بسم الله الرحمن الرحيم

هذا الملف هو الأول في سلسلة تعليم الـ JDeveloper

والتي أحاول تقديمها للمطورين العرب

وهذه الدروس مترجمة من الدروس المقدمة من شركة أوراكل المنتجة لبيئة التطوير تلك

هذا الملف متاح للجميع الانتفاع به ، ولكن إذا تم نسخ أو نقل المادة أرجو ذكر المصدر للحفاظ على المجهود الذي تم فيه ، وجزاكم اله خيراً كثيراً

أرجو من الله العلي القدير أن ينفع بها

أخوكم :
م: أحمد عطاالله
dev.ahmad3ttallah@yahoo.com
http://hope-sun.blogspot.com

 جميع الحقوق محفوظة

مقدمة لبيئة تطوير JDeveloper 11g

يمكنك الاطلاع على الدرس الأصلي من موقع أوراكل من هنا .

الهدف :
هذه المجموعة من الدروس توفر جولة على المكونات الأساسية لبيئة التطوير (Oracle JDeveloper 11g IDE) ، ولنرى امكانية استخدامها لإنشاء تطبيقات مبدئية .

الوقت اللازم :
حوالي 45 دقيقة .

هذا الدرس يغطي الموضوعات التالية :
1- نظرة عامة ، شروط أساسية .
2- بدء تشغيل Oracle JDeveloper 11g
3- إنشاء أول Application
4- إنشاء أول Java Class
5- استخدام الـ Source Editor مع الـ Java Class
6-اكتشاف الأخطاء في برنامج حافا Debugging
7- إدارة الملفات

 جميع الحقوق محفوظة

نظرة عامة ، شروط أساسية

نظرة عامة :

في هذا الدرس سوف نتعلم سوياً كيفية إنشاء Java Class ، و استخدام Java Class الجديدة التي أنشأتها بنفسك ، بعد ذلك نستكشف بعض مميزات و خصائص JDeveloper IDE شاملة Code Assist و Debugger .

شروط أساسية :

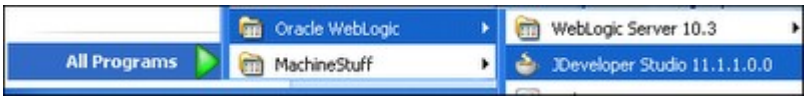
قبل أن تبدأ هذا الدرس يجب عليك أولاً تحميل Oracle JDeveloper 11g،يمكنك تحميله من موقع أوراكل من هنا .

بدء تشغيل Oracle JDeveloper 11g

1- ابدأ تشغيل Oracle JDeveloper 11g سواء من :

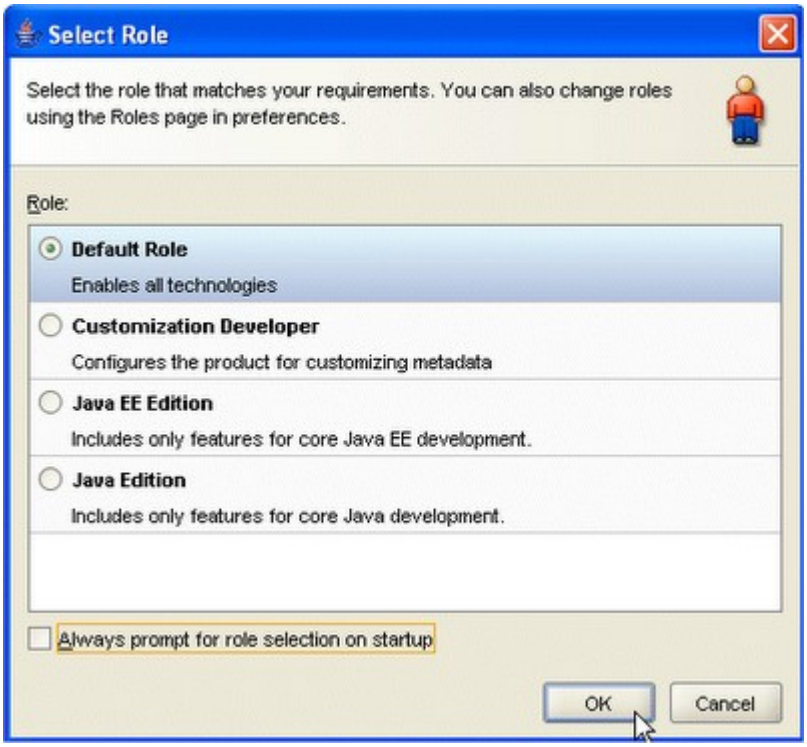
* قائمة Start

Start > All Programs > Oracle WebLogic > JDeveloper Studio 11.1.1.0.0



* أو لو كانت لديك النسخة الجاهزة فقط قم بتشغيل ملف jdeveloper.exe .

لو أن هناك رسالة ظهرت تسأل عن إذا ما كنت تريد أن تترك نسخة سابقة من JDeveloper أضغط **NO** .
2- في المربع الحواري Select Role اختار Default Role ثم اضغط OK .



ملاحظة :

التشكيل "Shaping" الذي يقوم على أساس الـ Role الذي اختاره المستخدم ، تتيح لبيئة الـ JDeveloper بتكييف نفسها .
تقوم بذلك بحيث أنها تلغي الـ Items التي لا حاجة إليها من القوائم "menus" و المفضلة "preferences" و المعرض الجديد "new ga" بالإضافة إلى الحقول الفردية في المربعات الحوارية "individual fields on dialogs".

التشكيل "Shaping" بإمكانه التحكم في القيم الافتراضية للحقول الموجودة في المربعات الحوارية المتضمنة في أداة ما .
عند أول مرة تبدأ تشغيل الـ JDeveloper ، انت مطالب باختيار role و الذي بدوره يصف كيفية استخدامك للـ JDeveloper .

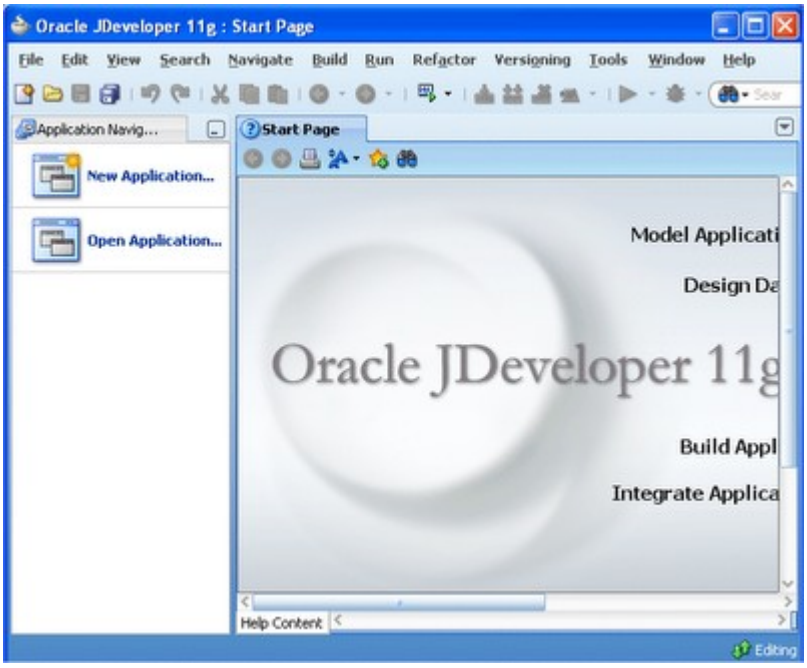
لو أنك اخترت الـ role الخاص بـ Java Developer أو CRM Applications Developer أو Default Role فستحصل على خبرات مختلفة تماماً في كل مرة .

جميع الحقوق محفوظة

1- مقدمة لبيئة تطوير JDeveloper 11g

3- بمجرد ما يتم تحميل البرنامج تظهر JDeveloper IDE . في أول مرة تفتح فيها الـ JDeveloper IDE ، تظهر شاشة البداية . يمكنك إعادة استدعاء تلك الشاشة فيما بعد من : Help Start Page .

خد بالك من الاختيارات المتعددة التي يمكنك من خلالها تعلم JDeveloper ، بعد استكشافك لتلك الخيارات أغلق صفحة البداية من X
في التبويب الخاص بها (الـ X بتظهر بس لما تحرك الماوس على التبويب بتاعها)



 جميع الحقوق محفوظة

إنشاء أول Application

الـ Application هو أعلى مستوى في الـ Control Structure ، التطبيقات يتم تخزينها بالامتداد .jws .

عندما تفتح Jdeveloper آخر تطبيق كنت تعمل عليه يفتح تلقائياً ، وبالتالي يمكنك متابعة ما كنت تقوم به من أعمال .

عند إنشائك تطبيق جديد يتيح لك الـ Jdeveloper إمكانية العمل عن طريق قالب Template - - ، القالب الذي تختاره يختار لك الإعدادات المبدئية لهيكل التطبيق الذي تقوم بإنشائه ، أي أسماء المجلدات داخل المشروع .

في الـ JDeveloper عادة تعمل على مشروع متضمن داخل تطبيق .

تطبيق الـ JDeveloper هو تجميع منطقي لمجموعة من الملفات ذات الصلة ، بإمكانك إضافة العديد من المشروعات داخل التطبيق الواحد وبالتالي يسهل عليك العمل داخل التطبيق كما يمكنك من تعديل أو إعادة استخدام الكود المصدري الخاص-Source code- بمشروعاتك .

مستكشف التطبيقات –Application navigator- هو الشاشة الرئيسية في الـ Jdeveloper والتي من خلاله يمكنك العمل على مكونات التطبيق الخاص بك .

بناء مستكشف التطبيقات عبارة عن الشكل الهرمي و الدعم الخاص بـ التطبيق و المشروعات و ملفات .html و أشياء أخرى كثيرة .

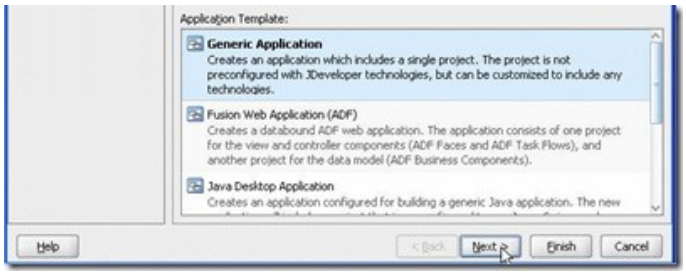
تلقائياً يكون مكان مستكشف التطبيقات في الجانب الأيسر من بيئة التطوير .

لإنشاء تطبيق اتبع الخطوات التالية :

1- اضغط على New Application في مستكشف التطبيقات .

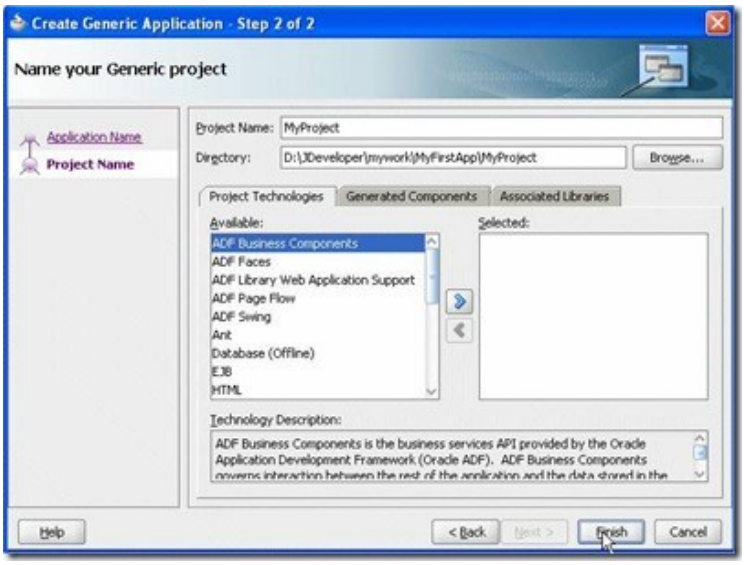
2- في المربع الحواري الذي يظهر ، عدّل الاسم الافتراضي للتطبيق إلى **MyFirstApp** ، **ملحوظة المسار -** Directory Name**- بتغير تلقائياً .**

3- اختار من قوالب التطبيقات –Application templates- القالب **Generic Application لكي يتم تعريف هذا التطبيق بمشروع واحد داخله والذي يتعامل مع كل تكنولوجيات الـ Jdeveloper .**



ثم اضغط على Next .

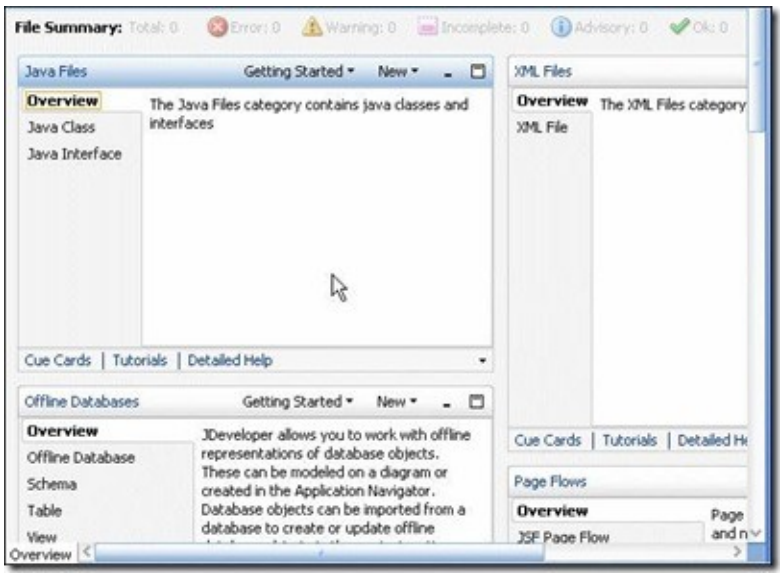
4- في هذه الخطوة الجديدة غيّر الاسم التلقائي للمشروع إلى **MyProject** ، **ثم اضغط على Finish .**



5- في مستكشف التطبيقات ، يتم عرض المشروعات و تكون في المستوى الثاني من الهيكل الهرمي تحت التطبيق . يجب أن يكون مستكشف التطبيقات كالتالي :

ملحوظة : مجلد المشروع يكون داخل مجلد التطبيق .

6- في المحرر –Visual Editor- يتم عرض التصنيفات المتعددة من الـ Jdeveloper وتحت كلّ منهم مجموعة من اللينكات التي تعرض تفاصيل ذلك التصنيف .



7- في مستكشف التطبيقات ، لاحظ الثلاث عناوين بداخله ، سنتحدث عنهم لاحقاً .

8- يمكنك فتحهم وعرض محتواهم بالضغط عليهم ضغطة واحدة ، و لغلقهم اضغط مرة أخرى .

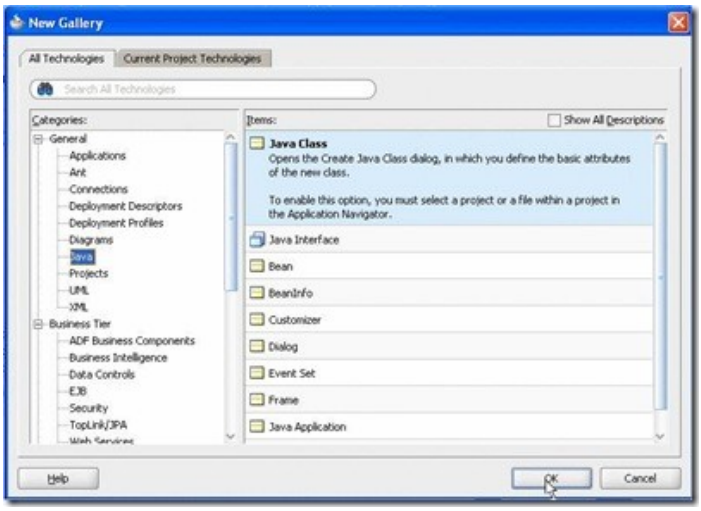
 جميع الحقوق محفوظة

إنشاء أول Java Class

لإنشاء Class جديدة اتبع الخطوات التالية :

1- اضغط بزر الماوس الأيمن على MyProject في مستكشف التطبيقات واختار New من القائمة المنسدلة .

2- تظهر لنا شاشة بعنوان New Gallery ، اضغط على علامة + و التي على يسار تصنيف General لعرض ما بها ، ومن القائمة التي تم عرضها اختار Java ، وبعد ذلك اختار Java Class من قائمة الـ Items على يمين الشاشة .

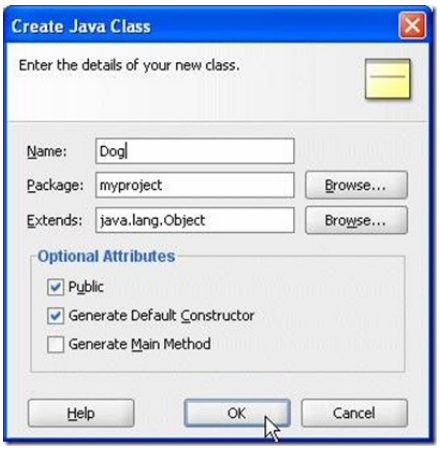


ثم اضغط على ok .

3- في المربع الحواري Create Java Class ، غيّر الاسم التلقائي إلى Dog .

لأننا لم نحدد package name ، عند إنشاء التطبيق يأخذ حقل الـ package name اسم المشروع ولكن بالحروف الصغيرة - lowercase- ، يمكنك تغيير ذلك الاسم إذا كنت ترغب ، ولكن ذلك غير مطلوب في هذا الدرس .

ثم اضغط ok .

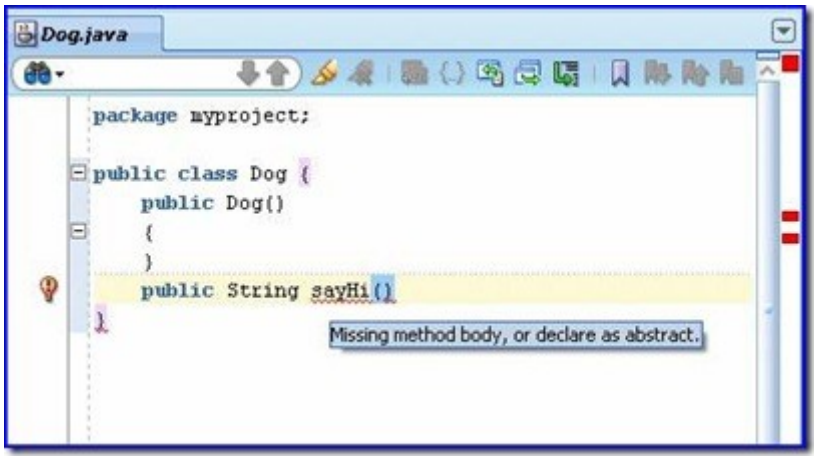


 جميع الحقوق محفوظة

4- الـ Class الجديدة يتم فتحها تلقائيا في المحرر النصي -source editor- والتي ترى من خلاله تعريف و شكل الـ Class .

أضف method إلى الـ class بعد الـ constructor ، اضغط [Enter] لإنشاء خط جديد ، ثم اكتب هذا الكود :

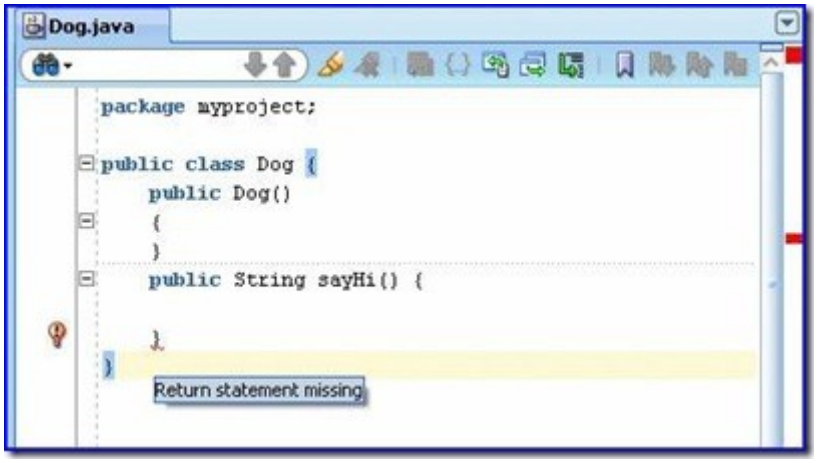
public String sayHi()



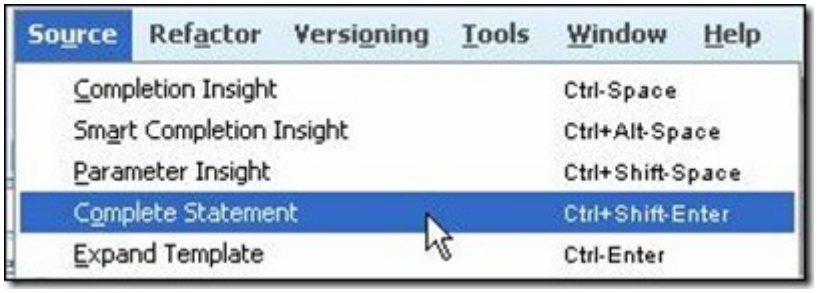
ملحوظة : الكود الذي كتبته للتو سيكون تحته خط أحمر مزجزج موضحاً أن هناك مشكلة في الـ syntax ، كما ان هناك مربعات حمراء في الجانب الأيمن من شاشة محرر الكود ، إذا مررت مؤشر الماوس على هذه المربعات أو الخط الاحمر المزجزج ، يظهر شريط منسدل يوضح معلومات عن طبيعة الخطأ أو الأخطاء .

5- في نهاية الخط اضغط على [Ctrl]+[Shift] [Enter] وسوف يقوم الـ Jdeveloper تلقائياً بإكمال هيكل الـ method ، وسوف تلاحظ بعض الأخطاء أو المشكلات التي كانت موجودة قد اختفت .

وطالما أن هناك علامات حمراء توضح أن هناك خطأ ما فإن الـ syntax يكون به شيء ناقص .



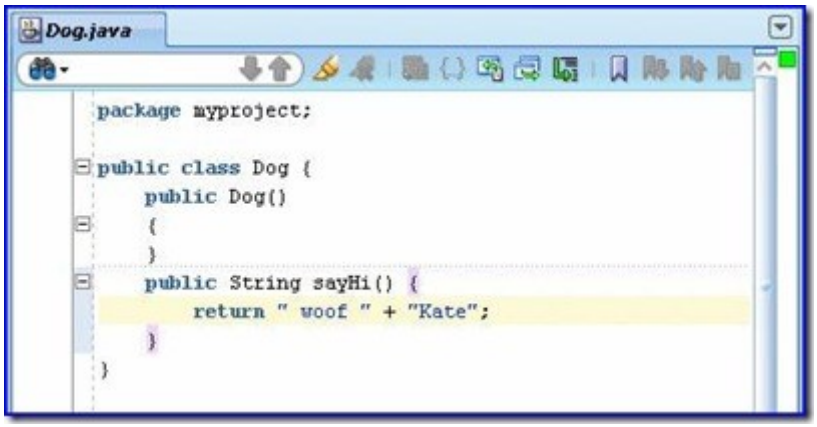
ملحوظة : يمكنك تفعيل التكملة التلقائية من قائمة Source | Complete Statement



 جميع الحقوق محفوظة

6- أضف هذا الكود داخل الأقواس الخاصة بالـ method :

"return " woof " + "Kate";



ملحوظة : عند كتابة علامتي التنصيص تجد أن الـ Jdeveloper يكتب علامتي التنصيص الأخرى تلقائياً ، كما يقوم بوضع مؤشر الكتابة -cursor – بينهما ليسهل عليك كتابة النص مباشرة ، لاحظ أيضاً ظهور مربع أخضر يوضح عدم وجود أخطاء .

 جميع الحقوق محفوظة

استخدام الـ Source Editor في الـ Java Class

المحرر هو أكثر الأماكن التي من خلالها يتم تطوير العمل الخاص بالتطبيق ، فيها يمكنك كتابة الكود و تصمم واجهة المستخدم –User interface- .

في هذا الجزء سوف نكتشف خصائص Java Source Editor .

1- استخدام قوالب الأكواد -code templates-.

2- تجميع -Compiling- و تشغيل -Running- Java class .

3- العمل مع مساعد الكود -Code Assist- .

4- البحث في الأكواد .

5- إعادة تصنيع الكود .

6- عرض تاريخ التعديلات على الكود .

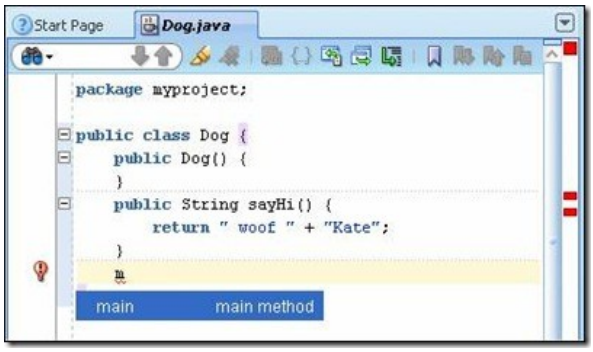
7- الإبحار في الكود .

استخدام قوالب الأكواد

قوالب الأكواد تساعدك على كتابة الكود بسرعة وبكفاءة بدون أخطاء عندما تكون في محرر الكود . يمكنك تعديل قالب موجود بالفعل أو عمل واحد خاص بك . في هذا المقال نعرض كيفية استخدام بعض هذه القوالب في Dog class .

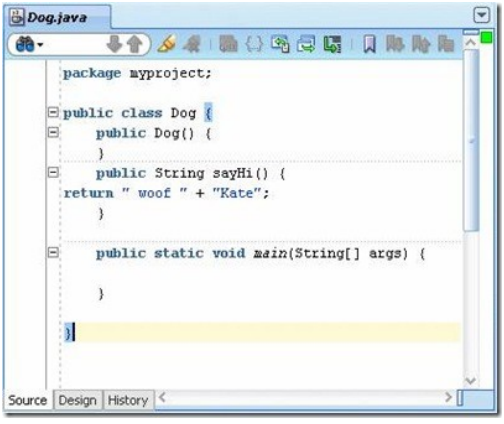
1- هناك قالب لإضافة الـ main method في الـ Java class .

اضغط **[Enter] لعمل سطر جديد بعد** sayHi() method . اكتب الحرف m ، اضغط على **[Ctrl]+[Enter] لاستعداء قوالب الأكواد .**



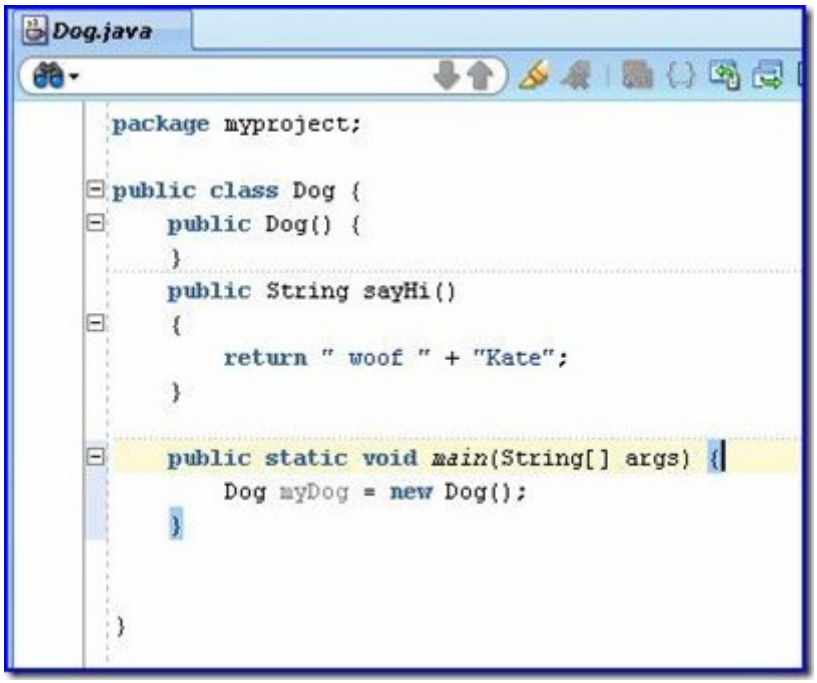
سوف يقترح عليك قالب الـ main method .

2- اضغط Enter مرة أخرى لقبول هذا القالب المقترح و إدراجه .

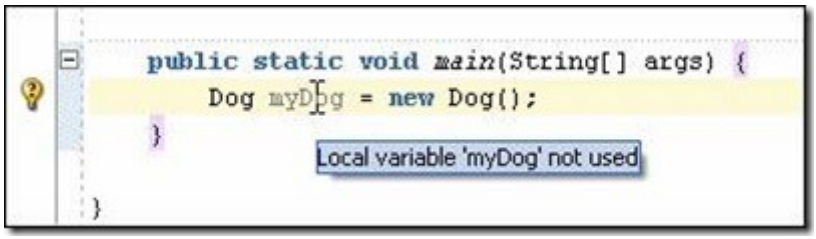


3- أضف object **Dog** في الـ Main method بكتابة هذا الكود :

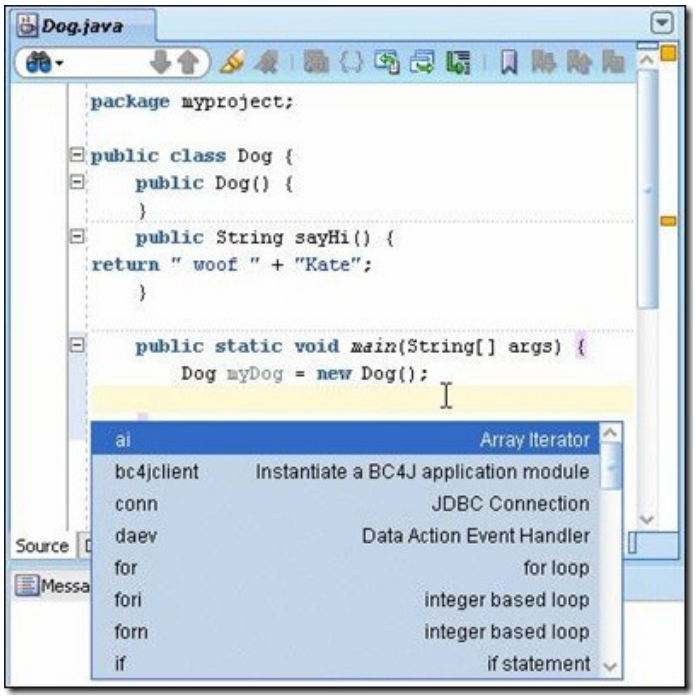
Dog myDog = new Dog();



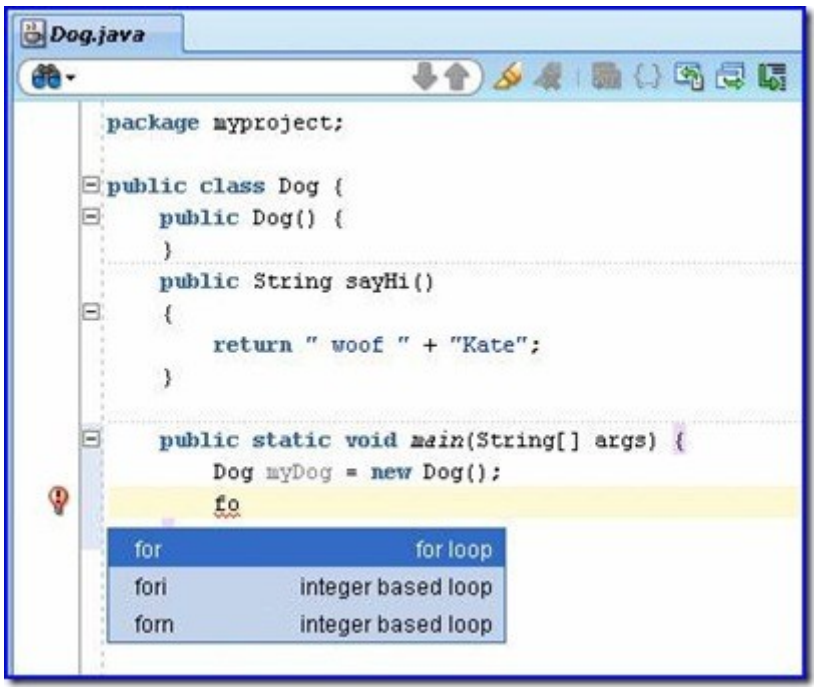
إذا حركت الماوس على المتغير myDog سوف ينسدل شريط يوضح أن هذا المتغير لم يتم استخدامه بعد .



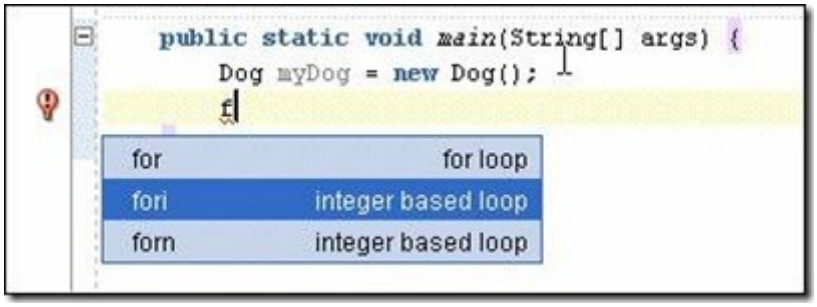
4- أضف سطر جديد ثم اضغط **[Ctrl]+[Enter] لترى القوالب المتاحة .**



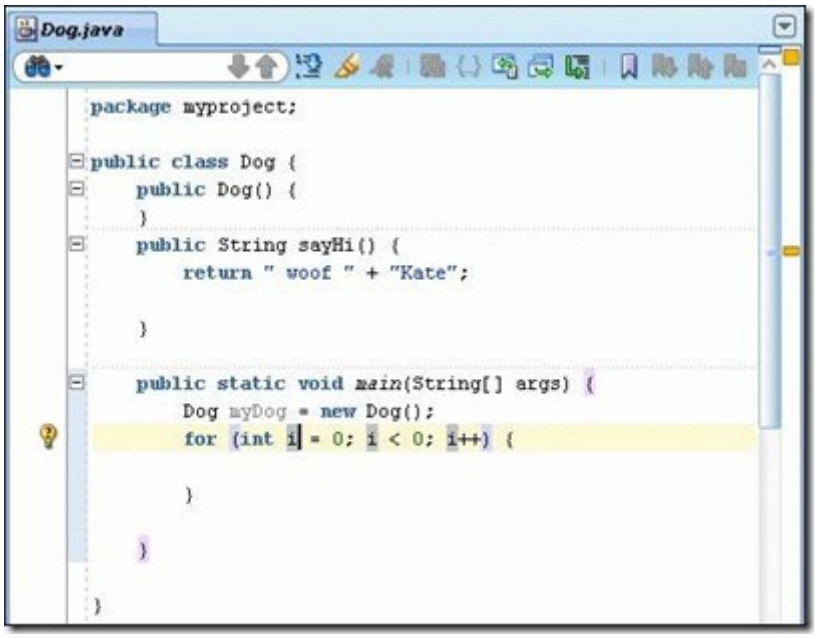
5- سوف ننشئ for loop ، أكتب fo لتحديد القوالب التي تبدأ بتلك الحروف،سوف تجد ثلاث قوالب تم اقتراحهم .



6- اضغط ضغطتين مزدوجتين على القالب الثاني -**fori integer based loop**- لاختياره .

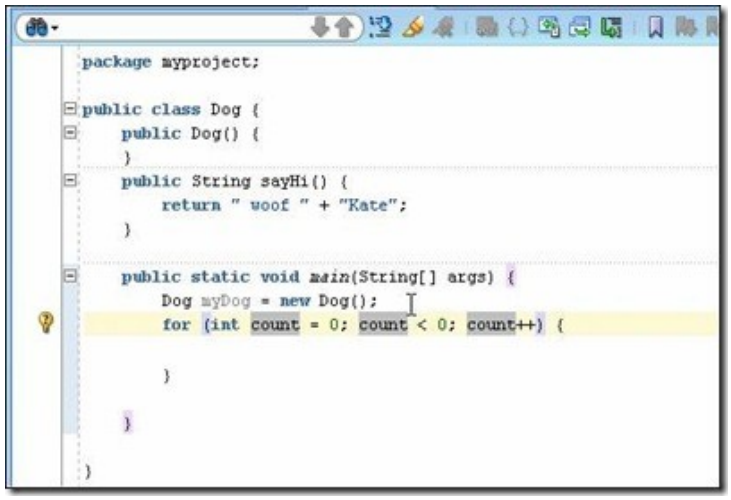


ستجد أن القالب تم إدراجه في الكود .

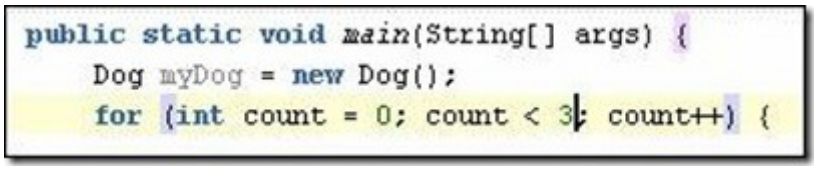


7- عدّل كود القالب .

استبدل i بـ count . لاحظ أن تغيير أول i يتبعه تغيير كل الـ i في باقي الكود .

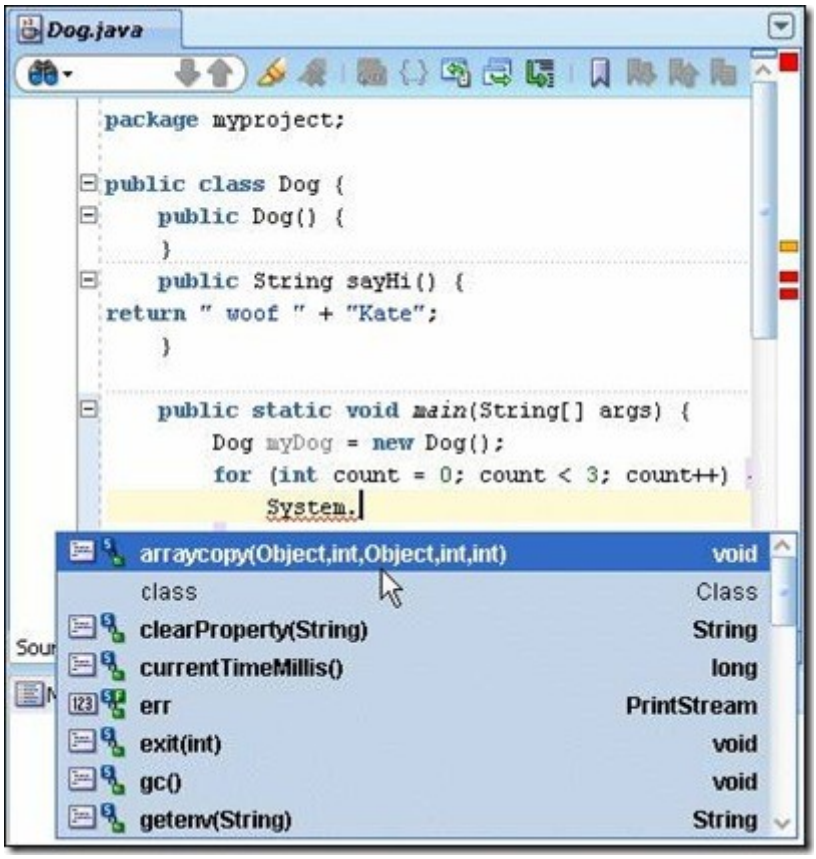


حدد الـ loop لتكون ثلاث مرات .

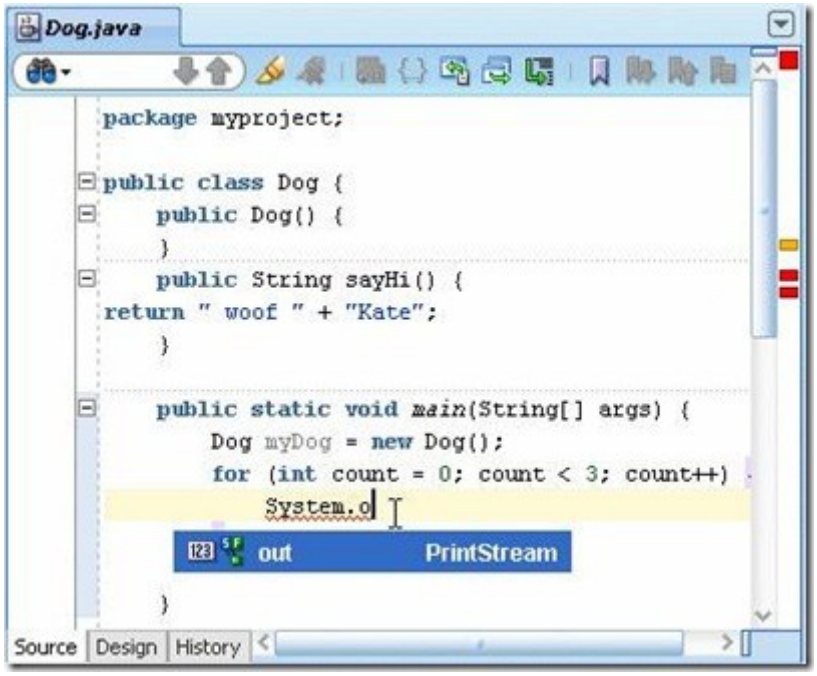


8- أدرج جملة System.out.println .

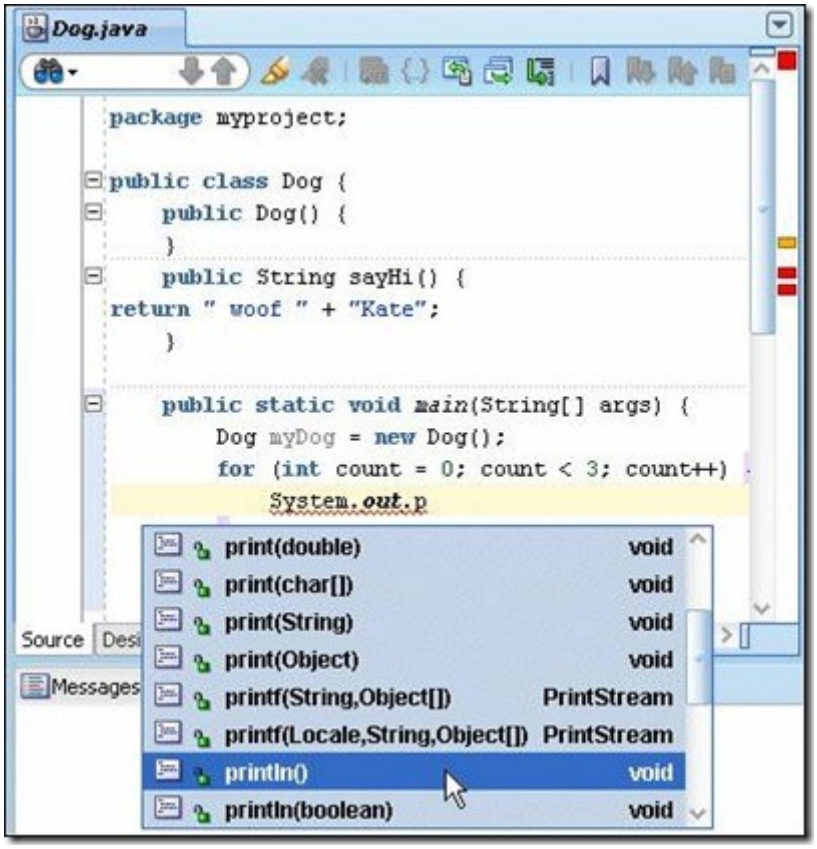
اجعل مؤشر الكتابة -cursor- في الفراغ بين أقواس الـ loop ، ثم اكتب **System. (تأكد من كتابة النقطة بعد الكلمة) ، سوف تظهر قائمة ببعض الأكواد المقترحة .**



اكتب الحرف o ثم اضغط Enter لاختيار الكود المقترح out .



اكتب . (نقطة) ، وعندما تظهر قائمة الأكواد المقترحة اكتب الحرف p ثم اضغط بالماوس ضغطتين على الكود المقترح **println()** .



ملحوظة : وسيلة يهلة لكتابة جملة System.out.println() أن تكتب sop ثم تضغط **[Ctrl]+[Enter] .**

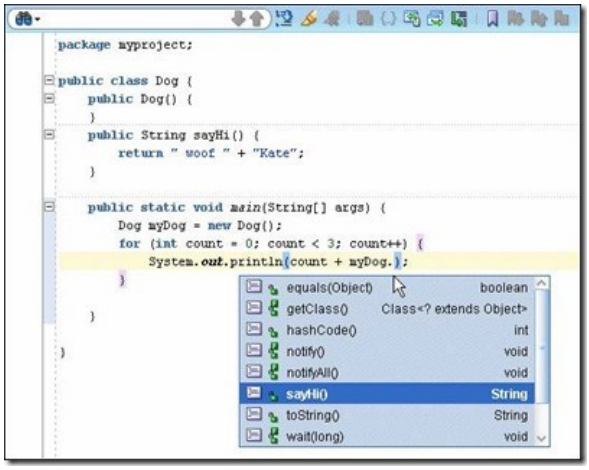
التكنيك السابق كان لتعريفك بخاصية التكملة التلقائية للكود .

 جميع الحقوق محفوظة

9- أضف الكود لاستخدام الـ loop لعرض sayHi message.

داخل الأقواس بعد كلمة println أكتب myDog.sayHi + count() .

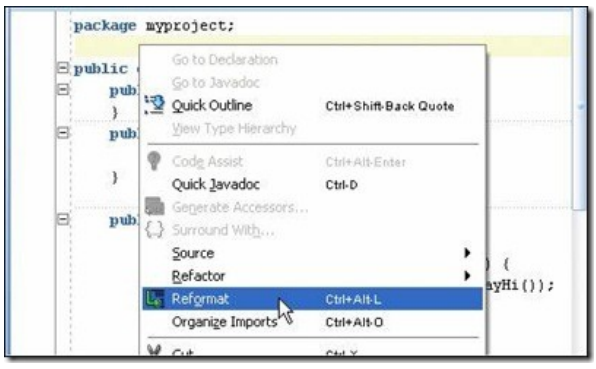
ابدأ بكتابة **myDog + count. ثم اختار sayHi** method من القائمة .



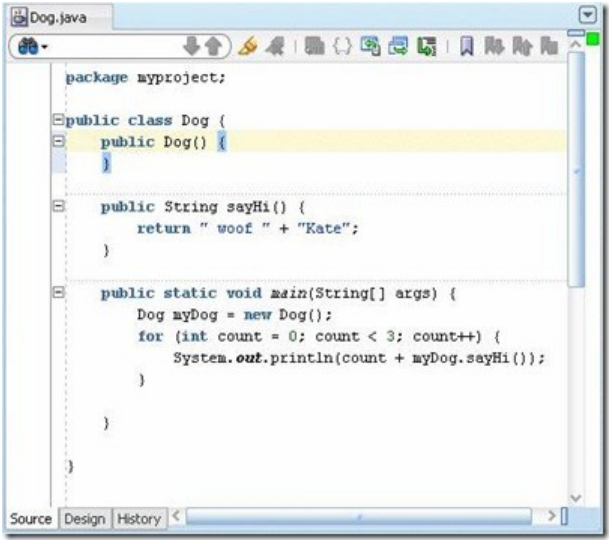
يجب أن يكون السطر كالتالي :

System.out.println(count + myDog.sayHi());

10- اضغط بزر الماوس الأيمن في أي مكان فارغ من محرر الكود ثم اختار **Reformat لكي يقوم Jdeveloper بتهيئة الكود (عملية تنسيق للكود لا أكثر) .**



11- يجب أن يكون الكود كالتالي :



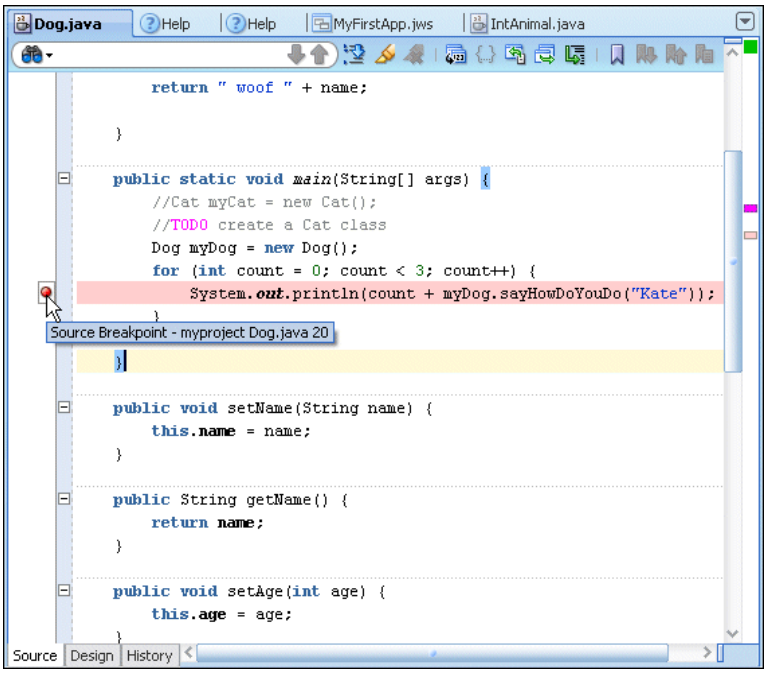
12- قم بتخزين عملك ، اذهب إلى **File-->Save All أو** Save All button في شريط الأدوات .

اكتشاف الأخطاء في برنامج جافا (Debugging)

مكتشف الأخطاء - أو مدقق الأخطاء - المتضمّن في الـ JDeveloper يتيح لك إمكانية اكتشاف الأخطاء و أنت في محرر الكود .
في هذا المقال سيتم عرض كيفية التحكم في مسار تنفيذ و تشغيل برنامج بوضع نقاط التوقف **(breakpoints)** .

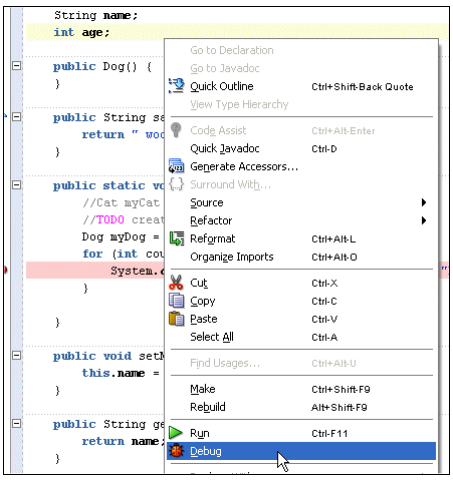
عند تنفيذ برنامج ووجود نقاط توقف **(breakpoints)** , البرنامج يتوقف , والممدقق يقوم يعرض السطر الذي تم عمل نقطة توقف عنده في محرر الكود .
باستطاعتك بعد ذلك استخدام المدقق لعرض حالة البرنامج .
1- أنشيء نقطة توقف في ملف Dog.java .لكي تفعل ذلك ,اضغط على الهامش **(margin)** على يسار السطر :

;(("System.out.println(count + myDog.sayHowDoYouDo("Kate

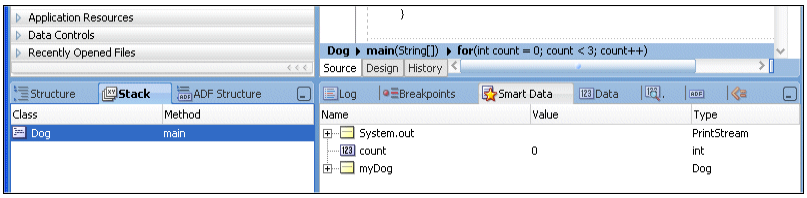


الأيقونة التي تعبر عن نقطة التوقف تجد أنها ظهرت في الهامش ، على يسار السطر الموضح أعلاه .

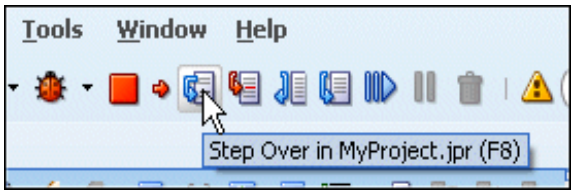
2- اضغط بزرار الماوس الأيمن في أي مكان فارغ من محرر الكود ثم اختار Debug من القائمة التي تظهر .



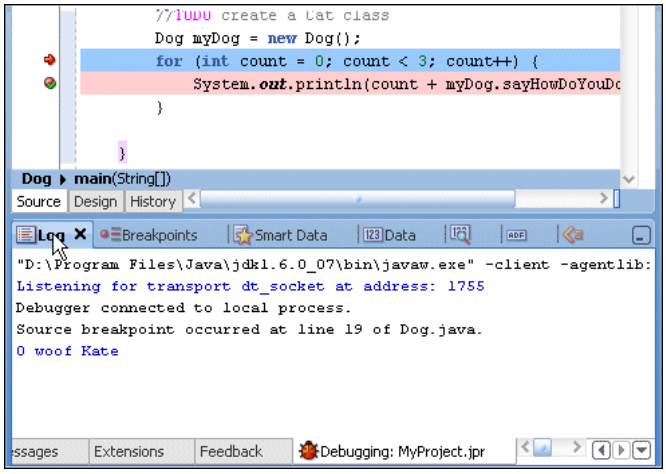
3- يتم تنفيذ البرنامج إلى أن يصل إلى نقطة التوقف . السهم الأحمر الذي يظهر في الهامش الأيسر من السطر يوضح أين توقف تنفيذ البرنامج .
ثم تفتح شاشة المدقق (debugger) وتعرض تتبع الأخطاء .



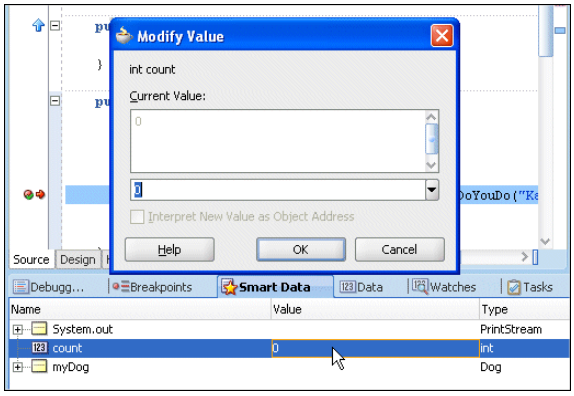
4- اضغط على أيقونة **Step Over** في شريط الأدوات بالأعلى لكي تنفذ أول عملية من عمليات التكرار بالـ myDog.sayHowDoYouDo() method .



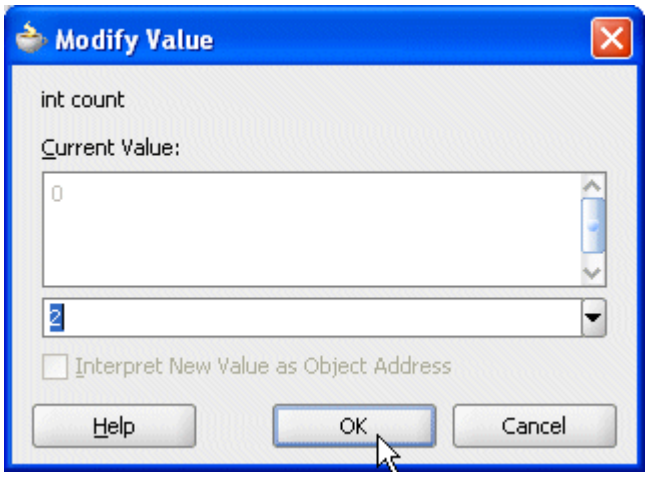
5- اضغط على تبويب **Debugging: MyProject.jpr** في أسفل شاشة الـ log ثم اضغط على تبويب **log** في شاشة المدقق .
ملحوظة : شاشة الـ log تعرض أول رسال من *woof Kate* .



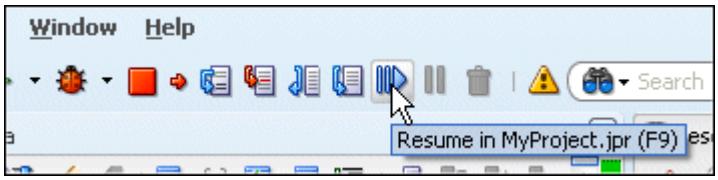
6- لاحظ شاشة **Smart Data** و التي على يمين تبويب Debugging Log . اختار المتغير **count** ثم اضغط مرتين على عمود **Value** لعرض شاشة **Modify Value** .



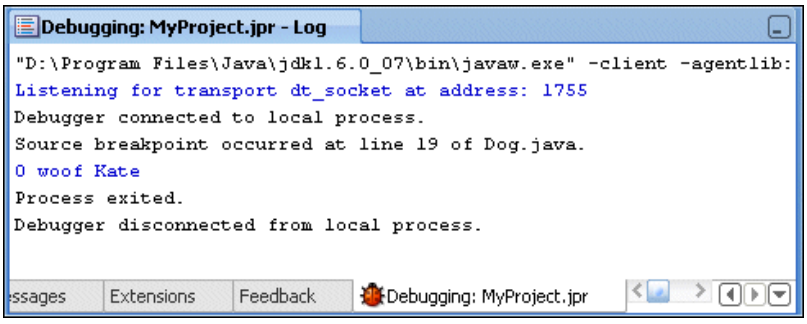
7- أكتب 2 كقيمة جديدة لذلك المتغير ، ثم اضغط **ok** .



8- في شريط الأدوات اضغط على زرار **Resume** لمتابعة عملية تنفيذ البرنامج .



9- المتغير count تمت زيادته وتعدّى القيمة المسموحة له ، لذلك فإن البرنامج سيغلق و المدقق سوف يغلق أيض (disconnects)آ .

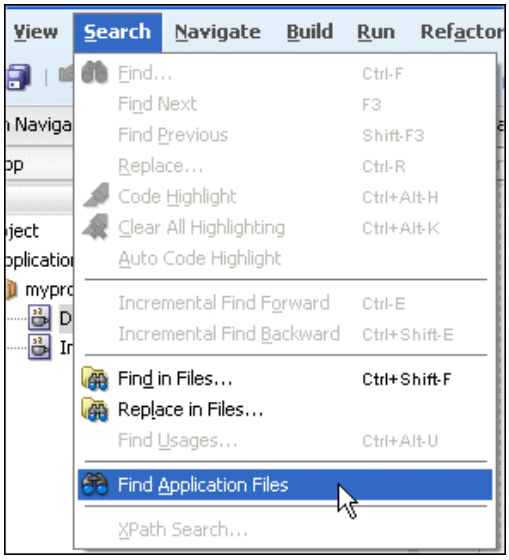


 جميع الحقوق محفوظة

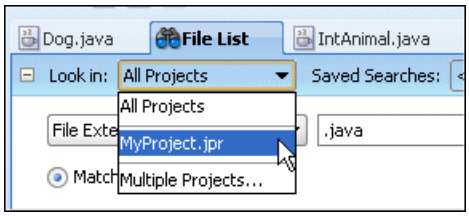
إدارة الملفات

بيئة التطوير JDeveloper تتيح البحث في الملفات التي تنتمي للتطبيق الذي تعمل فيه أو خلال أكثر من تطبيق ، لكي نختبر تلك الخاصية ، اتبع الخطوات التالية :

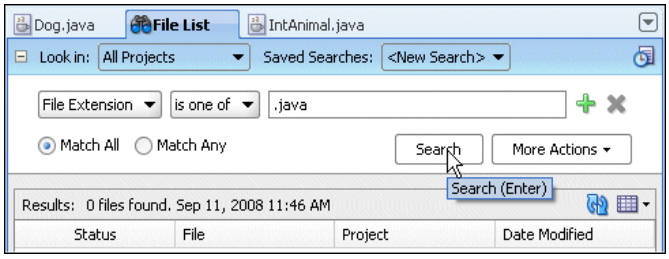
1- من شريط القوائم اختار **Search | Find Application Files** .



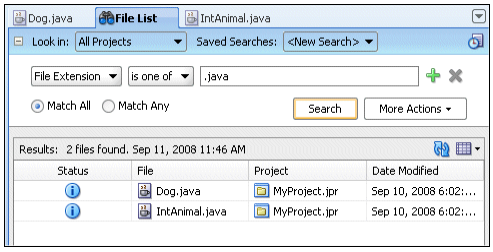
2- في تبويب **File List** استخدم حقل **Look in** لاختيار ما ستبحث فيه ، اتركه على الاختيار **All Projects** .



3- اختار **File Extension** واكتب **java.** لاستدعاء كل الملفات من هذا النوع ، ثم اضغط **Search** .



4- شاشة النتائج سوف تعرض الملفات ذات النوع الذي بحثت عنه .

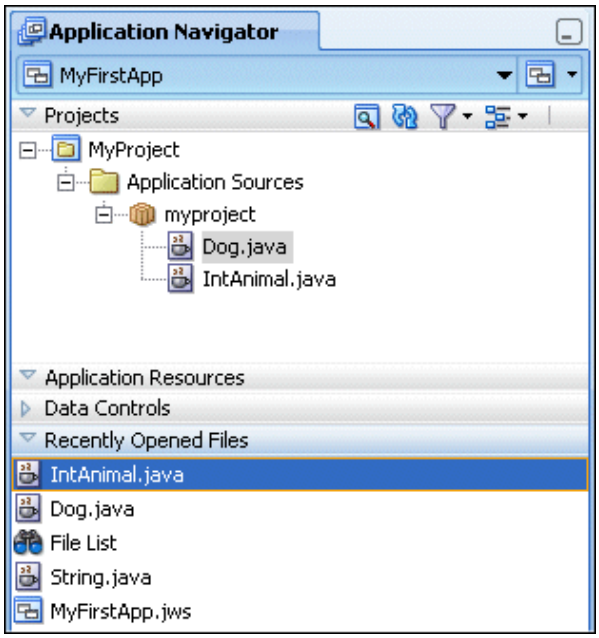


ملحوظة : إذا تم الضغط على أحد تلك الملفات سيتم فتحه في محرر الكود .

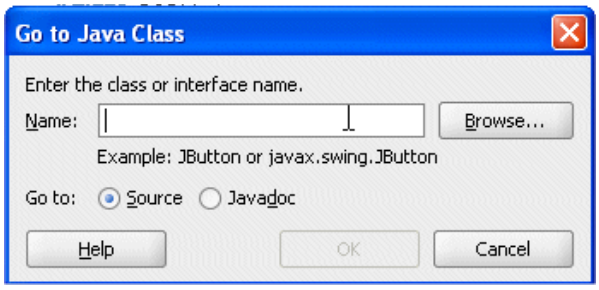
 جميع الحقوق محفوظة

5- يمكنك بحث معقد عن طريق استخدام تلك العلامة .

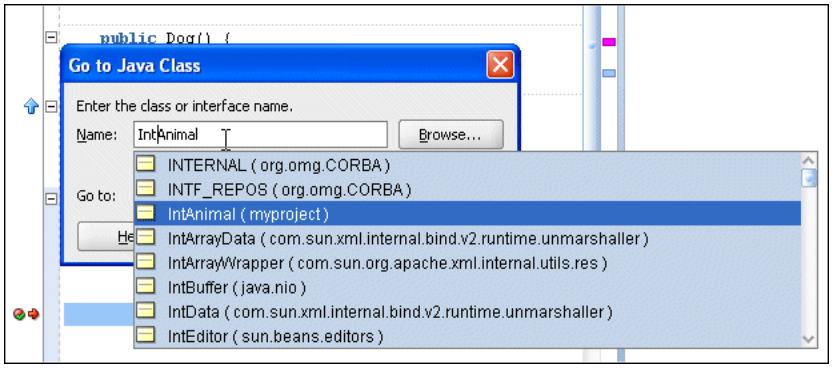
6- كما يمكنك استعادة التي تم استخدامها حديثاً من مستكشف التطبيقات ، اضغط على **Recently Opened Files** وسيفتح لك القائمة المنسدلة .



7- لاستدعاء java class اضغط على **[Ctrl] + [-]** لكي يفتح لك الشاشة التالية .

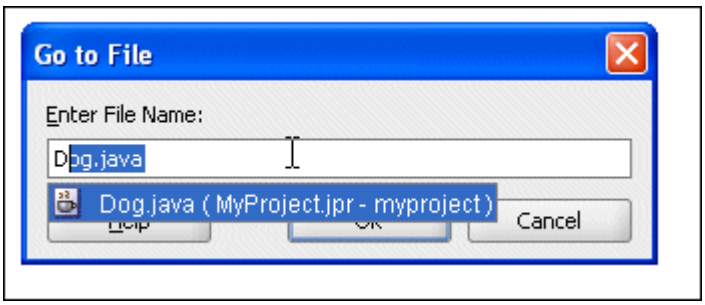


8- في حقل الاسم (Name) اكتب **Int** ، ثم اختار **IntAnimal.java** من القائمة ثم **Enter** .



 جميع الحقوق محفوظة

9- لاستدعاء ملف ، اضغط **[Alt]** + **[Ctrl]** + **[-]** لكي يفتح لك الشاشة التالية ، وفي حقل اسم الملف (File Name) اكتب **D** ثم اختار **Dog.java** من القائمة ثم **Enter** .



ملخّص

إلى هنا يمكننا القول أننا انتهينا من تقديم جولة مبدئية لبيئة التطوير JDeveloper , أنشأنا تطبيق ، مشروع و Java class ,
ثم استخدمنا الـ class لنكتشف عدداً من الخصائص لبيئة التطوير تلك الخاصة بالجافا و التي شملت قوالب الأكواد Code Templates ، مساعد الكود Code Assist ، إعادة التصنيع Refactoring و عرض تاريخ تعديلات الكود ، في النهاية استطعنا أن ندقق الأخطاء في برنامجنا باستخدام المدقق Debugger المتضمن في بيئة التطوير JDeveloper .
لقد تعلّمنا :
1- إنشاء تطبيق و مشروع جديدين في الـ JDeveloper .
2- إنشاء Java class .
3- الاستفادة من الخصائص المساعدة و في الـ Jaava IDE .
4- استخدام مدقق الكود .
5- استدعاء الملفات .

بعد الانتهاء من ذلك كله أرجو من الله العليّ القدير أن يأجرني على هذا العمل و ينفع به الكثير من أمتنا .

 جميع الحقوق محفوظة