الدرس الخامس:

سوف نتناول في هذه المحاضرة المواضيع التالية :

١- الأصناف والكائنات وبيانات الأعضاء ودوال الأعضاء.

٢- تعريف صنف مع دالة عضو واحدة.

٣- تعريف دالة عضو بوسيط واحد.

٤- تطبيق لغة الـ UML على أمثلة هذه المحاضرة.

في الفصل السابق قمنا بشرح برامج تقوم بعرض رسالة وأخذ بيانات من المستخدم وإجراء بعض العمليات الحسابية
عليها وإخراج النتائج، في هذا الفصل سوف نقوم بالبرمجة الكائنية وإنشاء برامج تقوم من خلالها بتشغيل المفاهيم
الأساسية لهذه التقنية. في المحاضرات السابقة كنا نقوم بالمهام من خلالة الدارة الرئيسية main ونقوم بكتابة جميع
العبارات فيها، ولكننا في هذه المحاضرة سوف نبتدي معكم في عزل هذه المهمات عن الدالة الرئيسة وتعليبها لإتاحة
استخدامها في برامج أخرى.

الأصناف والكائنات وأعضاء البيانات ودوال الأعضاء

أخذنا في الفصل المحاضرة الماضية مثال السيارة، حيث قلنا أننا عند الضغط على دواسة البنزين فإن سرعة السيارة سوف تزيد
بغض النظر عن كيفية حدوث هذا بالتفصيل. في ورقة تصميم السيارة، كل تفاصيل هذا موضح والخطوات معروفة لتأدية هذه المهمة
عند كتابة الأكواد، كل مهمة سوف تسدل لدالة لتقوم بها (مثل دالة الـ main كما اوضحنا في الماضرة الثانية). الدالة تصف
الميكانيكيات التي تصف مهمتها وهي تقوم أيضا بإخفاء كل هذا عن مستخدمها.
سوف نبدأ إن شاء الله بإنشاء وحدة برنامج تسمى الصنف لكي تحتوي على الدالة. فعندما نعرّف الدالة داخل صنف فإننا نسميها بعد
ذلك بدالة العضو. يحتوي كل صنف على دالة واحدة أو أكثر. على سبيل المثال، الصنف الذي يمثل حساب بنكي لأحد العملاء يحتوي على
دالة عضو لإيداع مبلغ من المال، وأخرى للسحب من الحساب وآخرى للإستعلام عن الرصيد الحالي.

بعد ذلك سوف تقوم بإنشاء كائن. لماذا نقوم بإنشاء كائن؟ كلي نجيب على هذا السؤال سف نصف الصنف بورقة التصميم المرسومة
لهيكل السيارة والوضائف التي تقوم بها ذلك لصناعة سيارة نستفيد منها، فهل يمكنك أن تقود هذه الورقة وتذهب بها وتقضي حاجياتك ؟
بالطبع لا يمكنك ذلك. فلكي تستفيد من ذلك التصميم، يجب عليك إنشاء سيارة فعلية من هذا التصميم وهذا هو ما يمثل الكائن، والعلاقة
بين ورقة التصميم للسيارة والسيارة كالعلاقة بين الصنف والكائن. ولهذا سميت لغة الـ C++ بلغة الكائن الموجة نسبةً لهذا العملية.
ايضا، يمكننا أن ننشيء عدة سيارات من هذا التصميم، كذلك يمكننا أن ننشيء عدة كائنات من نفس الصنف.
عندما تقود السيارة وتدوس على دواسة البنزين، أنت تقوم بإرسال رسالة إلى السيارة لتؤدي هذه المهمة (قيادة السيارة وزيادة السرعة)
، بالمقابل في هذا البرنامج أنت ترسل رسائل للكائن تعرف بمناداة دالة العضو للكائن لكي يقوم بمهمته. وهذه العملية تسمى " طلب الخدمة
من الكائن ".
إستخدمنا مثال السيارة لإعطاء مقدمة للصنف والكائن ودوال الأعضاء. لكن هناك لكل سيارة خصائص وصفات معينة، مثل عدد الأبواب
، سعة التانك، لون السيارة. هذه الأشياء جزء من التصميم الهندسي للسيارة ، فهي مربوطة بالسيارة عند عملية صيانتها ولا يمكن اهمالها.
مثال آخر أكثر وضوح، كائن الحساب البنكي لعميل معين لديه خاصية الرصيد التي تمثل مقدار المال في حسابة وليس في حساب عميل آخر.
الخصائص تكون معرفة بـ " أعضاء بيانات الصنف " في لغة الـ c++.

نضرة على أمثله هذا الفصل

سوف نتناول في هذه الحاضرة امثله ، عبارة عن برامج صغيرة لتوضيح التقنيات الجديدة.

المثال الأول : يمثل صنف دفتر الدرجات مع دالة عضو واحدة تقوم بعرض رسالة ترحيب عندما يتم استدعاؤها. في هذا المثال سوف
نقوم بإنشاء كائن لهذا الصنف ومن ثم يبدأ العمل الفعلي لهذا الكائن وهو طباعة رسالة ترحيب على الشاشة.
المثال الثاني : هو تعديل للمثال الأول وذلك بإضافة إسم الدورة إلى رسالة الترحيب ، ومن يقوم بتحديد هذا الإسم هو المستخدم.

والجدير بالذكر ، أن مثال دفتر الدرجات ليس فعليا عبارة عن مقدمة بسيطة ، ولا يوجد تخزين للدرجات، حيث نبدأ بذلك في الفصل
الرابع وتخزين البيانات يكون في الفصل السابع.

تعريف صنف مع دالة عضو واحدة
المثال الأول

كود

```
1  // Fig. 3.1: fig03_01.cpp
2  // Define class GradeBook with a member function displayMessage;
```

```
3  // Create a GradeBook object and call its displayMessage function.
4  #include <iostream>
5  using std::cout;
6  using std::endl;
7
8  // GradeBook class definition
9  class GradeBook
10 {
11 public:
12    // function that displays a welcome message to the GradeBook user
13    void displayMessage()
14    {
15       cout << "Welcome to the Grade Book!" << endl;
16    } // end function displayMessage
17 }; // end class GradeBook
18
19 // function main begins program execution
20 int main()
21 {
22    GradeBook myGradeBook; // create a GradeBook object named
myGradeBook
23    myGradeBook.displayMessage(); // call object's displayMessage function
24    return 0; // indicate successful termination
25 } // end main
```

يتكون هذا البرنام من صنف دفتر الدرجات GradeBook الذي من خلالة يقوم المعلم بمتابعة نتائج الإختبارات. يتم انشاء

كائن في الدالة الرئيسة main حيث تستخدم هذا الكائن من خلال دالة العضو لعرض رسالة ترحيب على الشاشة. أولا، سوف

نشرح كيفية تعريف صنف يحتوي على دالة عضو واحدة، ثم نشرح كيفية انشاء كائن لهذا الصنف وايضا كيف نقوم بإستدعاء دالة

العضو لكائن لعرض رسالة الترحيب.

الصنف دفتر الدرجات:

يجب قبل بداية دالة الـ main أن نعرف الصنف كما هو موضح في السطور (٩-١٧) وتسمى هذه السطور " تعريف الصنف defining a class ".نحن نقوم بتعريف الصنف من أجل المترجم لكي يتعرف على الصنف ودوال أعضاؤه وأعضاء بياناته.

تعريف دالة العضو displayMessage في السطور (١٣-١٦) بحيث تعرض رسالة ترحيب على الشاشة كما هو موضح في السطر (١٥). إلى السطر ١٩ويعتبر هذا الصنف عبارة عن ورقة تصميم ولن يتم تشغيله والإستفادة منه إلى بعد انشاء الكائن في السطر (٢٢) في الدالة main. بعدما قمنا بإنشاء كائن نقوم بإستدعاء دالة العضو في الصنف GradeBook من خلال السطر (٢٣).

تعريف الصنف يبدأ عند السطر (٩) والكلمة المفتاحية لهذا التعريف هي كلمة class متبوعة بإسم

الصنف GradeBook.-عادةً ما يكون إسم الصنف يبدأ بحرف كبير لكل كلمة فيه وذلك لتسهيل القراءة. ويسمى هذا الإسلوب في التسمية بالـcamel case السطور (١٠-١٧) هذا الجزء يسمى جسم الصنف ويبدأ بقوس فتح وينتهي بقوس إغلاق ({}) وينتهي بفاصلة منقوطة (الفاصلة المنقوطة ضرورية هنا).

خطأ برمجي شائع

نسيان الفاصلة المنقوطة عند نهاية كل تعريف في الصنف يسبب خطأ نصي.

تذكر ذلك ، أ، الدالة main تستدعى بشكل تلقائي عند كل تنفيذ للبرنامج. هي ليست كمعظم الدوال كدالة العضو displayMessgae، فهذه الدالة يجب أن تقوم بإستدعائها لتؤدي عملها كما هو موضح بالسطر ٢٣. يحتوي السطر ١١ على محدد وصول من نوع عام (public) وذلك لتحديد نوع الوصول لدالة العضو displayMessage بأن يكون عام أي من أي مكان في البرنامج. محدد الوصول يجب أن يتبعه نقطتين فوق بعض (:). فيما بعد سوف نتعلم استخدام محدد وصول من نوع خاص private لتحديد صلاحيات الوصول. كل دالة تقوم بوضيفة معينة وكذلك ربما ترجع قيمة عند اكتمال مهمتها. على سبيل المثال، الدالة التي تقوم بعملية جمع فإنها ترجع قيمة وهي ناتج الجمع. إذن، عندما تُعرف دالة يجب أن يحدد نوع القيمة المُرَجـَعَـه. في السطر ١٣، نوع القيمة المرجعه هو void أي لا شي، بمعنى آخر ، أن الدالة لا ترجع شي ، فقط تقوم بطباعة رسالة ترحيب.

المثالية في تسمية الدوال هو ابتداء إسم الدالة بحرف صغير وكل مقطع بعد ذلك يبدأ بحرف كبير، وذلك كما قمنا به في الدالة displayMessage في طريقة تسميتها. في السطر ١٣، تشير الأقواس بعد إسم الدالة أن هذه دالة ، أما كون الأقـواس فـيها فاضية (ليس بداخلها شي) فهذا يدل على أن الدالة لا تحتاج إلى بيانات إضافية كي تقوم بعملها (سوف ترى دوال تحتاج إلى بيانات إضافية فيها بعد).

السطر ١٣ كاملا يسمى ترويسة الدالة (فاتحة الدالة) function header ويأتي بعده جسم الدالة المحدد بأقواس الفتح والإغلاق ({}) في السطور (١٤، ١٦).

خلال جسم الدالة سوف ترى عبارات توضح ميكانيكية تنفيذ المهمة المكلفة بها هذا الدالة. دالة العضو displayMessage تحتوي عبارة واحدة للتنفيذ (السطر ١٥). ومهمتها طباعة رسالة ترحيب على الشاشة. بعد تنفيذ هذه العبارة تكون الدالة بذلك قد أنهت مهمتها.

خطأ برمجي شائع

إرجاع قيمة لدالة نوع المُرَجـَع فيها void سوف يولد خطأ ترجمه.

خطأ برمجي شائع

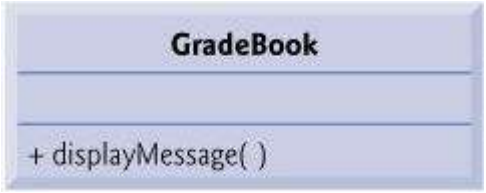
تعريف دالة داخل دالة آخرى.

فحص الصنف GradeBook :

في السطر ٢٢ قمنا بنشاء كائن من نوع GradeBook وأسميناه myGradeBook، لاحظ أن الكائن عبارة عن نوع قام بتعريفه المستخدم وهو GradeBook على أنه صنف ويحتوي على دالة واحدة تقوم بطباعة رسالة على الشاشة، هكذا نخبرالمترجم. في حالة أننا لم نقم بتعريف
هذا النوع كما في الأسطر (٩-١٧) فإن المترجم سوف يرسل رسالة خطأ مفادها أنه لم يتعرف على هذا النوع.
كل صنف جديد أن تقوم بإنشاؤه يعتبر نوع جديد نستطيع أن تستخدمه متى شئت. لهذا لاسبب سميت
هذه اللغة " بلغة قابلة للتمدد ". في السطر ٢٣ نستدعي دالة العضو displayMessage (معرفة في السطور ١٣-١٦)
بإستخدام المتغير myGradeBook متبوع بنقطه (.) ومن ثم إسم الدالة كماهو معين من قبل.
بعد عملية الإستدعاء ، التنفيذ سوف يتفرع ليذهب للدالة displayMessage وينفذ
العبارة (عبارة طباعة رسالة ترحيب على الشاشة) ، وعند انتهاء الدالة يرجع من حيث
تفرع ، أي إلى الدالة main، ثم يكمل التنفيذ إلى أن يصل إلى السطر ٢٤ ، نهاية الدالة الرئيسة.

تمثيل الصنف GradeBook بالـ UML

الـ UML لغة رسمية مستقلة ، يتسخدمها المبرمجون لتمثيل أنظمتهم الكونة من كائنات موجهه في نمط قياسي. في الـ UML كل عضو كل صنف يكون على شكل انموذج رسم تخطيطي لمستطيل بثلاثة أجزاء. كما في الشكل التالي:



الجزء العلوي يحتوي إسم الصنف، منتصف المستطيل يحتوي على خصائص الصنف (أعضاء البيانات في الـC++). الجز السفلي من المستطيل يحتوي على العمليات (دوال الأعضاء) وتكون موضوعة على شكل قائمة ، في الشكل السابق

وهو ما يمثل صنف الـ GradeBook لدينا هنا دالة عضو واحدة فقط موضوعة في هذه المستطيل ومسبوقة بإشارة (+) ، حيث تشير هذه الإشارة إلى نوع الوصول لها وهو من نوع عام public. هذا الرسم يختصر سطور عديدة ومجهود كبير في وصف الصنف.

تعريف دالة عضو بوسيط واحد

المثال الثاني

لكي نوضح ماهو الوسيط ، سوف نقوم بطرح هذا المثال. لنفرض أن دلينا كائن يمثل حساب بنكي لأحد العملاء، وأن دالة العضو هي دالة الإيداع deposit من الصنف account. عندما نريد الإيداع فإننا نحتاج إلى معلومات معينه مثل مقدار الإيداع، فعند عملية مناداة الدالة deposit سوف نرسل لها قيمة وهي مقدار الإيداع. تنسخ هذه القيمة في متغير في تعريف الدالة (ترويسة الدالة)،
هذا المتغير يعرف بالوسيط. بعدها تقوم دالة العضو بإضافة هذه المتغير لحساب الرصيد

تعريف وفحص الصنف GradeBook :

المثال التالي هو تعديل للمثال السابق

كود

```
1  // Fig. 3.3: fig03_03.cpp
2  // Define class GradeBook with a member function that takes a parameter;
3  // Create a GradeBook object and call its displayMessage function.
```

```
4  #include <iostream>
5  using std::cout;
6  using std::cin;
7  using std::endl;
8
9  #include <string> // program uses C++ standard string class
10  using std::string;
11  using std::getline;
12
13  // GradeBook class definition
14  class GradeBook
15  {
16  public:
17     // function that displays a welcome message to the GradeBook user
18     void displayMessage( string courseName )
19     {
20        cout << "Welcome to the grade book for\n" << courseName << "!"
21           << endl;
22     } // end function displayMessage
23  }; // end class GradeBook
24
25  // function main begins program execution
26  int main()
27  {
28     string nameOfCourse; // string of characters to store the course name
29     GradeBook myGradeBook; // create a GradeBook object named
myGradeBook
30
31     // prompt for and input course name
32     cout << "Please enter the course name:" << endl;
33     getline( cin, nameOfCourse ); // read a course name with blanks
34     cout << endl; // output a blank line
35
36     // call myGradeBook's displayMessage function
37     // and pass nameOfCourse as an argument
38     myGradeBook.displayMessage( nameOfCourse );
```

```
39     return 0; // indicate successful termination
40  } // end main
```

قمنا بإعادة تعريف الصنف GradeBook في السطور (١٤-٢٤) كي نستخدم وسيط يحتوي على اسم الدورة حيث تقوم الدالة displayMessage بعرضه مع رسالة الترحيب. لكي تعرض دالة العضو رسالة الترحيب مع اسم الدورة الذي يحدده المستخدم يجب أن نستخدم وسيط لذلك، وقد استخدمنا المتغير courseName كوسيط في السطر ١٨.

قبل استخدام التقنية الجديدة (الوسيط)، دعونا نرى طريقة التعامل مع الصنف الجديد في الدالة الرئيسة main.

قمنا بإنشاء متغير جديد من نوع سلسلة string في السطر ٢٨ واسميناه nameOfCourse واستخدمناه

لتخزين اسم الدورة المدخل بواسطة المستخدم. المتغير الجديد يمثل سلسلة من الرموز لأن اسم الدورة عبارة

عن سلسلة من الرموز ، مثال : " CS 100 Introduction to C++ programming ". في الواقع ، السلسلة (string) هي عبارة عن كائن مكتبة الـ C++ القياسية من صنف السلسلة " string". هذا الصنف

معرف في ملف الترويسة < string> ، والإسم string، مثل اسم cout، ينتمي إلى مساحة الأسماء std.

لكي تتم ترجمة السطر ٢٨ يجب أن نضمن السطر ٩ في البرنامج. لاحظ أن التصريح في السطر ١٠ يسمح لنا

بكتابة السطر ٢٨ بدون السابقة :: std.

في السطر ٢٩ أنشأنا الكائن myGradeBook من الصنف GradeBook. وفي السطر ٢٩ نطلب من المستخدم أن يدخل إسما للدورة من اختيارة. في السطر ٣٣ يقرأ الإسم من المستخدم ويسند للمتغير nameOfCourse

بإستخدام دالة الـ getline من المكتبة القياسية لتؤدي عملية الإدخال.

قبل أن نشرح هذا السطر من الكود (سطر ٣٣) دعونا نوضح لماذا نكتب (cin >> nameOfCourse؛)

بدلا من السطر ٣٣ لنأخذ اسم الدورة من المستخدم. ببساطة ، إسم الدورة يتكون من عدة كلمات ، بين كل إثنتين

فراغ ، فلو استخدمنا هذا السطر لتوقف التخزين عن أول فراغ وإحتوى المتغير nameOfCourse على

المقطع الأول فقط ، على سبيل المثال، لو قمنا بتخزين اسم الدورة على هذا الشكل " CS 101

Introduction to C++ programming " فإن

المتغير سوف يحوي المقطع الأول وهو CS ١٠١. أما الدالة getline، فتقوم بتخزين إسم الدورة إلى أول

عملية ضغط على المفتاح enter من لوحة المفاتيح، وبذلك يخزن اسم الدورة كاملا مع الفراغات.

في السطر ٣٣ ، الدالة getline(cin >nameOfCours)؛ تقرأ الرموز مع الفراغ من

كائن سيل الإدخال cin حتى أول رمز إدخال. لاحظ ما يلي : " أن المفتاح enter في لوحة المفاتيح

يعتبر رمز. أيضا ، أننا عندما نريد أن نستخدم الدالة getline يجب أن ندرج ملف

الترويسة < string > في البرنامج. ايضا تعتبر الـ getline أحد اسماء مساحة الأسماء std.

السطر ٣٨ ، نستدعي دالة العضو displayMessage للكائن myGradeBook ، في هذا السطر

يسمى " المتغير الممرر " argument، وهو ممرر لدالة العضو displayMessage لكي تستطيع

أن تكمل مهمتها. ففي هذه العملية تؤخذ نسخة من المتغير الممرر nameOfCourse وتوضع في

الوسيط coursName في تعريف الدالة في السطر ١٨، لاحظ أن اسم الدورة يعرض كجز من رسالة

الترحيب عند تنفيذ البرنامج.

المزيد حول المتغيرات الممررة والوسائط:

عند استخدام المتغيرات الممرة والوسائط ، يمكنك ملاحظة مايلي :

١- في السطر ١٨ ما بين الأقواس، يمكنك وضع أي عدد من الوسائط لهما إحتاجت الدالة لذلك، فما هذه

الوسائط إلا معلومات اضافية تحتاجها الدالة لإكمال مهمتها.

٢- في نفس السطر، الدالة لا تتطلب إلا وسيط واحد وهو coursName.

٣- كل وسيط يجب أن يحدد نوعة واسمه، الوسيط في هذا الحالة من نوع string " سلسلة رمزية " وامه courseName.

٤- في جسم الدالة نقوم بإستخدام هذا الوسيط في عملية الطباعة كما هو موضح في السطر٢٠ حيث يعرض

الإسم بعد رسالة الترحيب.

٥- لاحظ إسم المتغير الوسيط في السطر ١٨ يمكن أن يكون نفسه أو اسم مختلف عن اسم المتغير الممرر في

السطر ٣٨ (سوف نتعلم تفاصيل ذلك قريبا).

٦- عدد وترتيب الوسائط فيترويسة الدالة (سطر ١٨) يجب أن يكون مربوط بعدد وترتيب المتغيرات الممرره في استدعاء

الدالة (سطر ٣٨) ، كذلك النوع نفس الحالة.

خطأ برمجي شائع :

وضع فاصلة منقوطة بعد القوس الايمن بعد قائمة الوسائط في تعريف

الدالة سوف يولد خطأ نصي.

خطأ برمجي شائع :

تعريف وسائط الدالة مرة اخرى كمتغيرات محلية في الدالة يولد خطأ في الترجمة.

لممارسة برمجة جيدة:

لتجنب الغمووض ، تجنب استخدام نفس الأسماء لأسماء المتغيرات الممرره للدالة والوسائط المقابلة في تعريف الدالة .

لممارسة برمجة جيدة:

إستخدام أسماء للدوال ذات معنى وأيضا ، اسماء الوسائط، يجعل البرنامج سهل القراءة ويساعد على تجنب الإفراط في وضع التعليقات .

تحديث الـ UML بعد التعديل للصنف GradeBook:

الشكل التالي يمثل الـ UML بعد تعديل البرنامج :

التغيير فقط في المستطيل السفلي، حيث قمنا بإضافة وسيط اسمه coursName ونوعة string والنقطتان (:) وضعت للفاصل بين الوسيط ونوعة.
لاحظ أن الصنف GradeBook لا يحتوي حتى الآن أي من أعضاء البيانات (المستطيل في الوسط).

أنتهت المحاضرة

بسبب ترابط مواضيع هذا الفصل ، لا يوجد اسئلة تمارين لهذا الفصل حتى ننتهي منه في المحاضرة القادمة

وسوف تكون المحاضرة القادمة يوم السبت ، والمحاضرة التي تليها يوم الجمعة ، لترجع مواعيد المحاضرات إلى طبيعتها بعد ذلك

تم تحرير المشاركة بواسطة **BjarneStudent**: Nov 4 2007, 01:56 PM

الصور المرفقة