

جامعة دمياط
كلية التربية النوعية
قسم إعداد معلم الحاسب الآلي

محاضرات في
تطبيقات الحاسب الآلي في التعليم (١)
(برمجة الجافا - الجزء الأول)

مقدمة

عزيزي الطالب مرحبا بك في هذه المحاضرات التي تتناول مبادئ برمجة الجافا لما لتلك اللغة من تأثير كبير في مجال برمجة المواقع بصفة عامة ، وبرمجة المواقع التعليمية بصفة خاصة

ولتحقيق الأهداف المرجوة من تلك المحاضرات يجب على الطالب معرفة ما يلي:

- ١-التعامل مع التطبيقات العملية الموجودة بالمحاضرات.
- ٢- دور أستاذ المادة هو تيسير عملية التعلم وتوجيه الطالب إلى مصادر العلم والمعرفة وتبصيره بكل ما هو جديد في ذلك المجال.
- ٣- على الطالب أن يتفاعل بشكل ايجابي مع أستاذ المادة ومع مصادر المعرفة التي يوجهه إليها وعدم جعل نفسه مجرد مستقبل سلبي لما يفرض عليه وإبداء رأيه بكل وضوح وشفافية حول موضوعات المقرر ومناقشة أستاذ المادة عند لقائه به والمساعدة في تحقيق جودة التعليم.
- ٤- على الطالب أن يقوم بربط موضوعات كل فصل بما يسبقه وبما يلحقه من فصول ليكون المقرر متكاملًا مما يسهل الفهم وتحقيق تراكم المعرفة واستمرارها.
- ٥- ضرورة أن يحرص كل طالب على التواصل مع أستاذ المادة للإجابة على استفساراته حول ما لم يتمكن من فهمه بالمحاضرات

أرجو من الله أن يقلل هذا العمل، وأن يساعد في تمهيد وتنوير الطريق كما أرجو المغفرة لما به من نقص
الله اسأل أن ينفعنا بما علمنا وأن يعلمنا ما ينفعنا انه سميع الدعاء.

الفصل الأول

١

مقدمة الى لغة الجافا

الفصل الأول: مقدمة إلى لغة الجافا

الفصل الأول: مقدمة إلى لغة الجافا

مقدمة

الجافا هي لغة برمجة أنتجت عام ١٩٩٥ بواسطة شركة Sun Microsystems من خلال كل من James Gosling ، Henry McGilton وفريق عمل يضم آلاف الأشخاص بشركة Sun

وسميت بـ java بعد اجتماع James Gosling مع فريق العمل في أحد المقاهي فتوارد إلى أذهانهم اسم Java وهي القهوة المشهورة فسميت بهذا الاسم (شعار java هو فنجان القهوة) وذلك بهدف تطوير برامج لأجهزة فك التشفير والأجهزة المحمولة ولكن أصبحت فيما بعد شائعة لبناء تطبيقات الويب والسيرفرات وسطح المكتب والهواتف والروبوتات وتعمل على جميع أنظمة التشغيل : windows-Android-Unix-Linux-OS-Mac

وتناسب لغة java تطبيقات الانترنت (فهي تمثل قلب برمجة صفحات الويب) حيث تستخدم لإضافة المزايا الديناميكية لصفحات الويب مثل الرسوم المتحركة وأسعار البورصة الفورية وغيرها من المزايا الديناميكية

كما تتيح لغة Java كتابة برمجيات Applets وهي تطبيقات صغيرة يمكن إرسالها من الخادم server إلى برنامج مستعرض الانترنت الذي يقوم بفك شفرتها وتنفيذها بواسطة ما يسمى بآلة الجافا الافتراضية Java virtual machine - JVM والتي تكون مدمجة به أو تضاف إليه ، وينبغي لآلة الجافا الافتراضية أن تكون متوافقة مع المنصة platform التي تعمل عليها، أما برمجيات الجافا فيمكن تنفيذها على أي منصة platform تحتوي على آلة الجافا الافتراضية، سواء كانت ماكنتوش أو ويندوز أو غيرهما ولذلك توصف لغة الجافا بأنها مستقلة عن المنصة platform independent

وتوفر آلة الجافا الافتراضية تدابير أمنية لحماية موارد وبيانات الحاسب الذي يستضيفها من احتمالات العبث والتخريب، كما توفر معظم برامج التصفح شائعة الاستخدام الدعم للغة الجافا، وتعمل عديد من الشركات على إنتاج أدوات تطوير خاصة بلغة الجافا

وللتعرف على الطريقة التي طورت بها لغة الجافا في أوائل التسعينيات من القرن العشرين ١٩٩٠ اخترعت شركة Sun لغة الجافا حيث أن الشركة كلفت أحد المهندسين بوضع برامج لتشغيل الأجهزة التطبيقية الذكية مثل التلفزيون التفاعلي باستخدام لغة ++C وحينها وجد ذلك الشخص صعوبة في التعامل مع هذه اللغة لذا قام هو وفريق العمل المساعد له بتطوير هذه اللغة فولدت لغة جديدة تتوافق مع احتياجاته فكانت لغة الجافا، وقد خططت شركة Sun في تلك الفترة لاستغلال هذه اللغة الوليدة في التلفزيون التفاعلي وحدث نوع من البطء في مشروع التلفزيون التفاعلي ونتيجة لذلك فكرت شركة Sun في إيقاف مشروع تطوير هذه اللغة

ولكن حدث في تلك الفترة انتشار شبكة الانترنت بسرعة مذهلة مع انتشار نظام Windows ، وحيث أن لغة الجافا -التي اخترعت أصلاً لبرمجة الأجهزة التطبيقية- فيها من السمات ما يجعلها أكثر توافقاً مع شبكة الانترنت فقد كان لها السبق وأضافت الكثير إلى الإنترنت الذي كان قبلها مقصوراً على تبادل البيانات، ولكن المطورين بشركة Sun ابتكروا طريقة تجعل برامج الجافا تعمل بسهولة في صفحات الإنترنت وغيروا الاسم الذي كان قد أطلقه عليه مبتكرها ليصبح Java ومن هنا أصبحت الجافا مرتبطة في شهرتها بالإنترنت حيث أن برنامج الجافا صغير يوضع في صفحة من صفحات موقع على شبكة الانترنت يراه الملايين في جميع أنحاء العالم في نفس الوقت وقد كان هذا لا يتوفر الا مع Java مما أعطاها شهرة واسعة

الفصل الأول: مقدمة إلى لغة الجافا

ولقد أكدت لغة Java نفسها في المجال الذي طورت له أصلا فقد بدأ الآن التلفزيون التفاعلي في الانتشار والمشاهدة حسب الطلب وليس هذا فقط بل انتشر ما هو أكثر فائدة لشركة Sun وهو الهاتف المحمول حيث أن للغة الجافا دور كبير في إعداد البرامج التي يعمل بها في أجياله السابقة واللاحقة ولا نستغرب أن يحدث نوع من المنافسة بين شركة Microsoft ، وشركة Sun مما دفع Microsoft إلى حذف الآلة التخليقية للجافا JVM- Java Virtual Machine من الإصدارات الأولى Windows XP وهذه الآلة الافتراضية مسنولة عن عرض برامج java على الانترنت ولكن Microsoft تراجعت أمام طلب ملايين المستخدمين حول العالم فوضعتها مرة ثانية في الإصدارات اللاحقة وقد كانت قضية مشهورة ، وقد ساهم في شهرة الجافا أيضا برامجها التفاعلية التي تصلح لمعظم المناهج التعليمية في جميع مراحل التعليم وبالتالي فإن لها دورا كبيرا في التعليم الالكتروني والتعليم عن بعد والفصول الافتراضية

وتعد لغة الجافا من اللغات متعددة الأغراض ومتعددة المنصات تصلح لعدد كبير من التطبيقات، ومترجم جافا يقوم بإنتاج ملفات في شكل Byte code وهو يختلف عن الملفات التنفيذية التي تنتج عن لغات البرمجة الأخرى مثل C

وتحتاج البرامج المكتوبة بلغة الجافا إلى منصة في أنظمة التشغيل المختلفة لتتمكن برامجها من العمل في هذه الأنظمة وهذه المنصة تسمى آلة الجافا الافتراضية Java Virtual Machine. أو إختصارا بـ Java Run-time (JVM) ، وتتوفر هذه المنصة في عدد كبير من أنظمة التشغيل، وقبل تشغيل برنامج جافا لابد من التأكد من وجودها وكل نظام تشغيل يحتاج لآلة افتراضية خاصة به، مثلا نظام Windows 32 Bit يحتاج لآلة افتراضية مخصصة لـ Windows 32 Bit، Windows 64 Bit يحتاج لآلة افتراضية 64 Bit

مثال:

ملف تثبيت آلة جافا الافتراضية لنظام Windows 64 Bit : (jdk6u16windowsx64.exe)

وعند إنتاج برامج جافا يمكن تشغيلها على أي نظام تشغيل مباشرة عند وجود الآلة الافتراضية المناسبة، ولا يحتاج البرنامج لإعادة ترجمة حتى يعمل في أنظمة غير النظام الذي تم تطوير البرنامج فيه، مثلا يمكن تطوير برنامج جافا في بيئة Linux لإنتاج برامج يتم نقلها وتشغيلها مباشرة في Windows أو Macintosh ، وتختلف عنها لغة C في أنها تحتاج لإعادة ترجمة البرامج مرة أخرى في كل نظام تشغيل على حدة قبل تشغيل تلك البرامج ، لكن برامج لغة C لا تحتاج لآلة افتراضية في أنظمة التشغيل بل تتعامل مع نظام التشغيل مباشرة.

الفرق بين لغة Java ولغة JavaScript

على الرغم من تشابه الأسماء والذي يؤدي إلى خلط كثير من المبتدئين بين هاتين اللغتين إلا أن لغة Java (من إنتاج شركة Sun) تختلف اختلافاً كبيراً عن لغة Java Script (من إنتاج شركتي Sun Microsystems، Netscape) والتي سميت الجافا سكريبت لسبب تجاري وتسويقي يتعلق بقوة وانتشار لغة Java

فالجافا سكريبت لغة بسيطة جداً وتعد لغة Client Side أي يتم تنفيذها داخل مستعرض الويب Web Browser فالكود يرسل من الخادم Server بدون معالجة ويعالج على جهاز المستخدم client حيث أن المفسر interpreter الذي ينفذ تعليماتها يكون مدمجاً في المتصفح ، وهي قليلة الاستخدام لأنها تعتمد على المتصفح وإمكاناته.. الخ، ولا يمكن استخدامها لإنتاج برامج ضخمة وتشبه لغة ترميز النص الفائق الديناميكية DHTML - Dynamic Hyper Text Markup Language

وتستخدم لغة الجافا سكريبت لصناعة شينين : مؤثرات الجافا سكريبت ومؤثرات DHTML، فالجافا سكريبت هي نصوص تعمل وحدها داخل الصفحة لإحداث التأثيرات مثل HTML ، أما DHTML فهي تتفاعل مع المتصفح وتستخدم إمكاناته لإحداث التأثيرات مما يجعلها أقوى وأكثر تعقيداً وحجماً وإن كانت لا تصل إلى قوة الجافا أبلت

أنواع برامج java

الجافا هي لغة برمجة موجهة للكائنات Object Oriented programming، ابتكرت لتشغيل الأجهزة التطبيقية الذكية وهي بالأصل تطويراً للغة C++ ، ولا يوجد أنواع للغة الجافا وإنما توجد عدة إصدارات للغة وهي :

١- النوع الأول يطلق عليها Applet وهي برامج نستخدمها في المتصفح ولا تخلق صفحة ويب منها (ذلك النوع هو الأكثر انتشاراً ويمكن بسهولة تحويله إلى النوع الثاني Application) ولغة الجافا هي المستخدمة لصناعة الجافا أبلت java applet

٢- النوع الثاني يطلق عليه Application وهي برامج تعمل بدون الحاجة لمتصفح وتعمل مثل أي برامج أخرى

ملحوظة

لا يوجد فرق بين النوع الأول والثاني من الناحية البرمجية فكلاهما يتبع نفس القواعد ولكن عند كتابة النوع الثاني يجب علينا إنشاء نوافذ وقوائم.. الخ مما يزيد من حجم البرنامج، ولكن في النوع الأول فإن ذلك متوفر لنا عن طريق المتصفح

إصدارات لغة Java

يوجد إصدارات للغة Java وهي: J2SE ، J2ME (تخص أجهزة الهاتف المحمول والأجهزة اللاسلكية Wireless Devices) ، الإصدار J2EE هو اختصار لـ Java 2 Enterprise Edition وهي تزودنا بالتطبيقات الكبيرة على مستوى الشركات الكبيرة

(Java Standard Edition) Java SE

الإصدار القياسي من الجافا وهو الجزء الخاص ببرمجة سطح المكتب Standalone applications، وهو منصة حوسبة لتطوير ونشر التعليمات البرمجية المحمولة لبيئات سطح المكتب والخادم وعمل التطبيقات والأبليت العادية وهي الأساس وأول شيء يتم تعلمه، وكانت الجافا تعرف سابقا بكلمة Java 2 platform وهو الإصدار القياسي J2SE. وتستخدم المنصة لغة برنامج Java وهو جزء من عائلة منصات برنامج Java وهو الجزء الخاص ببرمجة سطح المكتب

(Java Enterprise Edition) Java EE

يختلف عن الإصدار القياسي أنه خاص ببرمجة الأجهزة الخادمة والتطبيقات الموزعة وبرمجة الويب وتطبيقات Servlet ، Enterprise Java Beans ، وهي للمشاريع الكبيرة Multi-tier Enterprises ، كما توفر Java EE نظاما أساسيا للمطورين بميزات المؤسسة مثل الحوسبة وخدمات الويب وكانت تعرف مسبقا بالاسم J2EE وتعرف حاليا بالاسم JakartaEE وهي مجموعة من المواصفات الملتفة حول الإصدار Java Standard Edition- Java SE

(Java Micro Edition) Java ME

إصدار خاص بالبرمجة على الهواتف المحمولة وتعتبر الخيار الأول لمبرمجي ألعاب الهواتف المحمولة، وهي بيئة قوية ومرنة تعمل على الأجهزة المحمولة والأجهزة المدمجة وأجهزة فك التشفير وأجهزة الوسائط الرقمية والطابعات والأجهزة الكفية

J2ME

مكتبة ضخمة جدا تجعل المطورين قادرين على بناء تطبيقات تتميز بالسرعة والقوة والسهولة، وتم تطويرها لبناء تطبيقات يوجد فيها واجهة مستخدم بالاعتماد على لغة الجافا، وتوفر تقنيات جاهزة للبرمجة يتم استخدامها بدون أي تعقيدات وتعمل على مختلف أنظمة التشغيل ويمكن استخدامها في أي لغة برمجة تنفذ بواسطة JVM مثل اللغات Java-JRuby

الفصل الأول: مقدمة إلى لغة الجافا

مزايا لغة Java

تعد لغة Java حاليا من أهم لغات البرمجة نظرا لتمييزها بما يلي:

١- ذات بيئة تشغيل خاصة بها JVM- Java Virtual Machine (الآلة التخيلية للجافا)
قامت لغة Java بإعداد بيئة تشغيل للبرامج التي تبني عليها وتسمى JVM والتي تقوم بترجمة البرنامج للغة الآلة وبالتالي لا ترتبط برامج لغة Java بنظام تشغيل معين

٢- تعمل على معظم برامج نظم التشغيل Platform Independent
أهم ما يميز java أنها لا تعتمد على نظام تشغيل معين (platform) لأنها تعمل على JVM (الآلة التخيلية للجافا) لذلك هي مستقلة عن طبيعة platform وهذا هو السبب الرئيسي لانتشارها الواسع حيث يتيح JVM تنفيذ البرنامج على أكثر من نظام تشغيل

مثلا عند التعامل مع لغة أخرى مثل C++ يتم استخدام المترجم Compiler لتحويل كود البرنامج إلى لغة الآلة و الملف الجديد الناتج عن الترجمة يعمل فقط على نظام التشغيل الذي ترجم فيه بمعنى انه إذا تم عمل الترجمة لبرنامج كتب بلغة C++ تحت نظام تشغيل Windows فان الملف الناتج بلغة الآلة لا يعمل إلا تحت Windows وإذا أردنا تشغيله تحت نظام تشغيل Unix يلزم إعادة الترجمة تحت هذا النظام

C++ → Machine Code

أما في الجافا فيتحول كود البرنامج الى ملف لا يمكن قراءته يسمى Class ثم يتحول الـ Class إلى ملف بلغة الآلة وهذا الـ Class يمكن تشغيله تحت أي نظام تشغيل دون عمل ترجمة للبرنامج مرة أخرى

Java File JDK → Class File JVM → Machine File

حيث أن :

JDK (Java Development kit أدوات تطوير الجافا)
المترجم الذي يستخدم لتحويل البرنامج المكتوب بلغة java إلى ملف امتداده Class وهو الملف الذي يمكن تشغيله على أن نظام تشغيل
JVM (Java Virtual Machine)
المترجم الذي يحول ملف Class إلى ملف بلغة الآلة

٣- تلتزم بقواعد البرمجة الموجهة بالأهداف OOP-object Oriented Programming
تمثل OOP طريقة حديثة للبرمجة والتي توفر كثير من الجهد والذي كان يبذل من خلال البرمجة التقليدية

حيث كانت البرمجة التقليدية توفر للمبرمج مكتبة من الدوال وتركيب نمطى للبرنامج ويقوم المبرمج باستخدام تلك الدوال مع تركيب البرنامج لإنشاء التطبيقات مما يضطره لكتابة سطور كثيرة وإعادة كتابتها عدة مرات فكانت الدالة تمثل وحدة بناء البرنامج
في حين تقوم البرمجة بواسطة الأهداف OOP بإنشاء عناصر متكاملة تحتوى على بيانات ودوال وبالتالي أصبحت وحدة بناء البرنامج هي الفصيلة أو العنصر Object مما سهل واختصر الكثير من الوقت والجهد
كما يمثل OOP ارتباط بين الطرق والخصائص لكائن حيث تقوم الوظائف بأداء وظيفة معينة ويتم فيها تقسيم المشاكل الكبيرة إلى عدة مشاكل صغيرة

الفصل الأول: مقدمة إلى لغة الجافا

- وينقسم حل أي مشكلة لثلاث مراحل :
- تحليل المشكلة وتحديد طريقة حلها وماذا تحتاج لحلها
- تصميم عملية التحليل السابقة بصورة تستطيع معها فهم المشكلة وطريقة حلها في خطوات متسلسلة وواضحة
- كتابة كود البرنامج باللغة التي نختارها.

٤- تتضمن مكتبة فصول كبيرة Class Library

نظرا لأن لغة Java تعتمد على OOP فهي تحتوى على مكتبة فصول قوية تتضمن كل الفصول مثل فصول التعامل مع قواعد البيانات Database وفصول التعامل مع الشبكات Networks وفصول التعامل مع الرسوم المجسمة 3D Graphics وفصول التعامل مع تطوير الويب Web Developing وغيرها

٥- تستند على قواعد لغة C ، C++

- ٦- يمكن باستخدام الجافا تنفيذ عمليات مختلفة في نفس الوقت
من الطبيعي أن يتم تنفيذ خطوات أي برنامج بطريقة متسلسلة أي بعد تنفيذ خطوة ينتقل إلى تنفيذ الخطوة التالية لكن في الجافا يمكن تنفيذ خطوات مختلفة في نفس الوقت.

٧- إدارة العمليات تلقائيا في الذاكرة

في اللغات الأخرى مثل C++ يتم التعامل مع المتغيرات والكائن يدويا أي من خلال البرنامج والمبرمج لكن في الجافا يتم التعامل مع الذاكرة تلقائيا دون تدخل من المبرمج وبذلك تكون أكثر أمانا من اللغات الأخرى

٨- لغة آمنة بشكل كبير جدا

عند تحميل أي برنامج من على شبكة الانترنت فإنه يمكن لأحد القرصنة أن يقوم بالتحكم في الجهاز من خلال كود البرنامج الذي تقوم بتحميله لأنه يمكن التحكم في المتغيرات والكائنات الموجودة في الذاكرة، ولكن من المستحيل حدوث ذلك في الجافا لأنه لا يمكن لأي شخص التحكم في الذاكرة لأن الملف الذي يتم تحميله في الذاكرة من النوع Class وهذا النوع لا يمكن التغيير فيه.

٩- تستخدم في عمل تطبيقات الانترنت وتسمى في تلك الحالة Java Applet

تتميز لغة الجافا بمميزات إضافية أخرى تجعلها من أكثر لغات البرمجة إثارة حيث تمكننا من عمل ما يلي: إضافة الحركة والصوت إلى صفحات الويب ، برمجة الألعاب والبرامج المساعدة ، إنشاء برامج ذات واجهة مستخدم رسومية ، توفر لغة الجافا بيئة تفاعلية عبر شبكة الانترنت وبالتالي تستخدم لكتابة برامج تعليمية على مواقع الويب وبرمجيات الفصول الافتراضية للتعليم الإلكتروني والتعليم عن بعد .

الفصل الأول: مقدمة إلى لغة الجافا

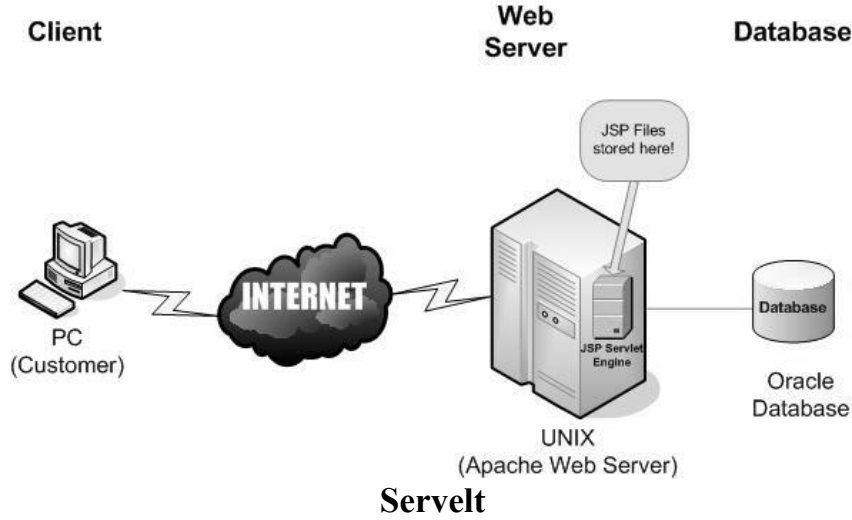
تقنيات تتعلق بلغة Java

Java Virtual Machine JVM (الآلة التخيلية للجافا)

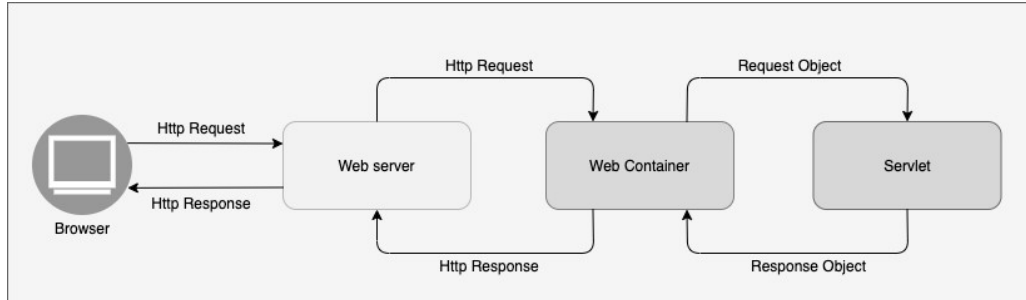
الهدف من إنشاء لغة Java للـ JVM لجعلها تعمل على جميع برامج نظم التشغيل من خلال إنشاء طبقة وسيطة Software تعمل كأنها برامج تشغيل للبرامج Run Time لكل نظام تشغيل

Java Server Pages - JSP

هي لغة بسيطة تتيح بناء صفحات انترنت ديناميكية ذات عمليات متغيرة (مثل عمليات قواعد البيانات) ويتم كتابة أكوادها داخل صفحات HTML مثل جميع الـ Scripts وهي إحدى صور الـ Script (يطلق عليها Server Side Script)

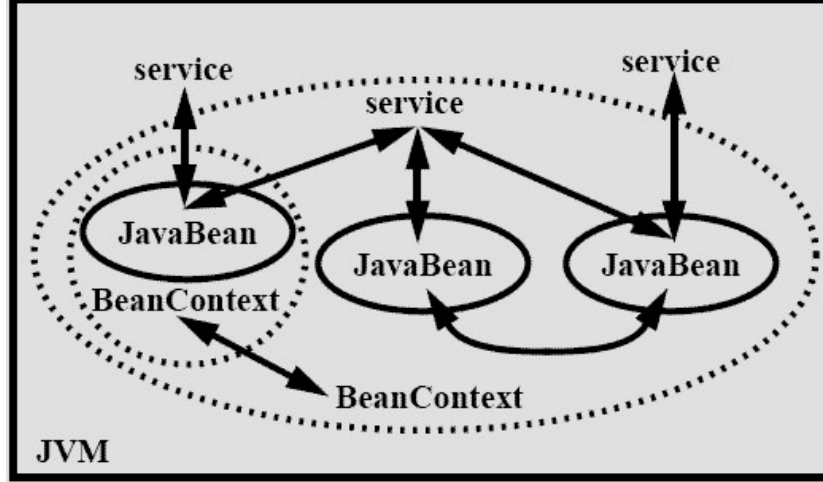


صورة من فصائل الجافا صممت لإعداد تطبيقات على شبكة الانترنت وهي عبارة عن كائن يتم تنفيذها داخل الخادم حيث تقوم باستلام طلبات Requests من العميل وإرسال ردود Response ، وينتمي هذا الكائن إلى الـ GenericServlet Class ، ويمكن استخدام Servlet لتطوير وتحسين أي نوع من الخوادم التي تعتمد أسلوب الطلب والرد وإضافة وظائف جديدة لها سواء كانت هذه الخوادم للبريد أو FTP أو غيرها و ليس خادم ويب فقط ، وتعمل بالشكل التالي:



javaBeans

هي Java Classes لتقليل كتابة الكود بحيث يمكن استخدامها في أكثر من صفحة ويب



وتوصف لغة الجافا بأنها :

- بسيطة (Simple)
- شينية (Object-Oriented)
- موزعة (Distributed)
- مفسرة (Interpreted)
- قوية (Robust)
- محمية (Secure)
- ذات بنين متعادل (Architecture Neutral)
- قابلة للتغيير (Portable)
- متعددة الوظائف (Multithreaded)
- ديناميكية (Dynamic)

ونفسر تلك الخصائص كما يلي:

أولاً: لغة بسيطة Simple

حيث قرر مبتكرو اللغة جعلها سهلة بحيث يتعلمها المبرمج بسرعة فمعظم أوامر java متشابهة مع لغة C++، C كما قام مصممو اللغة بحذف بعض الخصائص الموجودة في C++، C لجعل لغة Java صغيرة وبسيطة

ثانياً : لغة شينية (Object-Oriented)

صممت اللغة أساساً لتكون شينية من خلال تركيزها على البيانات والطرق Methods (في لغة java يطلق على functions ، طرق methods) حيث انه في اللغات الشينية يطلق على مجموعة من البيانات المصطلح class و تعمل الطرق على تشغيل ومعالجة تلك البيانات

ثالثا : لغة موزعة Distributed

أي تقدم دعم متميز للشبكات فعلى سبيل المثال URL عبارة عن Class داخل حزمة java.net فقراءة ملف في جهاز آخر في الشبكة كقراءة ملف في جهازك المحلي حيث تكمن قوة java في (Remote Method Invocation) وهو عبارة عن API لتشغيل الطرق من كائن في الشبكة كما لو أنها في جهازك

رابعا : لغة مفسرة Interpreted

عندما نعمل compile لبرنامج جافا فإنه يعطينا byte-code يقرنه Java Virtual Machine في حين أن لغات البرمجة الأخرى تعطي Machine code وببساطة يتطلب تشغيل برامج جافا أن يكون في جهازك مشغل java JVM

خامسا : لغة قوية Robust

عندما نطلق على برنامج أنه Robust فهذا معناه أنه لا يفشل عند محاولة تشغيله لكن عندما نطلق الشيء نفسه على لغة فمعناه أن اللغة تحد من وجود الأخطاء في البرامج باتخاذ الإجراءات اللازمة لجعل البرمجة بهذه اللغة لا يمكن أن تنتج برامج تحتوي أخطاء (قد توجد أخطاء في برامج java ولكنها محدودة جداً)

سادسا : لغة محمية Secure

من المعروف أن Applets مبرمجة بـ java وتحتاج إلى JVM ، وعند التعامل مع مواقع الويب فإننا نمر بمواقع عدة بعضها يحتوي على Applets والتي قد تتضمن برامج تضر بجهاز الحاسب لذا أخذ في الحسبان عند تصميم java عامل الحماية ووضع لها عدة طبقات لحماية الجهاز من البرامج والأكواد الضارة

سابعا : لغة ذات بنية متعادلة Neutral Architecture وقابلة للتغيير Portable

تتميز java بأنها تعمل على جميع الأجهزة بغض النظر عن نوع نظام تشغيل كل هذا بفضل ال Byte-Code ، JVM ، فالجافا لا تحول البرنامج إلى لغة الآلة إنما تحوله إلى ما يسمى Byte-Code ولكي ينفذ البرنامج فيجب توفر JVM في الجهاز الذي بدوره يقوم بقراءة ال Byte-Code و يحوله إلى ما يتناسب مع نظام التشغيل

ثامنا : متعددة الوظائف Multithreaded

تتميز java بقدرتها على تشغيل عدة مهام في نفس الوقت كما أنها تتيح من خلال خاصية synchronized التحكم بجزئية من البرنامج بحيث يعمل في وقت محدد.

تاسعا : لغة ديناميكية Dynamic Language

حيث يمكن في أي وقت تحميل ال class للمترجم

الفصل الأول: مقدمة إلى لغة الجافا

طرق كتابة برامج الجافا

الطريقة الأولى: الاستخدام المباشر للمكتبة Java Development Kit- JDK

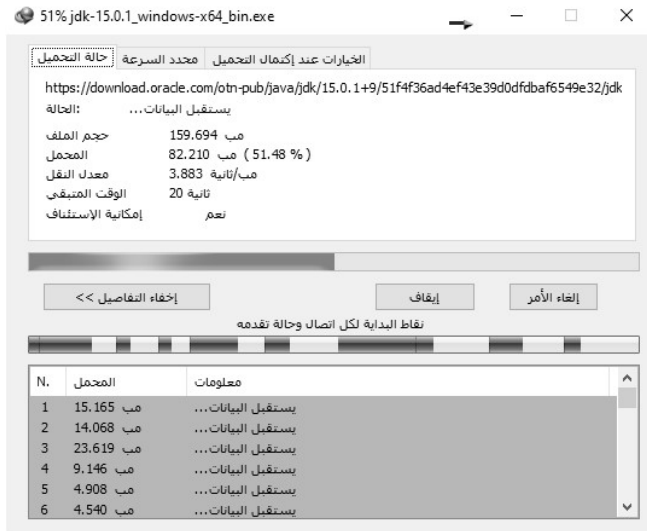
تتطلب تلك الطريقة ما يلي:

- ١- برنامج محرر سطور مثل برنامج Notepad
- ٢- مجموعة JDK (مكتبة ومترجم Java) ويمكن الحصول عليها من خلال تحميلها مباشرة من الموقع التالي:

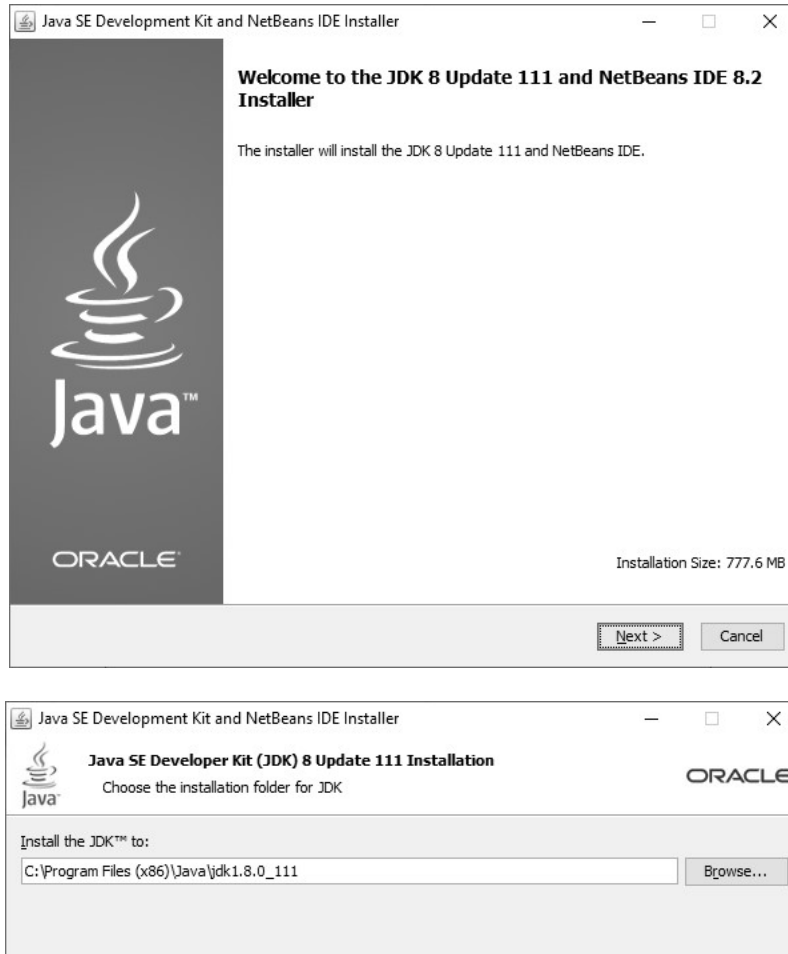
<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

وتتضمن تلك المكتبة الأدوات التالية :

- الملف **Javac** : يقوم بتحويل الملف المصدر إلى ملف تنفيذي **Class**
- الملف **Java** : يقوم بتنفيذ ملف **Java** التنفيذي
- الملف **Applet Viewer** : يستخدم لعرض تطبيقات **Java** من النوع **Applet** أثناء التطوير



الفصل الأول: مقدمة إلى لغة الجافا



الطريقة الثانية: استخدام البرامج المعدة للغة الجافا

حيث يمكن كتابة برامج java بالاستعانة بالبرامج التي يطلق عليها Visual حيث تتيح تلك النوعية من البرامج جميع متطلبات إعداد تطبيق الـ Java. (بيئة رسومية سهلة - مكتبة - مترجم) وتعد تلك الطريقة من أفضل وأقوى الطرق لكتابة برامج Java ، ومن أمثلة تلك البرامج ما يلي:

- برنامج Visual Café

- برنامج JavaBuilder

- برنامج JCreator

- برنامج Oracle JDeveloper (من إنتاج شركة Oracle)

- برنامج NetBeans IDE وهو من البرامج المتميزة لكتابة تطبيقات Java

أشهر البرامج (أداة التطوير التي يمكن استخدامها للبرمجة بلغة جافا

برنامج NetBeans

يعتبر بيئة تطويرية متكاملة مفتوحة المصدر و يعتبر الأشهر تم بناؤه بلغة جافا ، يستخدم لكتابة الاكواد البرمجية و عمل المشاريع في لغة الجافا ويدعم غيرها من اللغات مثل : C - C++ - HTML - PHP -
مميزاته : مجاني- يتوافر منه نسخ لجميع أنظمة التشغيل (Windows-Linux-MAC) -مفتوح المصدر-سهولة الاستخدام - التصحيح التلقائي للأكواد البرمجية-يتيح مجموعة كبيرة من الإضافات Plugins المجانية التي تساعد المبرمج في التطوير والتي يمكن تثبيتها بسهولة.

وبالتالي يعد هذا البرنامج من أفضل أدوات التطوير للغة جافا، وقد تم إعداده باستخدام لغة جافا نفسها بواسطة شركة أوراكل صاحبة تلك اللغة ، ويمكن استخدام هذه الأداة لتطوير برامج بلغات برمجة أخرى غير الجافا مثل برامج PHP ، C++

برنامج Eclipse

يعتبر بيئة تطوير متكاملة ومجانية وتم بناؤه بلغة جافا، يحتوى محرر للنصوص البرمجية يستخدم لكتابة الاكواد وعمل المشاريع البرمجية في لغة جافا وغيرها من اللغات مثل Ruby-Java Script - Python و غيرها العديد من اللغات مثل C++ ، C ،
مميزاته : نفس مزايا NetBeans السابق ذكرها ولكنها أخف وأسرع من أداة التطوير NetBeans ويتضمن إمكانية لبرمجة Android

برنامج IntelliJ

بيئة تطوير متكاملة يستخدم بشكل اساسي للـ Android ويعتبر خياراً جيداً لاستخدامه في كتابه الاكواد البرمجية و عمل المشاريع بلغة الجافا.
مميزاته: نفس المزايا السابق ذكرها

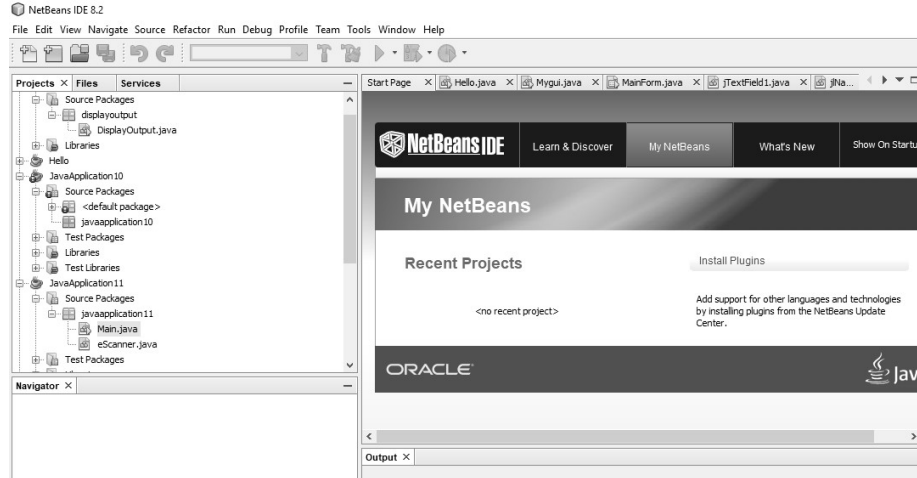
برنامج BlueJ

بيئة تطويرية متكاملة صممت لهدف تعليمي حيث يعتمد على استخدام الألوان بشكل خاص بهدف سهولة كتابة الأكواد خاصة بالنسبة للمبتدئين.
مميزاته: نفس المزايا السابق ذكرها
وتتكون لغة Java من : مجموعة من الحزم Packages ، وتتكون الحزم من مجموعة من الكائنات Objects ، ويتكون الكائن من مجموعة من الدوال Methods وهذه الدوال هي أوامر Java والتي تستخدم في كتابة البرنامج

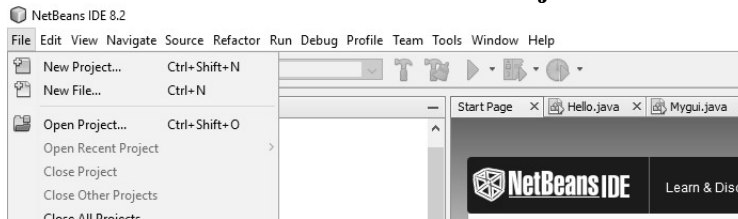
الفصل الأول: مقدمة إلى لغة الجافا

استخدام برنامج NetBeans IDE

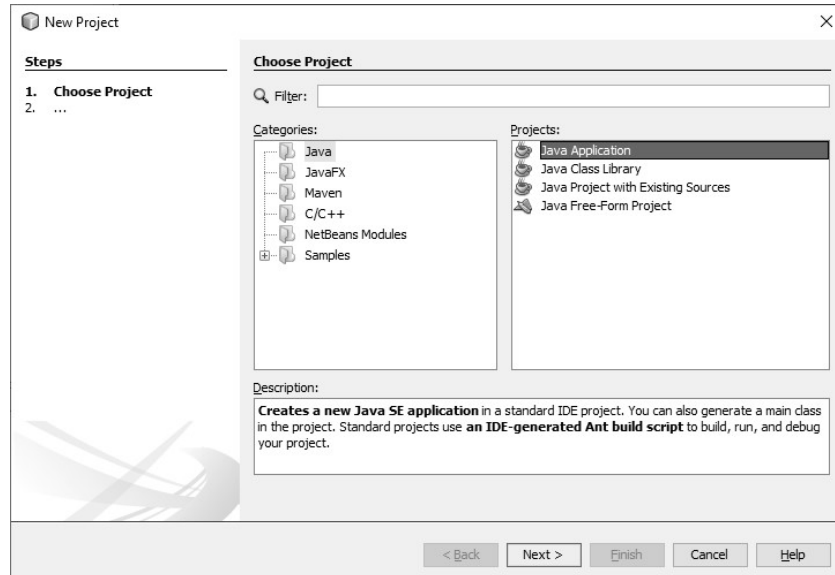
١- تشغيل البرنامج عندئذ نحصل على النافذة التالية :



نفتح قائمة File ونختار منها New Project



يظهر مربع حوارى يتضمن أنواع التطبيقات التي يمكن إنشاؤها، نختار Java (أسفل العنوان Categories) ونختار Java Application (أسفل العنوان Project) ثم نضغط على الزر Next كما يلي:



الفصل الأول: مقدمة إلى لغة الجافا

يظهر مربع حوار يطلب تحديد اسم التطبيق ومسار حفظ ملفاته (نترك باقي الخيارات الافتراضية كما هي)

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location: Browse...

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: Browse...

Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class

< Back Next > Finish Cancel Help

نضغط الزر **Finish** عندئذ يتم إنشاء التطبيق وتظهر نافذة كتابة الكود كما يلي :

```
* To change this template, choose Tools | Templates
* and open the template in the editor.
*/

package javaapplication1;

/**
 *
 * @author asd
 */
public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

الفصل الأول: مقدمة إلى لغة الجافا

تتضمن تلك النافذة الشكل الرئيسي لبرنامج Java. ويتضمن جميع أجزاء البرنامج من الـ **Public Class** وبداخلها الدالة الرئيسية **Main()**

من داخل الدالة الرئيسية **Main()** نقوم باستبدال سطر الكود:

// TODO code application logic here

بالسطر التالي:

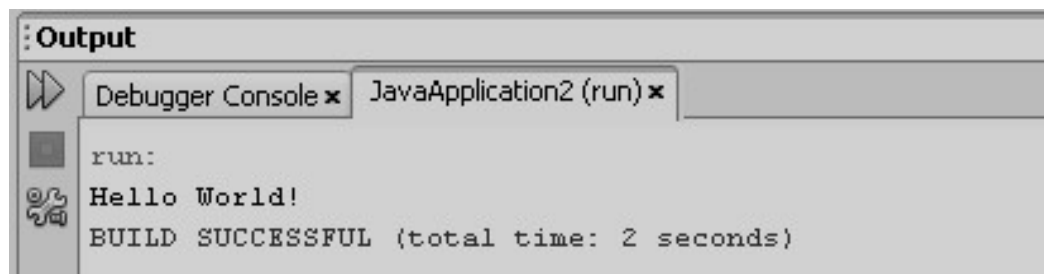
System.out.println("Hello World!"); // Display the string

```
package javaapplication2;

/**
 *
 * @author asd
 */
public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("Hello World!"); // Display the string
    }
}
```

تنفيذ البرنامج من خلال فتح قائمة **Run** ونختار منها **Run Main Project** (أو نضغط مباشرة مفتاح **F6**) عندئذ يظهر ناتج التنفيذ بالشكل التالي:



شرح شاشة الكود للبرنامج

- ١- يتكون برنامج Java من :
Method- : عبارة عن مجموعة من العمليات مكتوبة بلغة Java تنفذ بمجرد استدعاؤها ويقابلها Function في لغة C
Class- : عبارة عن البرنامج المكتوب بلغة Java وتحتوي على مكونات البرنامج
- ٢- الدالة الرئيسية Main هي الدالة التنفيذية في البرنامج ولا يمكن تشغيل أي برنامج بدون وجود تلك الدالة الرئيسية وهي تنفذ بدون استدعاء ولا تنتظر استدعاء
- ٣- يتكون أي برنامج من دالة واحدة على الأقل هي الدالة Main
- ٤- يمكن أن يكون هناك دوال أخرى تعرف قبل أو بعد الدالة Main ويتطلب تنفيذ تلك الدوال استدعاؤها داخل الدالة Main (لا يهم ترتيب الـ method في java حيث يمكن أن يكون الـ method الرئيسي في الآخر)
- ٥- يجب أن يكون الـ method التي هي خارج object من النوع static
- ٦- تتكون أي دالة (بما في ذلك الدالة الرئيسية Main) من :
- الإعلان عن الدالة وهو بمثابة عنوان الدالة ويتضمن (اسم الدالة - قائمة المتغيرات التي ستدخل إلى الدالة أو ستخرج منها)
- جسم الدالة ويوضع بين قوسين { } ويتضمن العبارات التي تشمل الأوامر
- يكون شكل الدالة الرئيسية كما يلي :

```
public static void main(String []arg)
{
    العبارات
}
```
- ٧- يتوافر في الدالة الرئيسية main الصفات التالية: الكلمة Void (تدل على نوع الدالة أي لا تعيد قيم) ، والكلمة main (اسم خاص بها ولا تأخذ اسم آخر) ، والكلمات String [] arg (تعني أن للدالة معامل من نوع مصفوفة حرفية String وتمثل معاملات البرنامج)
- ٨- يجب أن يكتب أي برنامج Java داخل فئة أو فصيلة حيث يتم كتابة الكلمة class (الحرف الأول c يكتب صغير) يليها الاسم الذي سيعطى لهذا الـ class ويجب أن يكون الحروف الأول لهذا الاسم كبير Capital أو كل قسم منه يبدأ بحرف كبير لأن الاسم يكون متلاصق مثل FirstExample (القسم الأول First والقسم الثاني Example) ويتم بعد ذلك كتابة باقى البرنامج داخل أقواس الفئة { }

الفصل الأول: مقدمة إلى لغة الجافا

٩- تتكون الجافا من مجموعة من الحزم Package ويوجد حزمة تستدعى ذاتيا دون أن نقوم باستدعائها في البرنامج وتسمى تلك الحزمة Java.lang ويوجد بداخلها مجموعة من الدوال أو الطرق Method مثل (System.out.println) حيث أن الـ System هو اسم الفصيلة Class (يتم كتابة الحرف الأول من كلمة System بحجم كبير) ، وبداخل اسم الفصيلة دالة Method تسمى out وبداخلها دالة أخرى تسمى Println ، أي أنه يتكون الـ Class من عدد من Method ويتكون الـ Method من Method أخرى وتتكون الحزمة Package من عدة Class ويمكن تلخيص كل ذلك كما يلي :

- تتكون الحزمة Java.lang من عدة Class ومن هذه الـ Class الفصيلة System

- يوجد داخل الفصيلة System دوال ومنها الدالة Out

- يوجد داخل الدالة Out الدالة Println الخاصة بالطباعة

- تنتهي جميع الجمل والعبارات بفاصلة منقوطة

الكلمة Public

تحدد هذه الكلمة صلاحية الوصول إلى الدالة، فمثلاً إذا اعتبرت أن الدالة موجودة بصنف class وهذا الصنف موجود بداخل مجلد وهناك مجلدات أخرى بها أصناف أخرى، في البرمجة الكائنية عادةً ما تحاول الأصناف مناداة دوال موجودة بأصناف أخرى، لذا كنوع من الأمان يتم تحديد صلاحية الوصول إلى الدالة (لأعضاء الصنف فقط، أم الأصناف بداخل المجلد نفسه، أم... إلخ) ، وهذه الكلمة اختيارية: بمعنى أنها من الممكن أن تختفي تماماً ويكتفى تعريف الدالة دون وجود أي محدد لصلاحية الوصول.

الكلمة Static

كلمة مفتاحية مشهورة جداً في لغة الجافا، وتؤدي static مهمة تتعلق بالذاكرة فهي تساعد المبرمج على إنشاء نظام فعال في استخدام الذاكرة أي يقلل من استخدام الذاكرة وذلك لأن كل متغير أو دالة أو غيره تم تعريفه بأنه static يعني ذلك أن هذا المتغير سيتم حجز مكانه في الذاكرة لمرة واحدة فقط ، فالمتغيرات والدوال التي يتم تعريفها بـ static هي تتبع الـ class وليس الـ object وهذا يعني أن كل متغير تم تعريفه على أنه static يتيح للمبرمج الوصول إليه دون إنشاء object عن الـ class حيث يحتاج فقط إلى استخدام تلك العبارة

classname.StaticElementName؛

البرنامج التالي يتم استدعاء المتغير name (يمثل إسم الطالب) ، وتغيير قيمته دون إنشاء object

```
1 public class CallStatic {
2     public static void main(String args[]) {
3         Student.name = "Elsaeed";
4         System.out.print(Student.name);
5     } //end main
6 }
7 class Student {
8     static String name;
9 }
```

ناتج التنفيذ:

Elsaeed

الاستخدامات المفيدة لكلمة static

١- استخدام static لتعريف المتغيرات الغير قابلة للتغيير منطقياً
كثيراً ما نجد متغيرات محددة القيمة داخل الفئة ومعرفة بـ final ولكنها غير معرفة بـ static وفي نفس الوقت قيمتها لن تتغير إطلاقاً خلال البرنامج لذا يتم جعلها مشتركة وتوفير المساحة بالذاكرة

٢- استخدام static مع الدوال التي من المنطقي استخدامها قبل إنشاء object
عند الرغبة في إنشاء class لسيارة مثلاً، فمن الدوال التي يجب وضعها static الدالة التي تغير وحدات قياس السرعة بين كيلومتر وميل ، فهذه الدالة قد تستخدمها بدون إنشاء class لإجراء أي تغيير ، أما الدوال مثل سرعة السيارة الحالية فيجب أن لا تعرف كـ static لأنها متعلقة بكل سيارة لوحدها.

الفصل الأول: مقدمة إلى لغة الجافا

٣- إجراء تغييرات على متغير static ، أنشئ دالة static
ليس دائماً، ولكن الشرط هو أن تكون لديك عمليات معقدة تجريها على متغير static، ففي هذه الحالة أنشئ دالة من النوع static حتى تستطيع أن تستدعيها دون أن تنشئ object ، فالمتغير الـ static بالإمكان تغييره وإسناد قيمة له وتعديلها دون إنشاء أي object

٤- استخدم static مع المتغيرات المشتركة بين الـ objects ، هذه المتغيرات إذا ما تغير في أحد الـ classes سوف تتغير في الباقي، عند تعريفك لـ class يخص طالب في جامعة، فمن المتغيرات التي يجب أن تعرف بـ static اسم الجامعة و تاريخ إنشائها حيث إن هذه المتغيرات غير قابلة للتغيير على مستوى الطلاب، فإذا أنشأت عدة objects فلن يتغير اسم الجامعة و هنا تأتي فائدة static بأنه مهما كان عدد الـ objects تبقى المساحة المحجوزة للمتغير الخاص باسم الجامعة واحدة، ولكن المتغيرات مثل اسم الطالب و عمره يتم إنشاؤها مع كل object لوحده.

٥- استخدم static مع الدوال غير المتعلقة بالـ object
توجد بعض الدوال التي لا تتعلق بالـ object الذي سيتم إنشاؤه، و لكن يتم إنشاؤها كدوال مساعدة، ليس بالضرورة أن تكون هذه الدوال من النوع الذي يُستدعى قبل إنشاء الـ object بل أي دالة مساعدة غير متعلقة بالـ object اجعلها static لتوفر بعض المساحة بالذاكرة.

الكلمة Void

void تعني فراغ أي أنه لن تعيد الدالة أي شيء، ولكن إذا أردتها أن تعيد شيئاً يجب أن تحدد نوع البيانات الذي ستعيده، مثلاً إذا كانت ستعيد عمر إنسان عندئذ نكتب int بدلاً من void

الكلمة main

اسم الدالة، والذي تستطيع اختياره وفقاً لمهمة الدالة، مثلاً إذا كانت الدالة التي ترغب بإنشائها هي المشي فسيكون اسم الدالة walk وهكذا

الكلمة String[] args

تسمى بالمعاملات parameters وهنا تستقبل الدالة مصفوفة من النوع String تسمى بـ args ، من الممكن أن تستقبل الدالة أنواعاً مختلفة من المتغيرات ويمكن تسميتها كما نريد: مثلاً (int firstNumber, int secondNumber)

ما يوجد بين القوسين المعقوفين {}

كل ما يوجد هنا هو منطق الدالة، أي الخطوات التي يجب تنفيذها لإكمال هدف الدالة، مثلاً في حالة الجمع بين القوسين المعقوفين يتم إجراء عملية الجمع وتخزين الناتج في متغير آخر وإعادة قيمة الناتج.

مبادئ كتابة كود Java

١- Class name : دائماً يبدأ class name بحرف كبير

مثال :

```
class First
```

إذا كان class name يتكون من أكثر من كلمة يتم جعل أول حرف من كل كلمة كبير

مثال :

```
class FirstJavaCla
```

٢- إسم الدالة : دائماً يبدأ اسم الدالة بحرف صغير

مثال :

```
public void display()
```

إذا كان اسم الدالة يتكون من أكثر من كلمة يتم جعل أول كلمة تبدأ بحرف صغير وجميع الكلمات التي تليها تبدأ بحرف كبير.

مثال :

```
public void displayUserInfo()
```

٣- إسم الملف الذي يحتوي على class

يجب أن يكون java class name مطابقاً لاسم الملف الموجود بداخله.

مثال:

إذا كان class name هو class elsaeed فيجب أن يكون موجوداً في ملف جافا بالاسم
elsaeed.java

ملحوظة

إذا كان class name مختلف عن اسم الملف فإن مترجم لغة جافا javac (والذي يقوم بتحويل كود الجافا للغة التي يفهمها الحاسب) لن يستطيع إيجاد class لأنه يبحث عن الملف أولاً ثم يتوقع أن يكون class يحمل نفس الاسم مما يترتب عليه مشاكل في الكود.

بلوك الأوامر Block Statements

يوجد في لغة Java بلوك للأوامر ويعني مجموعة من الجمل أو الأوامر محصورة بين القوسين { } وذلك بهدف الإعلان عن متغيرات ليس لها علاقة بما قبل أو بعد البلوك مثل تعريف متغير داخل بلوك داخل دالة فهو يكون خاص بها كما يلي:

```
{  
    Statement1  
    Statement2  
    Statement3  
}
```

البرنامج الأول

بعد تثبيت آلة جافا الافتراضية وأداة التطوير NetBeans نقوم باختيار **New/Project** ثم **Java/Java Application** ثم نقوم بتسمية البرنامج **hello** ليظهر لنا الكود التالي :

1	/*
2	<i>* To change this template, choose Tools Templates</i>
3	<i>* and open the template in the editor.</i>
4	*/
5	package hello;
6	/*
7	*
8	<i>* @author motaz</i>
9	*/
10	public class Hello {
11	/*
12	<i>* @param args the command line arguments</i>
13	*/
14	public static void main(String[] args) {
15	<i>// TODO code application logic here</i>
16	}
17	
18	}

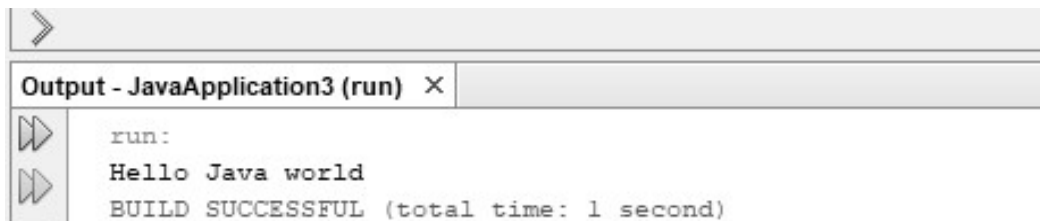
كتابة السطر التالي داخل الإجراء **main**

System.out.print("Hello Java world\n");

ليصبح الكود كالتالي:

```
1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package hello;
6  /*
7  *
8  * @author motaz
9  */
10 public class Hello {
11 /*
12 * @param args the command line arguments
13 */
14 public static void main(String[] args) {
15 // TODO code application logic here
16 System.out.print("Hello Java world\n");
17     }
18 }
19
```

يتم تشغيل البرنامج عن طريق المفتاح F6 لتظهر المخرجات أسفل شاشة NetBeans كما يلي:



الفصل الأول: مقدمة إلى لغة الجافا

جمل الطباعة print output في لغة Java

يوجد في لغة Java ثلاث دوال يمكن استخدامها للطباعة كما يلي:

الدالة	الغرض منها
System.out.print()	دالة تستخدم لعرض أي شيء يتم وضعه داخلها (نص - رقم - قيمة متغير) ويجب وضع النصوص بين علامتي تنصيص مزدوجة
System.out.println()	دالة تستخدم أيضا لعرض أي شيء يتم وضعه داخلها (نص - رقم - قيمة متغير) ، الفرق بينها وبين الدالة السابقة أنها تنتقل إلى سطر جديد لأسفل بعد عملية الطباعة الحالية
System.out.printf()	دالة تستخدم لعرض أي شيء يتم وضعه داخلها (نص - رقم - قيمة متغير) ، وتختلف فكرة واستخدام تلك الدالة عن الدالتين السابقتين حيث تم تصميمها لتحقيق مبدأ Concatenation بطريقة مختلفة فقط ، فكل شيء يراد عرضه يجب حجز مكان له ، فمثلا عند عرض عدد صحيح أي نوعه int يجب استخدام الرمز %d لحجز مكان له في دالة العرض ، وعند عرض عدد بفاصلة عشرية أي نوعه float يجب استخدام الرمز %f لحجز مكان له في دالة العرض ، وعند عرض نص عادي String يتم استخدام الرمز %s لحجز مكان له في دالة العرض.

Concatenation تعني سلسلة باللغة العربية ، وبرمجياً تعني دمج عدة أشياء مع بعضها سواء كانت نصوص أو أرقام وجعلها تبدو شيئاً واحداً.

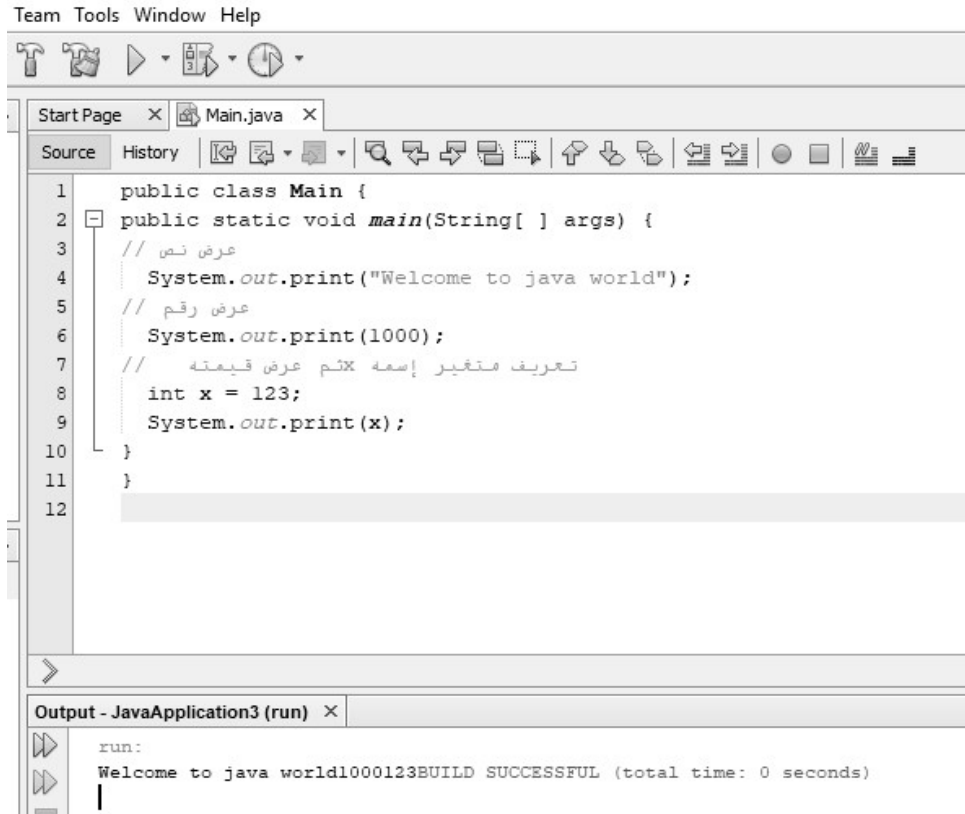
مثال:

عرض ثلاث أشياء باستخدام الدالة print()

```
1 public class Main {
2     public static void main(String[] args) {
3         // عرض نص
4         System.out.print("Welcome to java
5 world");
6         // عرض رقم
7         System.out.print(1000);
8         // تعريف متغير x ثم عرض قيمته
9         int x = 123;
10        System.out.print(x);
11    }
12 }
```

الفصل الأول: مقدمة إلى لغة الجافا

نضغط مفتاح F6 عندئذ تظهر نتيجة التنفيذ كما يلي:



The screenshot shows an IDE with a menu bar (Team, Tools, Window, Help) and a toolbar. The 'Source' tab is active, displaying the following Java code in 'Main.java':

```
1 public class Main {
2     public static void main(String[] args) {
3         // عرض نص
4         System.out.print("Welcome to java world");
5         // عرض رقم
6         System.out.print(1000);
7         // تعريف متغير اسمه x ثم عرض قيمته
8         int x = 123;
9         System.out.print(x);
10    }
11 }
12
```

The 'Output - JavaApplication3 (run)' tab is also visible, showing the execution result:

```
run:
Welcome to java world1000123BUILD SUCCESSFUL (total time: 0 seconds)
```

مثال : نفس المثال السابق ولكن باستخدام الدالة `println()`

```
1 public class Main {
2     public static void main(String[] args) {
3         // عرض نص
4         System.out.println("Welcome to java world");
5         // عرض رقم
6         System.out.println(1000);
7         // تعريف متغير x ثم عرض قيمته
8         int x = 123;
9         System.out.println(x);
10    }
11 }
```

الفصل الأول: مقدمة إلى لغة الجافا

```
1 public class Main {
2     public static void main(String[ ] args) {
3         // عرض نص
4         System.out.println("Welcome to java world");
5         // عرض رقم
6         System.out.println(1000);
7         // تعريف متغير اسمه x ثم عرض قيمته
8         int x = 123;
9         System.out.println(x);
10    }
11 }
12
```

Output - JavaApplication3 (run) X

```
run:
Welcome to java world
1000
123
BUILD SUCCESSFUL (total time: 0 seconds)
```

مثال :
استخدام الدالة `printf()` مع دمج النص الموجود في تلك الدالة مع رقم (يجب وضع فاصلة قبل كل متغير أو قيمة سيتم استبدالها)

```
1 public class Main {
2     public static void main(String[ ] args) {
3         // تعريف متغير x قيمته 10
4         int x = 10;
5         // وضع قيمة المتغير x مكان الـ %d بعدها يتم عرض كامل محتوى دالة الطباعة
6         System.out.printf("The value of x is: %d", x);
7     }
8 }
```

الفصل الأول: مقدمة إلى لغة الجافا

```
1 public class Main {
2     public static void main(String[] args) {
3         // تعريف متغير اسمه x قيمته 10
4         int x = 10;
5         // وضع قيمة المتغير x مكان الـ %d بعدما يتم عرض كامل محتوى دالة الطباعة
6         System.out.printf("The value of x is: %d", x);
7     }
8 }
9
10
```

Output - JavaApplication3 (run) ×

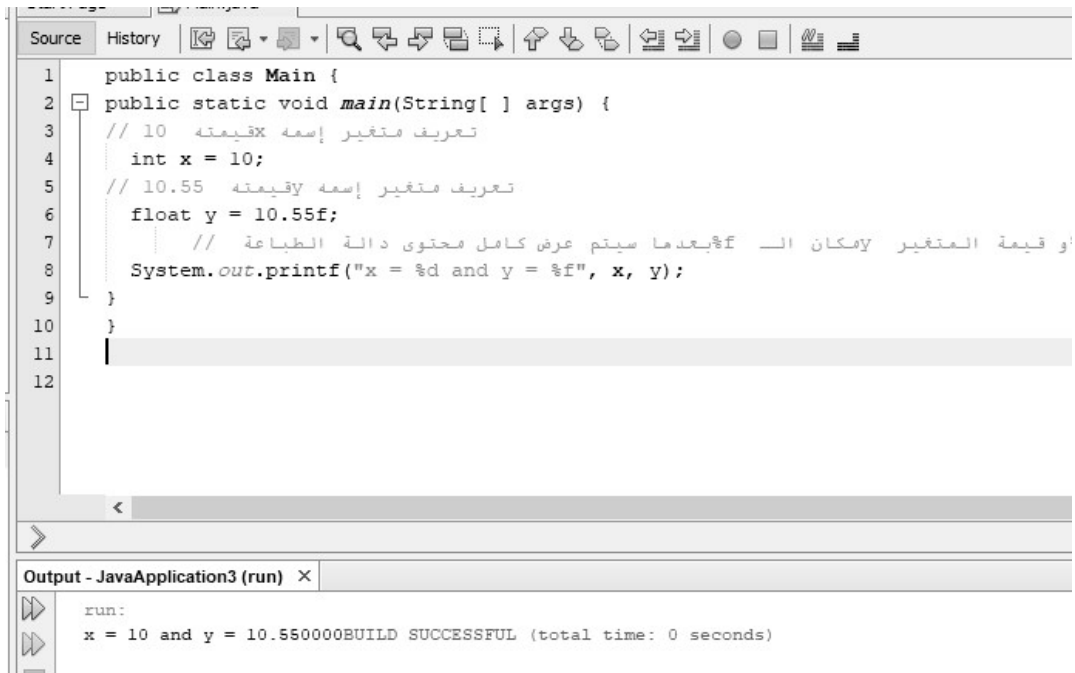
run:
The value of x is: 10BUILD SUCCESSFUL (total time: 0 seconds)

مثال:

دمج النص الموجود في الدالة printf() مع رقمين، الأول نوعه int ، والثاني نوعه float (يجب وضع فاصلة قبل كل متغير أو قيمة سيتم استبدالها)

```
1 public class Main {
2     public static void main(String[] args) {
3         // تعريف متغير x قيمته 10
4         int x = 10;
5         // تعريف متغير y قيمته 10.55f
6         float y = 10.55f;
7         // و قيمة المتغير y مكان الـ %f بعدها سيتم عرض كامل محتوى دالة الطباعة
8         // وضع قيمة المتغير x مكان الـ %d
9         System.out.printf("x = %d and y = %f", x, y);
10    }
```


الفصل الأول: مقدمة إلى لغة الجافا



The screenshot shows an IDE with a source editor and an output window. The source editor contains the following Java code:

```
1 public class Main {
2     public static void main(String[] args) {
3         // تعريف متغير اسمه x قيمته 10
4         int x = 10;
5         // تعريف متغير اسمه y قيمته 10.55
6         float y = 10.55f;
7         // قيمة المتغير y مكان الـ %f بعدما سيتم عرض كامل محتوى دالة الطباعة
8         System.out.printf("x = %d and y = %f", x, y);
9     }
10 }
11
12
```

The output window, titled "Output - JavaApplication3 (run)", shows the following text:

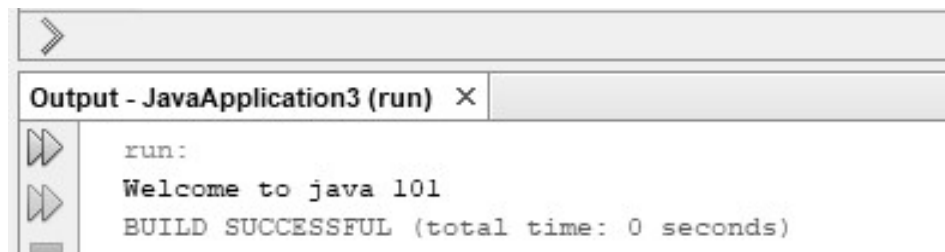
```
run:
x = 10 and y = 10.550000BUILD SUCCESSFUL (total time: 0 seconds)
```

مثال :

برنامج يحتوي على أمر طباعة يتيح دمج ثلاث كلمات ، ورقم ، وعرضهم كجملته واحدة

```
1 public class Main {
2     public static void main(String[] args) {
3         // دمج ثلاث كلمات ورقم وعرضهم كجملته واحدة
4         System.out.println("Welcome " + "to " + "java " + 101);
5     }
6 }
```

نضغط مفتاح F6 عندئذ تظهر نتيجة التنفيذ كما يلي:



The screenshot shows the output window with the following text:

```
run:
Welcome to java 101
BUILD SUCCESSFUL (total time: 0 seconds)
```

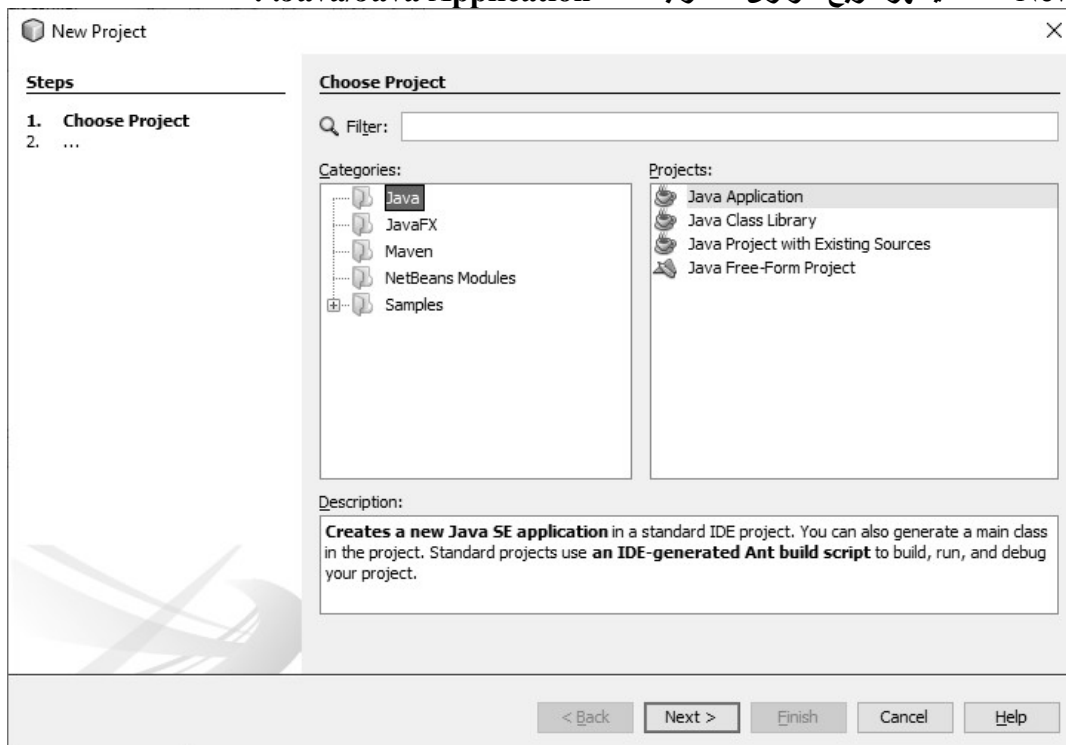
مثال لدالة جمع عددين

البرنامج التالي يحتوي على الدالة الرئيسية main والتي تنادى دالة الجمع sum ثم طباعه الناتج

```
1 static int sum(int firstNumber, int secondNumber){
2     int sum = firstNumber + secondNumber;
3     return sum;
4 }
5 public static void main(String args[]){
6     System.out.println("sum = " + sum(5,4));
7 }
```

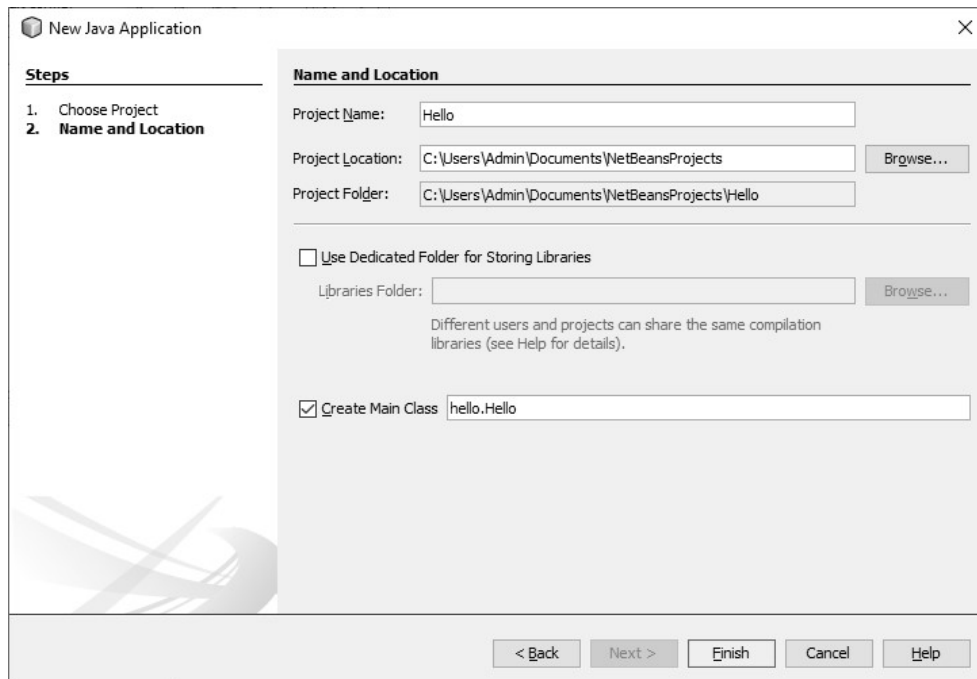
برنامج

بعد تثبيت آلة جافا الافتراضية وأداة التطوير NetBeans نقوم بفتح قائمة File ونختار Project New عندئذ يظهر مربع حوارى نختار بداخله Java/Java Application :



نضغط Next ثم تسمية البرنامج Hello

الفصل الأول: مقدمة إلى لغة الجافا



The image shows the 'New Java Application' dialog box in NetBeans. It has a 'Steps' pane on the left with two steps: '1. Choose Project' and '2. Name and Location'. The 'Name and Location' pane is active. It contains the following fields and options:

- Project Name:** Hello
- Project Location:** C:\Users\Admin\Documents\NetBeansProjects (with a 'Browse...' button)
- Project Folder:** C:\Users\Admin\Documents\NetBeansProjects\Hello
- ☐ Use Dedicated Folder for Storing Libraries
- Libraries Folder:** (with a 'Browse...' button)
- Different users and projects can share the same compilation libraries (see Help for details).
- ☒ Create Main Class hello.Hello

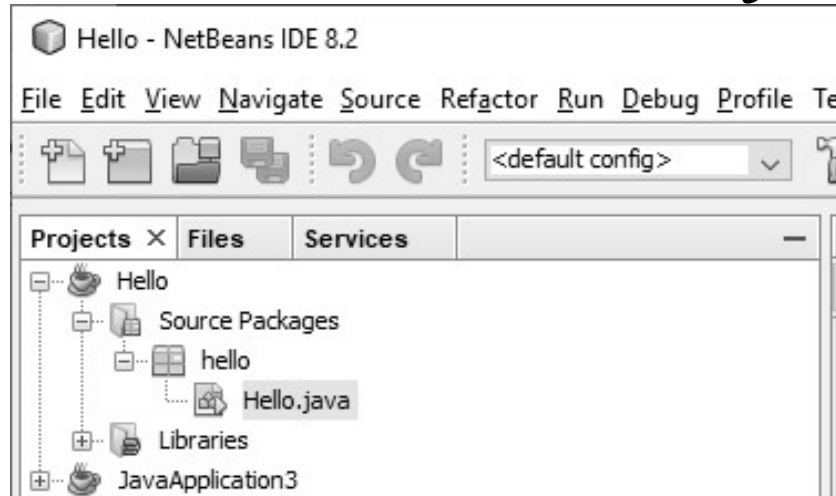
At the bottom, there are buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

نضغط الزر **Finish** عندئذ يظهر الكود التالي:

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package hello;
7
8  /**
9   *
10   * @author Admin
11   */
12  public class Hello {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19      }
20
21  }
```

الفصل الأول: مقدمة إلى لغة الجافا

فإذا لم يظهر الكود نقوم بفتح الملف `Hello.java` بواسطة شاشة المشروع التي تظهر يسار شاشة NetBeans كما يلي:



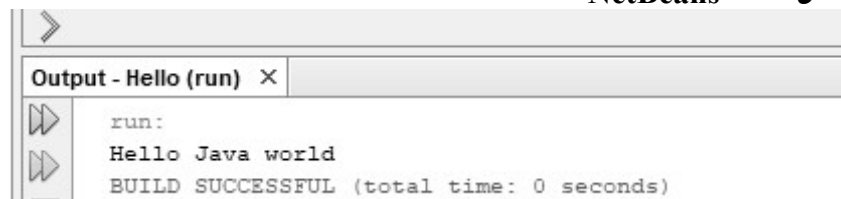
بعد ذلك نقوم بكتابة السطر التالي داخل الإجراء `Main`

```
System.out.print("Hello Java world\n");
```

ليصبح الكود كالتالي:

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package hello;
7
8   /**
9    *
10   * @author Admin
11   */
12  public class Hello {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19          System.out.print("Hello Java world\n");
20      }
```

يتم تشغيل البرنامج من خلال المفتاح `F6` أو الضغط على الزر `Run` (السهم ذو اللون الأخضر) ليظهر المخرجات أسفل شاشة NetBeans



الفصل الأول: مقدمة إلى لغة الجافا

لتشغيل البرنامج خارج أداة التطوير :

- ١- نقوم ببناء الملف التنفيذي بواسطة Build وذلك بالضغط على مفتاحي Shift+F
- ٢- نبحث عن الدليل الذي يحتوي على برنامج NetBeans ويكون اسمه في الغالب NetBeansProjects ثم داخل الدليل hello نجد دليل اسمه dist يحتوي على الملف التنفيذي في هذه الحالة يكون اسمه Hello

Building jar: C:\Users\Admin\Documents\NetBeansProjects\Hello\dist\Hello.jar

- ٣- يمكن تنفيذ هذا البرنامج في سطر الأوامر في نظام التشغيل بواسطة كتابة الأمر التالي:
java -jar hello.jar
- ٤- يمكن نقل هذا الملف التنفيذي من نوع Byte code إلى أي نظام تشغيل آخر يحتوي على آلة جافا الافتراضية ثم تنفيذه بهذه الطريقة
ونلاحظ أن حجم الملف التنفيذي صغير نسبياً (حوالي كيلو ونصف) وذلك لأننا لم نستخدم مكتبات إضافية
- ٥- بعد ذلك نقوم بتغيير الكود إلى التالي:

```
int num = 9;  
System.out.print(num + " * 2 = " + num * 2 + "\n")
```

وهذه طريقة لتعريف متغير صحيح أسميناه num وأسندنا له قيمة ابتدائية 9

وفي السطر الذي يليه قمنا بكتابة قيمة المتغير، ثم كتابة قيمته مضروبة في الرقم 2 وفي نهاية الإجراء أضفنا الرمز \n والذي يمثل رمز السطر الجديد في شاشة نظام التشغيل

طباعة التاريخ والساعة الحاليين نكتب هذه الأسطر

```
1 Date today = new Date();  
2 System.out.print("Today is: " + today.toString() + "\n");
```

ولابد من إضافة المكتبة المحتوية على الفئة Date في بداية البرنامج

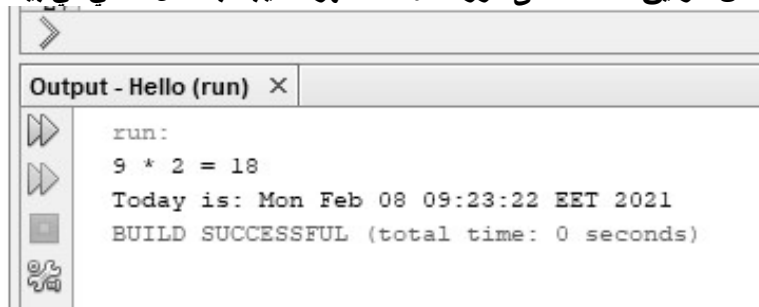
```
1 import java.util.Date;
```

فيصبح شكل كود البرنامج الكلي كما يلي:

```
1  */
2  * To change this license header, choose License Headers in Project
   Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  /*
6  package hello؛
7  import java.util.Date؛
8
9  **/
10 *
11 @ * author Admin
12 /*
13 public class Hello}
14
15 **/
16 @ *   param args the command line arguments
17 /*
18     public static void main(String[ ] args)}
19     //      TODO code application logic here
20     int num = 9؛
21     System.out.print(num + " * 2 = " + num * 2 + "\n")؛
22     Date today = new Date( )
23     System.out.print("Today is: " + today.toString( ) + "\n")؛
24     {
25     {
```

(يُمكن إضافة اسم المكتبة تلقائياً عند ظهور العلامة الصفراء شمال السطر الموجودة فيه الفئة)

تشغيل البرنامج عن طريق الضغط على الزر Run فتظهر النتيجة بالشكل التالي في بيئة NetBeans



الفصل الأول: مقدمة إلى لغة الجافا

برنامج واجهة رسومية

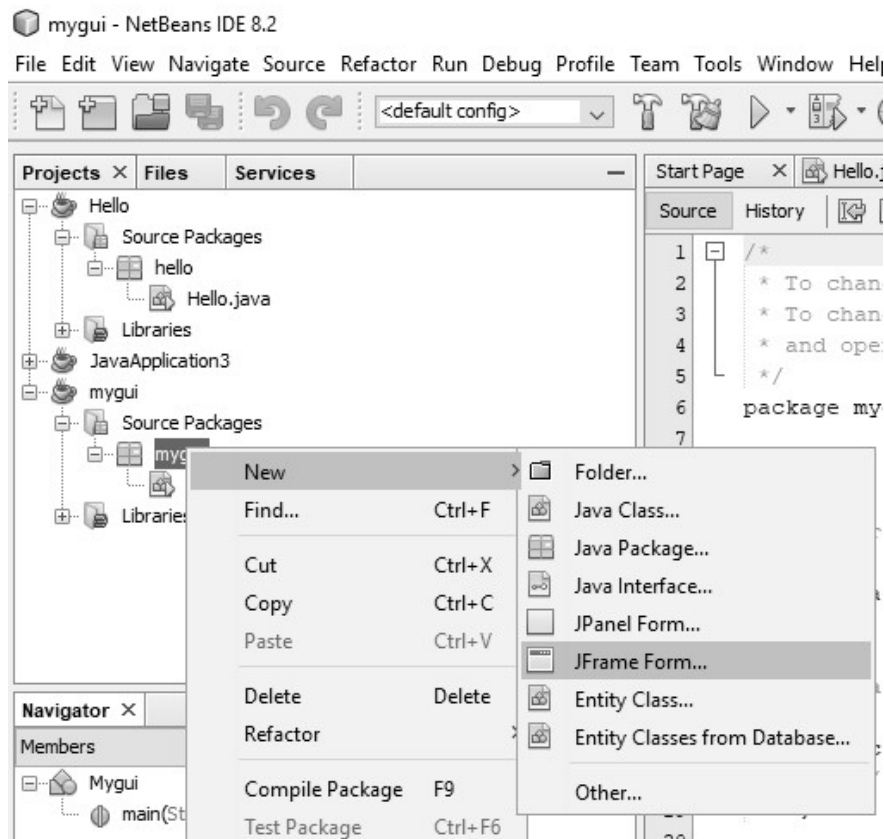
من أهم الأشياء في أدوات التطوير هو دعمها للواجهات الرسومية أو مايسمى بـ Widgets ، فكل نظام تشغيل يحتوي على مكتبة أو أكثر تمثل واجهة رسومية مثلا:
- يوجد في نظام Linux واجهات GTK ، QT
- يوجد في نظام Windows مكتبة ويندوز الرسومية
يوجد في نظام Macintosh مكتبات Carbon ، Cocoa

أما java فلها مكتباتها الخاصة والتي تعمل في كل هذه الأنظمة ومنها واجهة Swing

لكتابة أول برنامج ذو واجهة رسومية في جافا باستخدام NetBeans

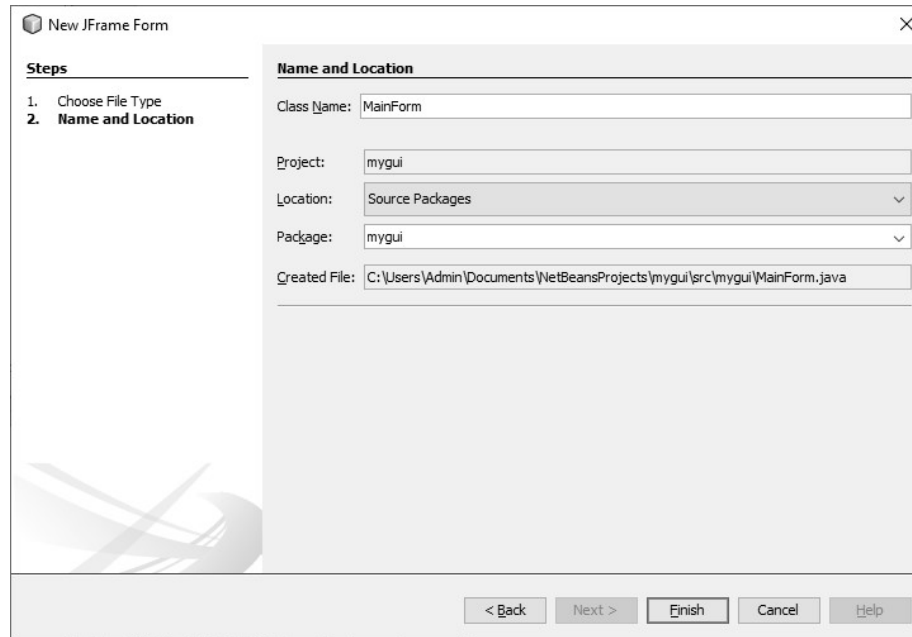
نختار File/New Project ثم نختار Java/Java Application ونقوم بتسميته mygui

في شاشة Projects نختار الحزمة mygui ثم بالزر الأيمن للماوس نختار New/JFrame Form

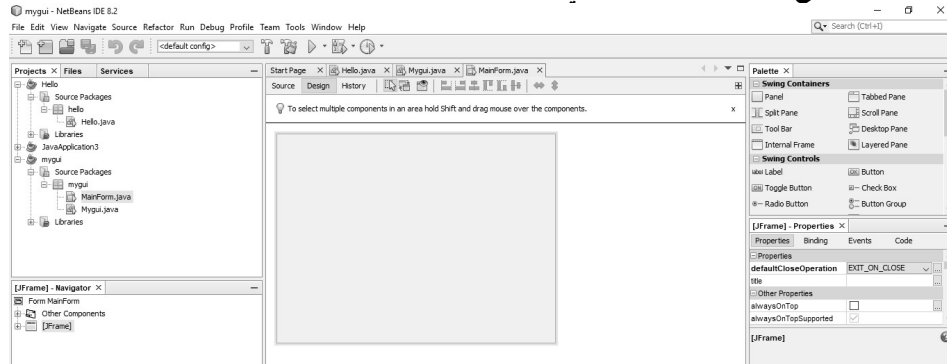


نسعى هذا الفورم MainForm

الفصل الأول: مقدمة إلى لغة الجافا



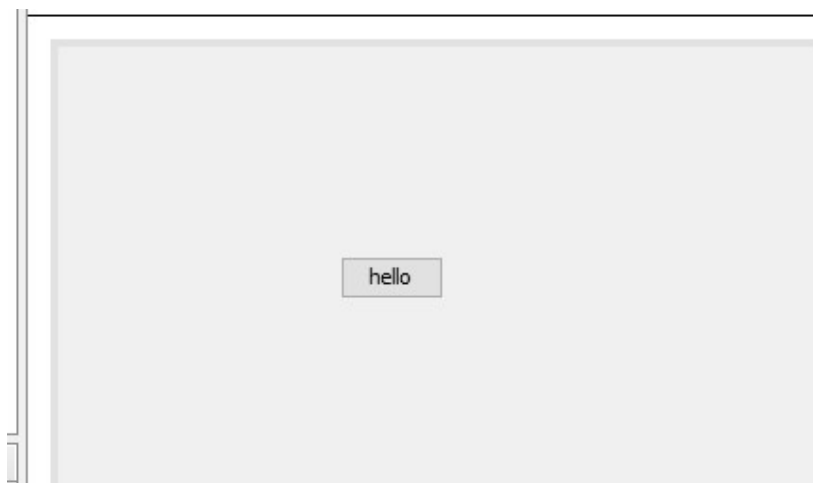
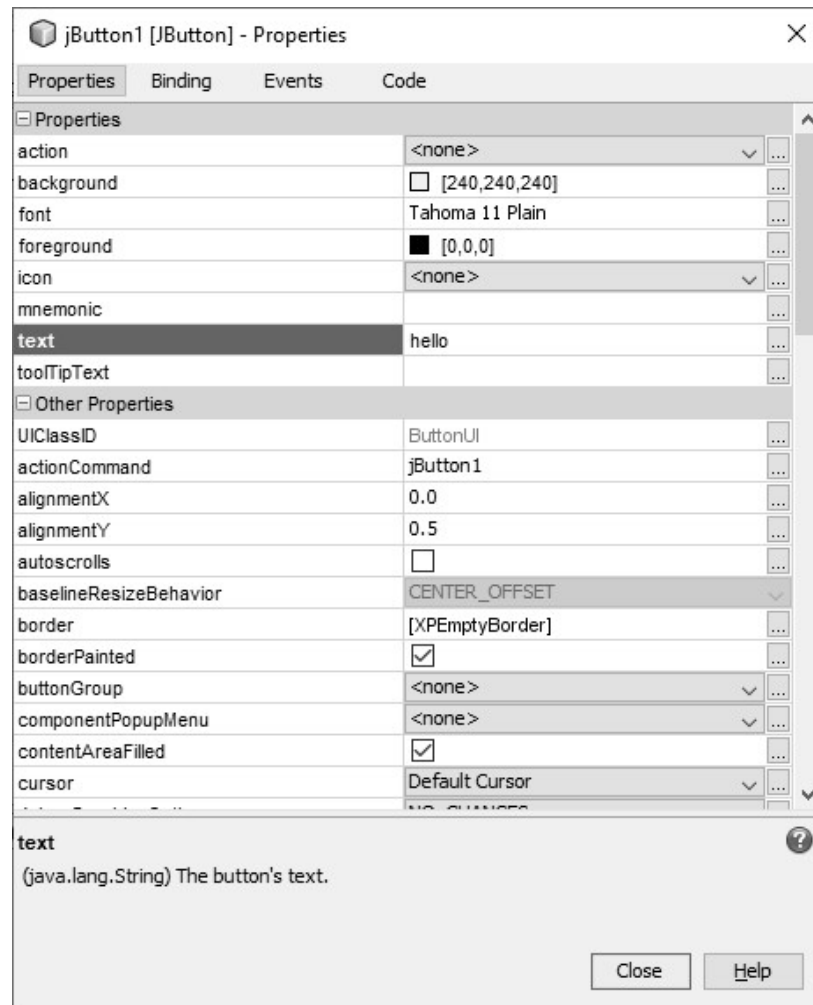
فيتم إضافته للمشروع و يظهر بالشكل التالي



يظهر الفورم الرئيسي المسمى MainForm.java في وسط الشاشة، وفي اليمين نلاحظ وجود عدد من المكونات في صفحة Palette ومن ضمنها الزر Button

نقوم بإدراج زر Button وسط الفورم الرئيسي، ثم تغيير عنوانه إلى Hello إما بالضغط على زر F2 ثم تغيير العنوان، أو بالنقر على الزر اليمين في الماوس على هذا الزر ثم نختار Properties ثم Text

الفصل الأول: مقدمة إلى لغة الجافا



الفصل الأول: مقدمة إلى لغة الجافا

نرجع مرة أخرى لنافذة الخصائص التي تظهر في اليمين والتي تتضمن عدد من المكونات في صفحة Palette ومن ضمنها الزر Events لنضيف حدث عند الضغط على الزر حيث نختار Events ثم في الخيار actionPerformed نختار الحدث jButton1ActionPerformed

عندئذ يظهر هذا الكود في شاشة Source

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent  
evt) { // TODO add your handling code here:  
}
```

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

أو يمكن إظهار هذا الكود بواسطة النقر المزدوج Double Click على الزر

يتم كتابة كود لإظهار عبارة (السلام عليكم) عند الضغط على هذا الزر

فيصبح الكود الحدث كالتالي :

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    String msg="السلام عليكم";  
    JOptionPane.showMessageDialog(null, msg);  
}
```

```
1 private void jButton1ActionPerformed(java.awt.event.ActionEvent  
   evt) { String msg = "السلام عليكم";  
2   JOptionPane.showMessageDialog(null, msg);  
3 }
```

ملحوظة

تم تعريف المتغير msg من النوع String ثم إسناد قيمة ابتدائية له "السلام عليكم" ، بعد ذلك نرجع للحزمة الرئيسية Mygui.java ثم نكتب الكود التالي في الإجراء main

```
public static void main(String args[]) {  
    MainForm form = new MainForm();  
    form.setVisible(true);  
    java.awt.EventQueue.invokeLater(new Runnable() {  
        public void run() {  
            new MainForm().setVisible(true);  
        }  
    });  
}
```

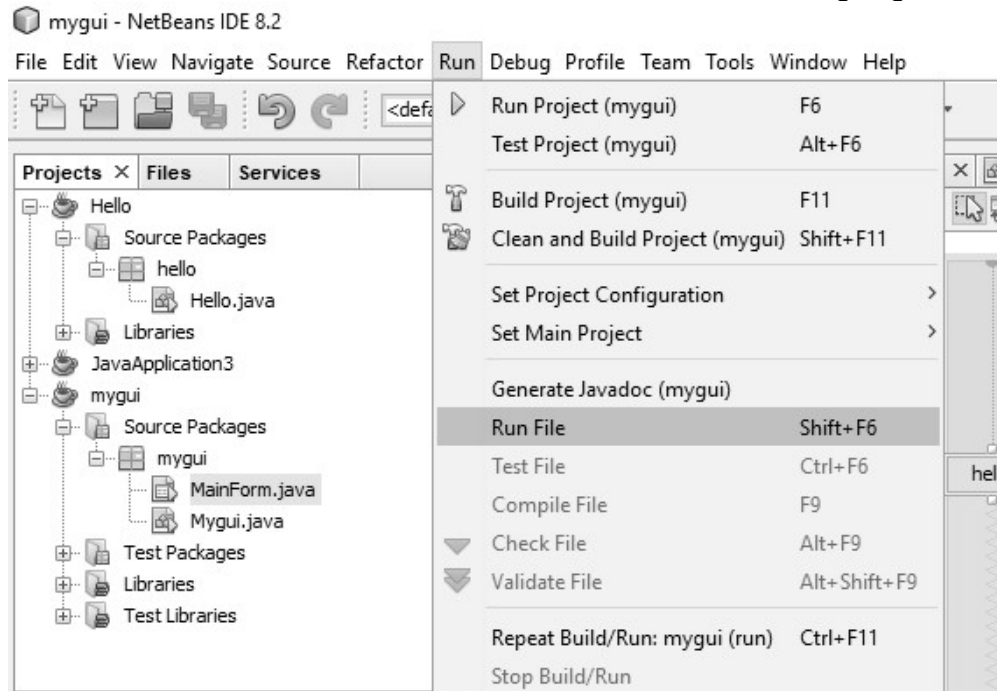
الفصل الأول: مقدمة إلى لغة الجافا

```
1 public static void main(String args[ ]) {  
2     MainForm form = new MainForm();  
3     form.setVisible(true);
```

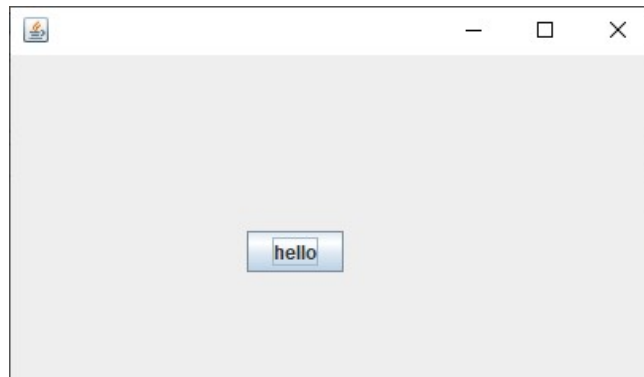
-السطر الثاني: نعرّف الكائن form من النوع MainForm الذي تم تصميمه ثم إنشاء نسخة من هذا النوع وتهيئته بواسطة `new MainForm`

-السطر الثالث: إظهار الفورم في الشاشة.

لتنفيذ البرنامج نفتح قائمة Run ونختار منها Run File



عندئذ تظهر واجهة البرنامج بالشكل التالي :



الفصل الأول: مقدمة إلى لغة الجافا

نضغط الزر Hello عندئذ تظهر الرسالة التالية:



نرجع مرة أخرى للفورم في شاشة Design

نقوم بإدراج الأداة TextField1 لندخل اسم المستخدم

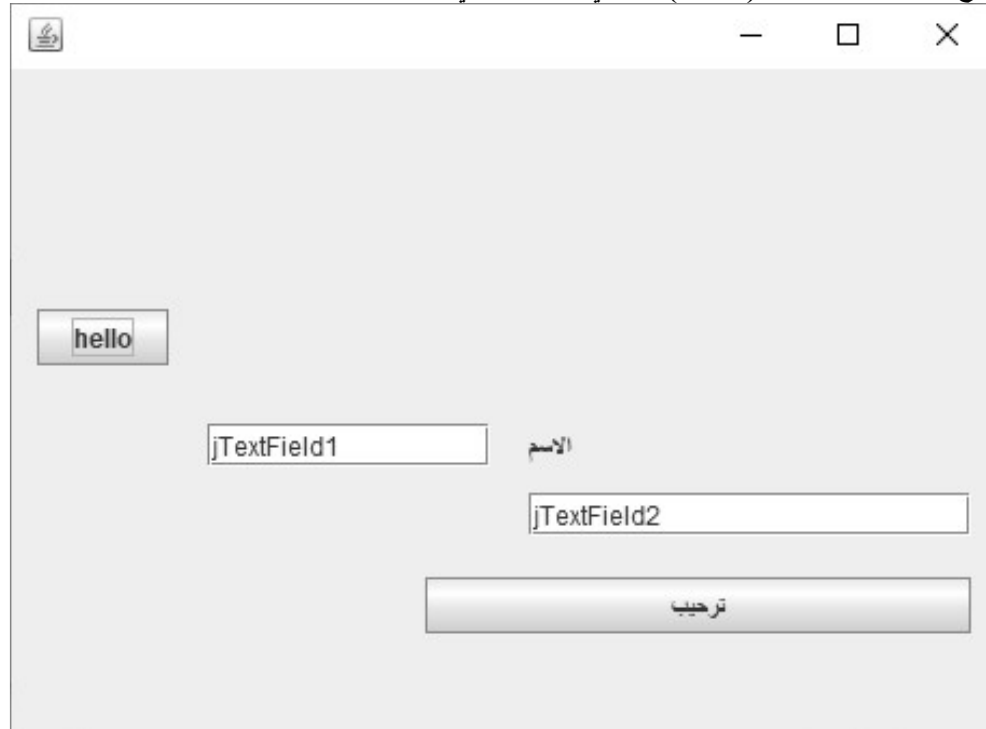
ثم أداة من نوع Label نكتب فيه كلمة (الاسم)

ثم مكون آخر من نوع TextField2 لتقوم بإظهار اسم المستخدم الذي تم إدخاله في الأداة
TextField1

ثم الانتقال إلى واجهة الخصائص وتنشيط التبويب Code والتعامل مع الخاصية Variable Name وتغيير الاسم ليصبح jlName



ثم ندرج زر نكتب فيه كلمة (ترحيب) كما في الشكل التالي:



في الحدث ActionPerfomed للزر Button2 نكتب الكود التالي لإظهار اسم المستخدم في الأداة TextField2

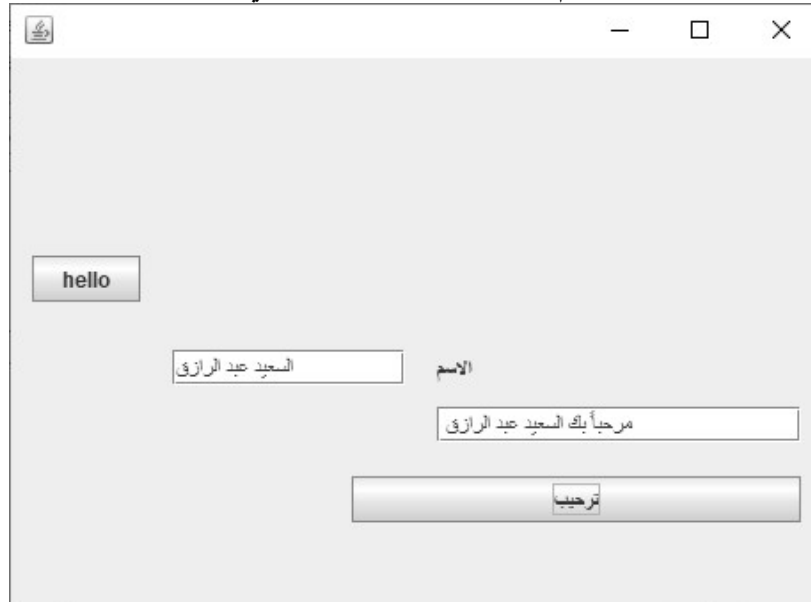
1	private void jButton2ActionPerformed(java.awt.event.ActionEvent
2	evt) { jlname.setText (" مرحباً بك " + jTextField1.getText()) ; }

```
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
    jlname.setText(" بك مرحبا " + jTextField1.getText());  
}
```

- الإجراء getText : يستخدم لقراءة محتوى الحقل النصي Text Field1
- الإجراء setText : يستخدم لوضع محتوى في الحقل Text Field2

الفصل الأول: مقدمة إلى لغة الجافا

تنفيذ البرنامج:
نفتح قائمة Run ونختار منها Run File لتظهر نافذة نكتب الاسم امام عنوان الاسم ثم نضغط
على الزر ترحيب عندئذ يظهر الاسم مسبقاً بكلمة مرحباً بك في الحقل TextField2 كما يلي:

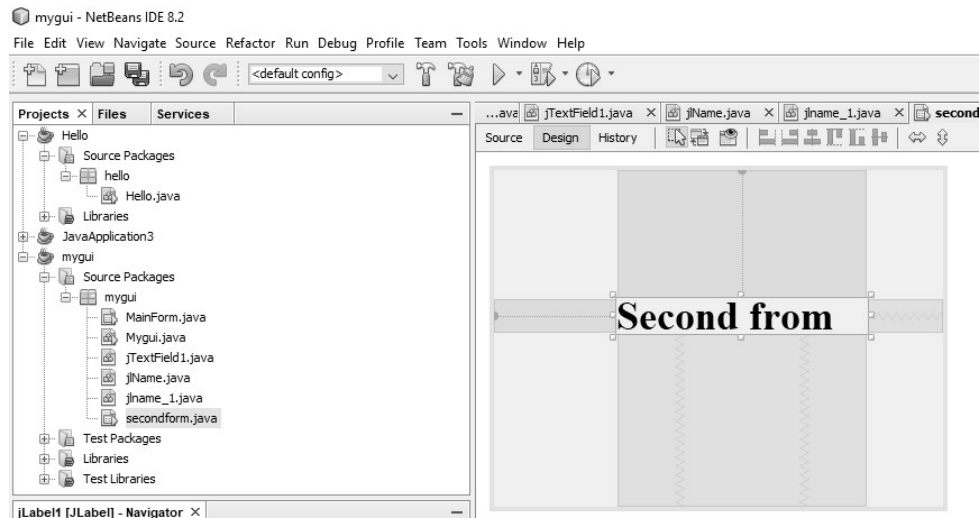
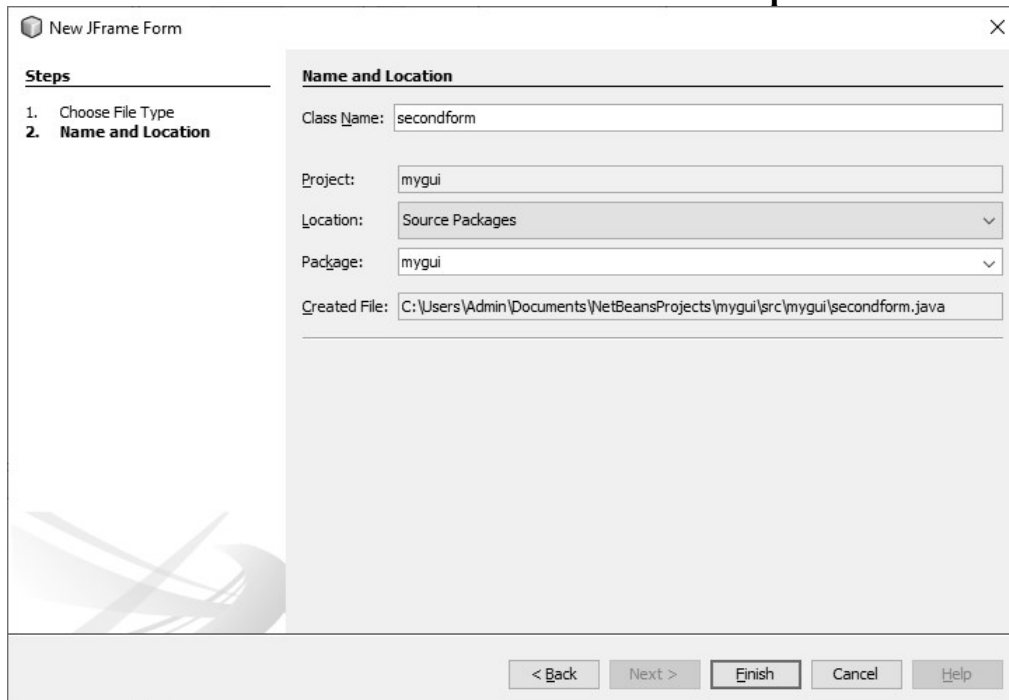


الفصل الأول: مقدمة إلى لغة الجافا

الفورم الثاني

لإضافة وإظهار فورم ثاني في نفس البرنامج، نتبع الخطوات التالية:

-إضافة JFrame Form (من خلال التعامل مع شاشة Projects ونختار الحزمة mygui ثم بالزر الايمن للماوس نختار New/JFrame Form) وتسميته SecondForm ونضع فيه Label نكتب فيه عبارة "Second Form" ونزيد حجم الخط في هذا العنوان (ليكون ٣٦) بواسطة .Properties/Font



الفصل الأول: مقدمة إلى لغة الجافا

في خصائص هذا الفورم الجديد نقوم بتغيير الخاصية `defaultCloseOperation` الى `DISPOSE` بدلا من `EXIT_ON_CLOSE` لأننا إذا تركناها في الخيار الأخير يتم إغلاق البرنامج عندما نغلق الفورم الثاني، وجرت العادة أن يتم إغلاق أي برنامج عند إغلاق شاشته الرئيسية، إغلاق الشاشات الفرعية يفترض به أن يقودنا إلى الشاشات الرئيسية

