, 800, 733.3333, True
Dock 1 has 2 slips
Slips and details for Dock 1
  Slip 1 width 10 length 20 attached to Dock 1 location Main Cove has electricity True has
water False
  boat GA34567, 28, Tartan, 2002 customer for boat is Eleanor, Atlanta, 123-4567
  lease 3/15/2004, 3/15/2005, 800, 733.3333, True
  Slip 2 width 12 length 25 attached to Dock 1 location Main Cove has electricity True has
water False
  boat MO98765, 32, Catalina, 2003 customer for boat is JoAnn, St Louis, 987-6543
  lease 4/20/2004, 4/20/2005, 900, 0, False
Example navigating from slip1 to slip1:
Slip 1 width 10 length 20 attached to Dock 1 location Main Cove has electricity True has
water False
```

الشكل رقم (٩,٢٩). مخرجات إجراء اختبار الإجراء LeaseAnnualSlip.

عزيزي القارئ لاحظ أن الرمز vbCrLf يعني إضافة سطر جديد في هذا المكان.

ينشئ إجراء الاختبار أولاً كائناً من الصنف Dock وكائنان من الصنف Slip. ثم ينشئ كائنان من الصنف Customer وكائنان من الصنف Boat. وتذكر أن كائن الصنف Slip يقيم علاقة الترابط مع كائن الصنف Dock في كلا الاتجاهين تلقائياً. وكذلك كائن الصنف Boat يقيم علاقة الترابط مع كائن الصنف Customer في كلا الاتجاهين تلقائياً.

يتم تأجير كل مرسى إلى عميل هكذا:

```
'lease slips to customers
slip1.LeaseAnnualSlip(cust1, New DateTime(2004, 3, 15), True)
slip2.LeaseAnnualSlip(cust2, New DateTime(2004, 4, 20), False)
```

لاحظ أن تاريخ بداية العقدين يتم إنشائهما داخل إجراء إنشاء العقد السنوي. ولاحظ أيضاً أن العميل الأول سيدفع قيمة العقد على دفعة واحدة، أما العميل الثاني فسيدفع قيمة العقد على أقساط شهرية. وتذكر أيضاً أن معظم المعالجة تتم داخل الإجراء LeaseAnnualSlip الذي ينشئ أولاً كائناً من الصنف AnnualLease ثم إقامة

العلاقة بينه وبين كل من العميل والمرسى. وأخيراً تستدعى الأوامر التالية والمسؤولة عن إسناد كل مركب داخل مرسى. ومن الملاحظ أننا فصلنا هذه الأوامر عن الإجراء LeaseAnnualSlip لأنه ربما لا يملك العميل مركباً الآن ومن المخطط أن يشتري مركباً في القريب العاجل. لاحظ أن الأوامر التالية تتحول من كائن إلى كائن بداية من كائنات الصنف Slip وصولاً إلى كائن الصنف Boat ثم استدعاء الإجراء AssignBoatToSlip.

```
'add boats to slips using lease references
slip1.GetLease().GetCustomer().GetBoat().AssignBoatToSlip(slip1)
slip2.GetLease().GetCustomer().GetBoat().AssignBoatToSlip(slip2)
```

الآن تستطيع أن تتحقق من صحة جمع علاقة الترابط :

- علاقة المرسى بالرصيف.
- علاقة المرسى بالمركب.
- علاقة المرسى بالعقد.
- علاقة الرصيف بالمرسى ثم بالمركب ثم بالعميل ثم بالعقد.

إن آخر أوامر اختبار تتجول بين الكائنات لاختبار علاقات الترابط بداية من المرسى إلى العقد وصولاً بالمركب والعميل ثم العودة ثانية إلى المرسى في دائرة كاملة هكذا :

```
'navigate from slip back to slip via lease, Customer, and Boat
Console.WriteLine("Example navigating from slip1 to slip1: " _
 & vbCrLf _
 & slip1.GetLease().GetCustomer().GetBoat().GetSlip().TellAboutSelf())
```

ملخص الفصل
Chapter Summary

- تأخذ علاقات المرافقة (Association Relationships) بين الأصناف درجة كبيرة من الأهمية في تطوير نظم المعلومات، ويتم تمثيل علاقة المرافقة بين الأصناف على شكل خط في نموذج Class Diagram وتوصف بجملة فعلية. فعلى سبيل المثال يحتوي نموذج Class Diagram لشركة برادشو مارينا على العديد من العلاقات المرافقة مثل : العميل (الصنف Customer) يملك مركباً (الصنف Boat) والمركب يرسي في مرسى (الصنف Slip) وهكذا.
- يجب أن نلاحظ أن علاقة الترافق بين كائنين تمثل بعلاقة ثنائية الاتجاه، فعلى سبيل المثال علاقة الترافق بين العميل والمركب تمثل بعلاقة ثنائية الاتجاه. الأول : العميل يملك مركباً، والثاني : المركب مملوك للعميل؛ ولذلك يجب أن يأخذ مطورو النظام ذلك في الاعتبار، حيث يتم معالجة كل اتجاه على حده. ربما تختلف

اتجاهات العلاقة، والتي يمكن أن يكون أحدها إجبارياً (Mandatory) والآخر اختيارياً (Optional). كما أن عدد أطراف العلاقة (Multiplicity) بين الطرفين ربما يكون مختلفاً. فعلى سبيل المثال، الرصيف يحتوي على العديد من المراسي ("..١)، أما المرسى فهو جزء من رصيف واحد فقط (١).

- تطور علاقات الترابط واحد- الى- واحد في لغة VB.Net بواسطة تعريف متغير إشارة (Reference Variable) كصفة داخل الصنف الاول وبالتالي يستطيع أن يشير إلى كائن من الصنف الثاني. وبالتالي يمكن تعريف إجراءات مرور لكل صفة لتسهيل التفاعل بين الكائنين.
- تطور علاقات الترابط واحد-إلى-مجموعة في لغة VB .NET بواسطة تعريف كائن من الصنف ArrayList لإنشاء حاوية تخزين متغيرات إشارة. فعلى سبيل المثال، لتطوير علاقة الترافق واحد إلى مجموعة بين المرسى (Slip) والرصيف (Dock) أضفنا صفة باسم slips من النوع ArrayList داخل الصنف Dock لتحتوي على متغيرات إشارة من الصنف Slip، وتعريف إجراءات مرور للتعامل مع هذه الصفة.
- تمكننا علاقات الترابط بين الكائنات من التجول بين الكائنات، فإذا كان لدينا كائن من الصنف Customer (عميل ما)، نستطيع أن نتجول من كائن إلى كائن للحصول على معلومات الرصيف الذي يحتوى على مرسى مركب هذا العميل هكذا:

```
cust1.GetBoat().GetSlip().GetDock().TellAboutSelf()
```

- يمثل صنف المرافق علاقة ترابط بين صنفين ولكنها علاقة تحتوي على صفات، فعلى سبيل المثال يمثل الصنف Lease صنف ترافق، حيث ارتبط وجوده بوجود علاقة ترافق بين الصنف Customer والصنف Slip.

أسئلة المراجعة

Review Questions

١- صف اتجاهي علاقة المرافقة بين السيارة ومالكها.

٢- أعطِ مثالاً للعلاقة الاختيارية، مثل العلاقة بين شخص Person وحيوان أليف Pet. هل يمكن أن يكون اتجاه واحد اختيارياً والآخر يكون إجبارياً؟

٣- أعطِ مثالاً على العلاقة الإجبارية، مثل العلاقة بين الأم Mother والطفل Child. هل كلا الاتجاهين إجباري؟

٤- أعطِ مثالاً على العلاقة التكوينية.

٥- أعطِ مثالاً على العلاقة التجميعية.

٦- بعباراتك الخاصة، صف كيف تبرمج علاقات المرافقة في الفيجوال بيسك.

٧- برمجة علاقات المرافقة واحد إلى واحد كيف تكون مختلفة عن علاقات المرافقة واحد إلى المجموعة؟

٨- اكتب الجملة البرمجية التي تعرف صفة لحيوان أليف لاستخدامها في علاقة ترابط مع شخص.

٩- اكتب رأس إجراء الإنشاء لصنف Pet بمعاملات name و breed ومتغير إشارة Person الذي يرتبط به.

١٠- اكتب الجملة البرمجية التي تعرف ArrayList في الصنف Person للاحتفاظ بمتغيرات إشارة بمجموعة كائنات من الصنف Pet.

١١- اكتب إجراء مرور كامل للصنف Pet يسمى SetPerson وافترض أن الحيوان الأليف Pet لديه شخص Person واحد.

١٢- اكتب إجراء المرور الكامل للصنف Pet يسمى AssignPetToPerson الذي يؤسس العلاقة في كل من الاتجاهين.

١٣- باعتبار أن الأم Mother والطفل Child أصناف تم ذكرها في السؤال رقم ٣ السابق، اكتب الجمل البرمجية لتعريف صفة Mother في Child و children ArrayList في Mother.

١٤- اكتب إجراء المرور الكامل للأطفال Child يسمى AssignChildToMother الذي يؤسس العلاقة في كل من الاتجاهين.

١٥- باعتبار مجموعة من الأصناف التي ترتبط سوياً في هذا الترتيب: House، Street، City، State، Country، Continent، Planet. افترض إجراءات مرور قياسية مستخدمة، مثل GetHouse، GetStreet، ... ، إلخ والذي يرجع متغيرات الإشارة للحالات المرتبطة. اكتب جملة الفيجوال بيسك التي تتجول من House للحصول على اسم الكوكب planet الموجود به House.

أسئلة المناقشة

Discussion Questions

١- لماذا علاقات المرافقة هامة جداً في تطوير نظم المعلومات؟ وهل تتخيل تطوير نظم المعلومات بدون وجود علاقات مرافقة؟ ناقش ذلك.

٢- تمثل علاقة الترافق بين كائنين بعلاقة ثنائية الاتجاه، فعلى سبيل المثال يستطيع المركب أن يعرف المرسى الذي سيرسي عليه، وربما لا يستطيع المرسى أن يعرف المركب الذي سيرسي عليه. ناقش متى يكون مفيداً إنشاء علاقة في اتجاه واحد؟

٣- لقد أوضح هذا الفصل كيف نطور علاقات الترابط واحد-إلى-واحد وعلاقات الترابط واحد-إلى-مجموعة في لغة VB .NET. هل تستطيع أن توضح كيف نطور علاقات الترابط مجموعة-إلى- مجموعة مستخدمين لغة VB .NET؟

٤- تستخدم قواعد البيانات العلاقية المفاتيح الأجنبية لتطوير علاقات الترابط بين الجداول. وضح الفروق الأساسية بين استخدام المفاتيح الأجنبية لتطوير علاقات الترابط داخل قواعد البيانات واستخدام متغيرات الإشارة لتطوير علاقات الترابط بين الكائنات.

مشاريع الفصل
Projects

١- لقد تعاملنا في أسئلة المراجعة على مثال الصنف Person والصنف Pet. قم بتطوير مشروع مستخدماً لغة VB .NET لهذا المثال، وتأكد أن برنامج الاختبار يختبر علاقة الترابط بين الصنفين في كلا الاتجاهين.

٢- لقد تعاملنا في أسئلة المراجعة على مثال الصنف Mother والصنف Child. قم بتطوير مشروع مستخدماً لغة VB .NET لهذا المثال، وتأكد أن برنامج الاختبار يختبر علاقة الترابط بين الصنفين في كلا الاتجاهين.

٣- افترض أن شركة برادشو مارينا أرادت أن تسجل بيانات خدمات المراكب. قم بمراجعة نموذج Class Digram الخاص بالشركة لإضافة الصنف الجديد BoatServiceRecord الذي سيرتبط بالصنف Boat بشرط أنه ربما يسجل لكل مركب خدمة أو أكثر، وكل خدمة يجب أن تسجل لمركب واحد. سيحتوي الصنف BoatServiceRecord على الصفة invoice number (رقم الفاتورة) والصفة service date (تاريخ الخدمة) والصفة total charges (التكلفة الكلية). قم بتطوير مشروع يحتوي على الصنف Boat والصنف BoatServiceRecord والصنف Customer. أضف الإجراء RecordBoatService داخل الصنف Boat المسؤول عن إنشاء كائنات من الصنف BoatServiceRecord ، ثم اكتب برنامج الاختبار المطلوب للمشروع.