

Mahmoud Fayed Book

Programming Secrets - Part [2]

أعمق أسرار البرمجة

الجزء الثانى



إهداء

إلى والدتي الحبيبة التي طالما تحملت الالم من اجل سعادتى
إلى كل عاشق للعلم ومحب للمعرفة
إلى المبرمجين الشباب من مصر والسعودية والمغرب والجزائر وسورية والعراق
ولبنان وفلسطين والإمارات والكويت والبحرين وقطر وعمان واليمن و جميع الدول
العربية والإسلامية الغالية على قلوبنا جميعا
إلى الزملاء ماهر محمود القط ومحمد عبدالستار الاصقيلي ومحمود عبدالعظيم
الاهونى واحمد سامى مرسى وهانى سعيد وتامر بدر القرط .
الى الاعزاء أحمد يحيى عمار- محمود يحيى عمار - أحمد عبدالمهيمن غانم - أحمد
كمال رزق - محمد كمال رزق - محمود محمد ظاهر - عبدالبارى محمد ظاهر - احمد
محمد ظاهر - شادى سعيد فايد
إلى الاخ الغالى (إبراهيم احمد إبراهيم دنيا) والاسرة الكريمة

شكر خاص :

م : سمير إبراهيم على فايد
م : محمد سمير إبراهيم فايد
د : فادى سعيد إبراهيم فايد
م : محمد سعود (مهدى)
م : مجدى محمد أبو العطا
م : أحمد عبدالعظيم غزية
م : احمد الكورانى
م : أحمد حمدى عرفة
م : محمد الهدهد
د : عمرو طلبية
د.م : صلاح شعبان
د.م : إيهاب عزيز خليل
د.م : حمدى قلاش
د.م : جمال محروس
د.م : محمد شعلة
د.م : محمد امون
د.م : أيمن السيد
د.م : مرفت موسى
د.م : هدى سرور

المحتويات :-

● مقدمة الجزء الثانى

- كيفية الاتصال بالمؤلف 5
- أهداف الجزء الثانى 5
- كلمة المؤلف 5

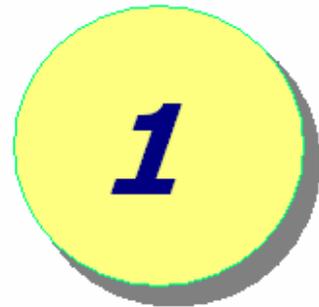
● احدث تكنولوجيا البرمجة

- فلسفة فن البرمجة 10
- تكنولوجيا معاصرة 27
- البرمجة بدون كود RPWI 72

● سمات التطبيقات المتطورة

- السمات المميزة للتطبيقات 128
- أدوات برمجة النظم 201
- فن برمجة النظم 206

مقدمة
الجزء الثانى



كيفية الاتصال بالمؤلف

Picture	
Name	Mahmoud Samir Ibrahim Fayed
Age	20
Job	Programmer
Email	msfclipper@hotmail.com & msfclipper@yahoo.com

أهداف الجزء الثانى

- دعم الرقى الفكرى للمبرمجين من خلال التعرض لفلسفة تكنولوجيا البرمجة
- دعم الرقى الفنى للمبرمجين من خلال التعرض لاساليب فن برمجة النظم
- التعرض لاحدث التكنولوجيا المعاصرة
- التعرض لسلمات التطبيقات المعاصرة

كلمة المؤلف

اعتاد المبرمجين فى الشرق الاوسط على قراءة الكتب العربية - فقط فى المراحل الأولى لتعلم البرمجة - ولكن اذا ما ثبتت اقدمهم فى المجال وتبين لهم الطريق فانهم سرعات ما ينصرفون تماما عن الكتب العربية ويلجأون الى الكتب الغربية بمختلف لغاتها الانجليزية او الفرنسية او الاسبانية او غيرها- وذلك نظرا لكونها مراجع اصلية كاملة لتكنولوجيا البرمجة - وهذا ماتفقده الكتب العربية - حيث انها كتب وليست مراجع - وللمؤلفين العذر

فى ذلك لان اعداد المراجع الكبيرة امر مكلف يحتاج الى وقت طويل ودعم مادي لايتوفر بسهولة.

كما ان اتقان المبرمجين العرب للغة الانجليزية امر شائع مما يسمح لهم بسهولة الاندماج مع العالم الغربى فيمكن للمبرمج العربى قراءة الكتب او المقالات الانجليزية ويمكن ايضا له المشاركة فى المنتديات والمشاريع مفتوحة المصدر لتبادل الخبرات وتحقيق الفائدة المشتركة.

ولعل البعض يسأل طالما تعلمنا اللغة الانجليزية - فلماذا نكتب الكتب العربية وخاصة ان اللغة الانجليزية هى لغة العلم ومن المستحيل نظريا ترجمة جميع الكتب الانجليزية الى اللغة العربية وخاصة انها ملايين الكتب - كما ان الاستمرار فى البرمجة لمن ليس لهم دراية باللغة الانجليزية امر شبه مستحيل - يكاد يكون صعب جدا - فى الحقيقة ان توفر الكتب العربية يعنى سرعة فى التعلم - ويتيح الفرصة للشباب صغير السن من هو دون السادسة عشر ولايجيدون الانجليزية بطلاقة - وحيث ان الشباب صغير السن بحاجة الى كتب للمبتدئين فتجد ان ٧٠% تقريبا من الكتب العربية تخاطب تلك الفئة - وكتب المبتدئين هى الاكثر مبيعا فى سوق الكتب العربية - لهذا عندما قررت الكتابة وبالتحديد كتابة الكتب المجانية بدات بكتاب (أسرع طريق لتعلم البرمجة) وهو فى الحقيقة عبارة عن كتيب صغير الحجم لكنه دسم بالمعلومات بالنسبة للمبتدئين وفى الحقيقة حقق هذا الكتيب الصغير نجاحا منقطع النظير لم اكن اتوقعه ابدا حيث وصل عدد القراء للكتيب الى اكثر من ٢٥ ألف قارئ من مختلف الدول العربية مما يدل على الاقبال على عالم البرمجة وخاصة ان اسم الكتاب مغرى جدا مع العلم انه يستند على لغة Visual FoxPro وهى لغة ليست شهيرة مثل لغة Java اللغة الاولى فى العالم - او لغة C اللغة الثانية فى العالم او لغة ++C اللغة الثالثة فى العالم - او لغة Visual Basic اللغة الرابعة فى العالم من حيث عدد المبرمجين والاشخاص المهتمين باللغة - ولعلك تنصدم اذا علمت ان لغة Visual FoxPro هى اللغة رقم ٢٠ على مستوى العالم من حيث عدد المبرمجين باللغة على الرغم من انها اللغة الاولى فى العالم لتطوير انظمة قواعد البيانات ويقول عنها الخبراء انها Hidden Gem in Microsoft Products اى الجوهرة الخفية فى منتجات مايكروسوفت.

وبعد تجربة كتاب (أسرع طريق لتعلم البرمجة) جاءت فكرة تقديم كتاب يخاطب المبرمج المحترف مما قد يساعد على إعادة ثقتهم بالكتب العربية وهنا قد نقف قليلا عند نقطة الاحتراف - لتحديد عند اي مستوى هذا الاحتراف ولكن بعيدا عن الفلسفة يمكن ان اقول لك ابتعد تماما عن هذه الكلمة وتمسك بكلمة (المبرمج) لان الكتاب يخاطب Programmers وليس المطورون Developers - قد تكون مبرمج Programmer وتعمل كمطور Developer تقوم باستخدام الادوات الحديثة - لكن اذا كنت Developer فمن الصعب ان تكون مبرمج Programmer وسوف يحتاج الامر منك لمجهود حتى تحقق ذلك

ان الكتاب يخاطب (المبرمجين) المحترفين فى اعداد التطبيقات التجارية - هنا قد حددنا مستوى الاحتراف وهو القدرة على انتاج تطبيقات تخدم السوق فى الشرق الاوسط والذى يركز بنسبة ٩٠% على التطبيقات التجارية والنسبة الباقية للبرامج التعليمية والوسائط المتعددة Multi Media و المواجهة بالحاسب Computer Interfacing

والمستوى الجديد الذى يقدمه الكتاب هو الدعم الفنى والعلمى الكافى لتصميم وبرمجة مختلف انواع النظم وقد تعرض الجزء الاول من الكتاب الى نموذج سير العمليات ونمط البرمجة وواجهة المستخدم بايجاز - لانه لو اتبعنا التفصيل فقد يصل حجم الكتاب الى ثلاثة اضعاف حجمه مما يسبب إرهاق للقارىء وضغط لايتحمله لان المفاهيم معقدة والايجاز مطلوب حتى لاتشوش الافكار.

اما هذا الجزء فانه يشمل الجانب النفسى للمبرمج - ان هذا الجزء يعرض افكار فلسفية رائعة تريح المبرمج من اي ضغط - كما انها تجيب على الكثير من الاسئلة التى تحير المبرمجين مثل (هل اتعلم اكثر من لغة ؟ ماهى افضل لغة ؟ هل اعمل بالادوات المجانية ؟ ماذا عن السوق ومتطلباته ؟ اشعر بالضعف امام المشاريع مفتوحة المصدر ؟ كيف اصل الى اعلى مستويات الاحتراف ؟ هل انا مبرمج محترف ؟ العمر يتقدم ومازلت مبرمج عادى ! كيف اكون مبرمج مهم ! كيف اكون مبرمج عالمى !)

وبالتاكيد ليس الهدف هو الفلسفة وانما الهدف هى استخدامها للوصول للاجابات التى لا بد ان تصل اليها قبل الخوض فى معركة برمجة النظم

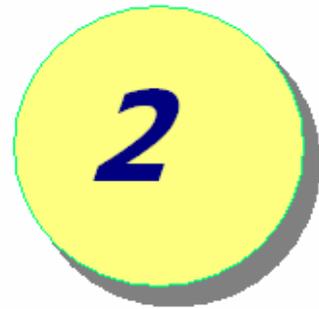
وبالتاكيد الجزء الثانى ليس كله عبارة عن فلسفة والا انصرف عن الكتاب كل قارىء لانى اعلم جيدا ان المبرمجين لا يقتنعون بالحوار

مهما كان مهما وانما لايرضى عقولهم سوى الاسطر البرمجية او رسومات التصميمات القيمة ولهذا اتحدث عن فن برمجة النظم وكيف تقوم ببناء نظام بصرف النظر عن الهدف من وراءه - بحيث يكون النظام اشبه بالحصن من حيث الاستقرار - واشبه باللوحة الفنية من حيث الابداع - واشبه بناطحات السحاب من حيث الحجم.

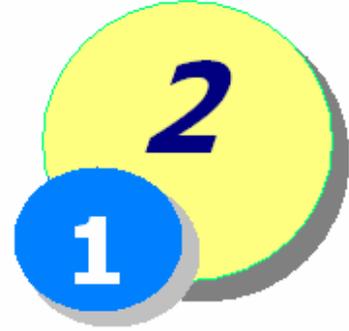
ولعلك تسال الان هل يمكن لمبرمج واحد ان يحقق ذلك - تكون الاجابة نعم مبرمج واحد يمكنه ان ياسس امبراطورية ولكنه لايمكنه ان يقوم بنائها منفردا إلا ما ندر.

كما يستعرض الكتب ابرز التكنولوجيا المعاصرة فى تطوير التطبيقات وسمات التطبيقات المعاصرة - ثم ينصرف بك بعيدا عن هذه التكنولوجيا ويعرض لك الجانب الاخر من اللعبة - الجانب الخفى وهو تكنولوجيا تطوير النظم وليس التطبيقات حتى تلاحظ الفرق الشديد ولتجمع منها اكبر قد ممكن لانها العدة والسلاح الذى سوف تستخدمه فى المعركة - معركة برمجة النظم بمختلف أنواعها.

احداث تكنولوجيا البرمجة



فلسفة فن البرمجة



قد يتسال البعض وربما الكثير - منذ متى والفلسفة تدخل فى علم البرمجة - فنرد ببساطة - ان جذور علم البرمجة تعود الى الفلسفة وتعود الى المنطق وتعود الى الرياضيات والكيمياء والفيزياء والاحياء ومعظم العلوم الاخرى - حيث ان علم البرمجة نشأ معتمدا على جميع العلوم - ثم بعد ذلك قام برد الجميل وساهم فى تطور هذه العلوم تطورا كبيرا.

ان دراسة الفلسفة - وخصوصا فلسفة فن البرمجة ليست من اجل العلم بالشئ وانما بغرض رفع المستوى الفكرى للمبرمجين من حيث القدرة على اتخاذ القرار المناسب بسرعة - ان المبرمج ليس مجرد شخص يستطيع التعامل مع الحاسب ويجيد تسخيرها - فان الامر اكبر من ذلك بكثير - ان المبرمج شخص متعدد الادوار حيث انه علاوة عن كونه مبرمج فانه قائد قد يدير مجموعة من المبرمجين الاخرين الذين يعملون معه فى المشروع وقد يكون فرد من افراد هذا الفريق عليه واجبات والتزامات ينبغى ان يفي بها - وقد يكون محلل يتابع تغيرات السوق ليحدد متطلباته ويغزو السوق بافكاره المنتجة وقد يكون عالم عليه ان يحدد جيدا ماذا ينتقى من العلم لكى يتعلمه ومايجب الا يجهله من التكنولوجيا.

خلاصة الامر ان البرمجة (قيادة) للحاسب وللمشاريع وللأفراد و(تخطيط) للمستقبل و (ادراك) لمتطلبات الواقع والبيئة المحيطة و (علم) بمختلف فروع البرمجة و (فن) يستلزم القدرة على الابداع وكل ذلك لابد ان يستند على (فلسفة) تدير حركة كل هذه المتطلبات(القيادة - التخطيط - الادراك - العلم - الفن) معا.

ملاحظات

القيادة تعنى الادارة

التخطيط يعنى التصميم

الادراك يعنى القدرة على التحليل

ونلاحظ ان هذه احد العناصر الاساسية لتصميم وتطوير البرمجيات

ويجدر بنا الان التعرض لمجموعة مختارة من الدروس الفلسفية
التي تلائم المبرمجين.

دراسة سقوط المطر

ان المطر حالة فنية رائعة ونعمة كبيرة لاينكرها احد وهو احد
اسباب استمرار الحياة على الارض - واذا تابعا المطر جيدا -
سوف تلاحظ انك ان فكرت فى متابعة قطرة واحدة فانك ريثما
تنتهى من اتخاذ القرار وتبدا فى المتابعة - فاذا بالقطرة قد
سقطت على الارض وانتهى امرها - اى ان فكرة متابعة قطرة
واحدة تنتهى بمجرد التفكير فيها.

واذا نظرت الى امد بعيد - على مرمى البصر ووجدت قطرات الماء
تسقط امامك فانك تنبهر بالمتظر الجمالى ولا تشعر باى خوف -
فقط المتعة حتى انك تنجذب نحو المطر ولا تتردد فى السير تحته
الا اذا كانت والدتك عودتك منذ الصغر الا تفعل ذلك خوفا على
الملابس او خوفا على صحتك - ولكن مع هذه المتعة تجد ان
ذاكرتك لم تلاحظ قطرة واحدة بعينها طوال رحلة سقوطها.

اذا نظرت الا ارض اثناء سقوط المطر الغزير فانك تجد ان الماء قد
تجمع على الارض خاصة اذا كان الطريق غير مجهز لذلك بمعنى
انه يحجز الماء - عندها قد يشعرك ذلك بالقلق - وربما الخوف اذا
زاد مستوى الماء لدرجة تمنعك من المشى - وربما وصلت
لدرجة من الرعب اذا زاد مستوى الماء لدرجة انك اصبحت فى
بحر.

والسؤال الان : ما رايتك فى هذا الكلام ؟

هل تشعر ان هذا يفيدك كمبرمج فى اى شىء ؟

هل ترى فى قطرات المطر شىء تحسه فى البرمجة ؟

فى الحقيقة اذا كنت تحب الفلسفة مثلى وتعشق البرمجة فانك من المؤكد انك قد وصلت الى ماقصده

عزيزى القارىء ان قطرة المطر هنا تمثل سطر برمجي واحد - يمكن ان تقرأه ضمن اكواد اى نظام او برنامج.

ولهذا اذا فكرت فى قراءة اى سطر برمجي فانه بمجرد الانتهاء من الفكرة - فان الامر تقريبا قد انتهى حيث ان عملية فهم اى سطر على حده لا تتجاوز بضع ثوانى وخاصة لمن يتقن اللغة ويحترفها

وهذا يعنى انك ان فتحت شغيرة مصدرية Source Code قد قام بكتابتها مبرمج اخر واتبعت اسلوب قراءة كل سطر على حده - او مايقال (الاطلاع على الكود - بسرعة) فانك لن تشعر بيقمته وخاصة اذا لم تصدم باشياء لاتفهمها.

وذلك يذكرنى بما كنت افعله وانا فى الحادية عشر من عمري حيث كنت امتلك مرجع لاوامر لغة البرمجة كليبر تاليف الاستاذ مجدى محمد ابو العطا - وكنت بمساعدة هذا المرجع اطلع على البرامج التى كان قد قام ببرمجتها والدى باستخدام نفس اللغة - وحينها كنت كلما نظرت الى كل امر على حده وبمساعدة الكتاب اشعر ان الامر فى غاية البساطة - بصراحة كانت متعة الطفولة حيث كنت اشعر ان الامر ليس صعبا.

ذلك الامر ايضا يشبه ماكنا نفعله عند دراسة مادة الرياضيات وبالاخص الاثباتات حيث كنا ننظر الى الاثبات فنرى ان كل القوانين التى تم استخدامها فى الحل هى قوانين وردت فى المنهج وهذا امر طبيعى ولكن ذلك كان يجعلنا نشعر ان الامر فى غاية البساطة وسهل.

والان اذا نظرنا على الدوال التى يشملها النظام او الفصائل التى يحتويها فاننا نشعر بمتعة وربما بساطة الامر ولكن الخلاصة انك لم تستنج محتويات اى دالة بصورة كاملة او اى فصيلة بصورة كاملة وذلك شبيه بالنظر الى الامد البعيد تحت المطر.

وإذا نظرنا الى عدد الاسطر البرمجية او مساحة الشفيرة المصدرية فاننا قد نشعر بالرغبة فى النوم خاصة عند الاطلاع على المشاريع الكبيرة التى قد تصل الى الالاف من الاسطر البرمجية - وذلك يخيف خاصة انها نظم تتسم بالتعقيد - وليس فقط الحجم الكبير وذلك اشبه بالنظر الى قطرات المطر على الارض والتى تجمعت لتكون ماء مرتفع قد يشكل خطورة.

والسوال الان : ماذا افعل - انظر الى ماذا ؟؟

فى الواقع لا تتطلع اساسا للشفيرة المصدرية الا بعد ان تتاهل لذلك - حيث ينبغى عليك اولا

- ✚ التعامل مع النظام كمستخدم
- ✚ القدرة على تخيل تصور كامل للنظام - كمستخدم
- ✚ محاولة تحليل وتخيل مكونات النظام معتمدا على فهمك له
- ✚ النظر من خلال الامد البعيد اولا ولكن فى الاتجاه الذى تراه مناسباً

- ✚ النظر الى كل قطرة على حده فى نفس اتجاه الامد البعيد الذى قد حددته فى الخطوة السابقة
- ✚ تكرار الخطوتين السابقتين حتى تلم بالنظام

عليك ايضا ان تعلم جيدا ان طريقة دراسة النظم تختلف بالهدف من الدراسة - فقد يكون الهدف

- ✚ التعلم
 - ✚ صيانة النظام من الاخطاء
 - ✚ تطوير النظام
- فى حالة الرغبة فى التعلم فانك تتعلم الجزء الذى تريده بحيث ترفع مستواك وقد تترك بعض الاجزاء التى تراها صعبة للغاية ويستحيل التعامل معها فى الوقت الحالى فى حالة الرغبة فى صيانة النظام من الاخطاء فان تركز على تعلم الجزء الذى به الخطأ وقد تصلح الخطأ باكثر من اسلوب مثل اعادة كتابة اجزاء كاملة من جديد باسلوبك الخاص - او تصحيح ماكتب سابقا وذلك يتطلب فهم كامل اما تطوير النظام فهذا امر غاية فى الخطورة حيث انك قد تطور جزء من النظام ببراعة ويشكرك الجميع على ذلك فى البداية ثم تفاحىء انك قد تسببت فى عطل بقية اجزاء النظام الاخرى.

ان العقيدة هى كل مايؤمن به الانسان ويعتقد فيه ولا يبدله ابدا والدين هو اسمى درجات العقيدة لدرجة انه عندما تذكر هذه الكلمة فان المعنى الذى يتبادر الى اذهاننا جميعا هو الدين والايمان - ويرى علماء الفلسفة ان الانسان يحيى ويعيش فى سلام واستقرار اذا كانت له عقيدة قوية لاتهتز بتغيرات العصر وما يحدث فى الواقع من تبديلات للامور ويقال ايضا انه اذا غير الانسان عقيدته فهذا اعلان لتغير جذرى قد طرأ بهذا الانسان بصرف النظر عن هل هذا التغير حسن ام غير حسن - وتكمن القضية فى ان الانسان يرفض تغير العقيدة - وهذا امر طبيعى ولهذا فان الانسان غالبا وبنسبة ٩٩% يكون على دين اسرته - وللعقيدة سر خطير واثر بالغ فى نفس الانسان الذى يعتقد فيها حيث انها تكون جذوره واغلى شىء يتمسك به واذا كان لدى الانسان شىء اهم من اياه وامه وزوجته واولاده فانها العقيدة لا مفر - ويلبها بالتاكيد فى الدرجة الوطن واللغة حيث ان الوطن احب مكان لا شك الى نفس الانسان واللغة الاولى التى تربي عليها بالتاكيد هى احب لغة اليه - وان تعلم كل لغات العالم فانه يظل يفكر باللغة الاولى التى رباها عليها اهله ومن نشأ معهم منذ الصغر.

والسؤال الان : ما دخل العقيدة فى البرمجة ؟

فى الحقيقة لها دخل كبير - لقد ذكرنا ان الدين اولى درجات العقيدة والتى اقصد بها كل ما يتمسك به الانسان - يلى ذلك الوطن واللغة - وبالنسبة للمبرمجين فانهم فى خطر كبير حيث ان كل مايفعله الانسان فى يومه يؤثر فى تكوينه والمبرمج المحترف انسان يقضى اكثر من ٥٠% من وقته امام الحاسب ولهذا لعالم البرمجة تاثير داخلى رهيب على المبرمج الذى قد تكون البرمجة عقيدة بالنسبة له - لا يستطيع ان يتركها ابدا - فقد تحولت الى ادمان - وقد تكون لغة البرمجة التى يبرمج بها هى عقيدة بالنسبة له - يدافع عنها بشراسة - بل ويهاجم كل اللغات الاخرى بدون دراية بها اساسا - وقد يتحيز المبرمج لنظام تشغيل فيكون عقيدة بالنسبة له يدافع عنه وان كان به من العيوب ما به - و المطلوب من كل مبرمج ان لاتتمكن من قلبه لغة

البرمجة التى يجيدها بل يتغلب على ذلك ويقبل الرأىء الاخر ولا يرفض النقض - بل يحاول ان يطور نفسه - فمن المطلوب بعد اتقان لغة برمجة ان تتعلم لغة برمجة جديدة كل ٢-٤ اعوام بحيث تتابع التطور وتنمى مهاراتك لان تعلم لغة برمجة جديدة افضل بكثير من تجميع عشرات المكتبات للحصول على مميزات تلك اللغة الجديدة فى اللغة التى تستخدمها.

البرمجة مهنة - اجعلها وظيفة وليست عقيدة واقبل ان تخرجها من حياتك ان لزم الامر (اقصد اجعل فكرك يقبل ذلك ولكن هذه ليست دعوة لترك البرمجة - على العكس تماما ولكنها دعوة للتخلص من ادمان البرمجة)
نظم التشغيل ولغات البرمجة - اجعلها ادوات يمكن ان تغيرها فى اى وقت لتحقيق هدفك اما بكفاءة اكبر او بسرعة اكثر او بسهولة.

اذا اردت ان تستمتع بهذا الامر فسوف تجد فى المنتديات ومواقع الانترنت العديد من المعارك الطاحنة بين المبرمجين كل منهم يتحيز للغة البرمجة التى يجيدها ويتحدث عن مميزاتا ويظن انها غير موجودة الا فى اللغة التى يعرفها والعجيب ان اغلب المميزات التى يذكرها المبرمجين تكاد تكون المواصفات الاساسية لاي برنامج او نظام يستحق ان يطلق عليه لغة برمجة.

يحضر فى ذهنى الان مقولة شهيرة تناسب هذا الامر

ليس الجندي بالسلاح وانما السلاح بالجندي

الشعراء ومولفين القصص

من منا لم يقرأ او يدرس شعرا - من منا لم يسمع عن المتنبي وابى فراس الحمدانى ومحمود سامى البارودى واحمد شوقى وابراهيم ناجى - وغيرهم الكثير من الشعراء - ومن منا لم يقرأ قصة لنجيب محفوظ او طه حسين او فيصل ندا او غيرهم من

الكتاب المبدعين - لو فتحت كتابا به قصة او شعر - لوجدت ان المادة واحدة - لغة عربية من ٢٨ حرفا فقط - و قواعد نحوية ثابتة وكلمات لغوية محددة ومعروفة ومسجلة فى المعجم - ولكن الخيال لا يمكن ان يقيد ابدا ولا يمكن ان يكون له حدود ولهذا رغم ثبات الادوات من خروف وقواعد وكلمات الا ان الابداع مختلف من شخص لآخر وترى احيانا توارد فى الافكار ولكن الاختلاف يظل موجود.

والسؤال الان : ما علاقة ذلك بالبرمجة ؟

عزيزى المبرمج انت تقوم بدور الشاعر او مؤلف القصص ولكن باسلوب اخر - بدلا من استخدام لغة البشر فانك تستخدم لغة البرمجة وبدلا من كتابة الشعر او القصص فانك تقوم بكتابة البرامج.

عفوا - ما الفائدة من هذا الكلام ؟

حسنا تخيل لو انك قرأت قصيدة للشاعر احمد شوقى ثم جاء صديق لك وقال انه كتب قصيدة - فعندما جئت لتقرأها وجدت انها قصيدة احمد شوقى - فما موقفك من زميلك - بالتأكيد سوف تقول له انت سارق

ولو وجدت انه كتب قصيدة من ١٠٠ بيت وكل ١٠ ابيات قام بكتابتها شاعر مختلف - فهنا قد تقول له انت حرامى كبير - لكن حركة حلوة - بس انسب لكل واحد ابياته وقول انك مجمعها بس.

لكن لو وجدت القصيدة التى كتبها زميلك كلها من ابداعه - من المؤكد سوف يختلف رايك حسب ابداع زميلك - ولكن الابداع فى حد ذاته مهم ومطلوب وان كان الالهام ان يكون جميل.

عدنا للفلسفة من جديد - اين هذا فى البرمجة ؟

ان الشيفرات مفتوحة المصدر هى ابيات الشعر القديمة لذلك حفظك لها واستخدامها لا يعد ابداع - الا اذا جمعت من اكثر من مصدر - ولكن الابداع الحقيقى ان تكتب من فكرك.

ان الدول العربية سميت عربية لان سكانها يتحدثون باللغة العربية وان كنا نعلم ان الاسماء لا تعلق - فقد يتصادف وان تعلق بعضها - ان اهل الصين لهم لغتهم الخاصة والتي يمكن ان تكتب من اليمين الى اليسار او من اليسار الى اليمين او من اعلى لاسفل او من اسفل لاعلى - لغة مرنة - ولسكان فرنسا لغتهم الفرنسية وقد تجد فى بعض البلاد تنتشر مثل اكثر من لغة ولكن السؤال الان هل يمكن لبشر ان يتعلم كل لغات العالم ؟

نعم ممكن ! اذا كان نبيا - فقد كان نبى نوح عليه السلام يعرف كل اللغات الموجودة فى عصره - حيث انه ذكر فى كتاب قصص الانبياء لابن كثير انه بعد الطوفان قد اصبح لكل زوجين لغة يتحدثون بها وكان يترجم بين البشر سيدنا نوح وبالطبع زمن النبوة قد انتهى ونحن بشر عاديون وقد يكون عظيما منا من يجيد اكثر من لغتين ببراعة وان كنت سمعت عن اناس يعرفون ٧ لغات.

والسؤال الان : ما علاقة ذلك بالبرمجة ؟

ان المدة اللازمة لاتقان لغة برمجة تعادل نصف المدة اللازمة لاتقان لغة بشرية - ومن يفكر فى تعلم اكثر من ٥ لغات برمجة فانه فى الواقع مغامر ويستحق ان يسجل اسمه مع العباقرة - فى الحقيقة تعرفت من خلال الانترنت على اناس تعلموا اثنى عشر لغة برمجة خلال ٢٢ سنة فى مجال البرمجة منذ عام ١٩٨٤ حتى عام ٢٠٠٦ ولكن تعلم اكثر من لغة لا يعنى احتراف جميع هذه اللغات - فمثلا قد تجد من يتحدث العربية والانجليزية والفرنسية والاسبانية ولكنه يستطيع ان يكتب القصص والشعر بالعربية وكذلك الانجليزية ولكنه لا يستطيع كتابة الشعر بالفرنسية ولا يستطيع كتابة الشعر او القصص بالاسبانية وهناك من يجيد هذه اللغات الاربعة ولا يعرف اساسا كتابة الشعر او القصص بها وحينها ليس له الا ان يعمل مترجما.

الخلاصة

✚ لا تفكر فى تعلم لغة الا اذا وجدت الاسباب الكافية لذلك
✚ ليس المهم ان تتعلم كل اللغات - المهم ان تتقن اللغات
التي تتعلمها
✚ الابداع هو العلامة الوحيدة لاتقانك للغة البرمجة

ولعلها فرصة لذكر القواعد الاساسية التي ينبغي تعلمها فى اى
لغة برمجة تفكر فى استخدامها وقد يحدث ان لا تتوفر بعض هذه
الخيارات

- ✚ Framework (Optional)
- ✚ IDE (Optional)
- ✚ Project Manager(Optional)
- ✚ Designers (Optional)
- ✚ Editors (Optional), use notepad.exe or edit.exe
- ✚ Debugger
- ✚ Documentation (Help)
- ✚ Compiler
- ✚ Linker
- ✚ Library Maker
- ✚ Make utility
- ✚ Code
 - Data Structure
 - Control Structure
 - Programming Paradigm
 - Structure Programming
 - Object Oriented Programming
 - Programming Style of language
 - Basic functions
 - Basic Classes
- ✚ Writing your own library
- ✚ Using others library
- ✚ Extension using other language (C-API)
- ✚ Calling OS Interface (WIN32 API for example)
- ✚ Accessing Low Level resources (CPU registers, Memory & I/O Devices)

ان ادارة المنشآت والمصانع امر فى غاية الصعوبة وبحاجة فعلا الى عقلية مفكرة - ومن قصص النجاح المشهورة ان يدير الطبيب المبدع المستشفى بمنهى البراعة - وان يدير المبرمج المحترف شركة لتصنيع البرمجيات بمنتهى الكفاءة وان الادارة الناجحة تسلزم معرفة جيدة وربما احيانا معرفة كاملة بادق تفاصيل المنشأة التى يتم ادارتها.

ما علاقة ذلك بالبرمجة - هل تقصد المعنى المباشر - بمعنى ادارة شركات البرمجة؟؟

بالتاكيد لا وانما اتحدث عن المبرمجين الذين يطورون تطبيقات بلغات عالية المستوى - تخيل مثلا مبرمج بلغة سى C وعلى مستوى على من الاحتراف واخر مثله ولكن يتميز بخبرة فى لغات من مستوى اقل مثل لغة الاسبلى - بالتاكيد ان المبرمج الثانى يتفوق على الاول

وكذلك مبرمج بلغة من الجيل الرابع مثل فيجوال فوكس برو او كليبر او اوراكل - اذا كان يحترف اللغة وهناك مبرمج اخر يحترفها مع العلم بلغة من لغات المستوى الاقل مثل لغة سى فبالتاكيد ان الثانى يتفوق - واذا كان هناك مبرمج ثالث يحترف اللغة ويجيد لغة من مستوى ادنى مثل لغة سى ولغة اسمبلى ايضا - فان المبرمج الثالث يتفوق ويتميز

وبالمثل مبرمجين لغات الجيل الخامس من اللغات الطبيعية وانظمة الذكاء الاصطناعى من يتقن لغات ادنى فى المستوى من هذه اللغات فانه يتميز وتكون نسبة الابداع اعلى.

المبرمج بلغات عالية المستوى اشبه بالمدير - الذى يكون اكثر نجاحا ان كان ملما بتفاصيل لغات المستويات الادنى

ان للبيئة تاثير كبير على تكوين شخصية الانسان - فغالبا ماينشأ العلماء والباحثين فى وسط ملئ بالمفكرين والمبدعين - وغالبا ما ينشأ المجرمين فى عالم مشهور بالجريمة - وغالبا ما يكون الاغنياء قد استمدوا اساس اموالهم من اسلافهم (الاباء والاجداد) مع العلم والايمان بالتاكيد ان الطفرة قد تحدث وينشا العالم من ظهر فاسد ويولد المجرم فى اسرة طيبة وينشا الغنى فى اصعب ظروف من الفقر.

ان تحقيق الطفرة امر غاية فى الصعوبة ولكنه ممكن. ايضا قد ينشا الانسان بصورة سليمة ولكن المجتمع يدفعه ويثر فيه الى غير مانشا عليه - فكم من مجرم بغير قصد او سابق اصرار.

والسؤال الان : ما علاقة ذلك بالبرمجة ؟

ان البرامج كائنات مثل الانسان تماما ولكن لها عالمها الخاص وبيئتها المحيطة التى تتاثر بها وتؤثر فيها - فقد تكون مبرمج مبدع تود عمل برنامج مميز - الا انك تعمل فى بيئة ضعيفة مثل DOS على سبيل المثال الذى اصبح قديما - قد تراكم التراب عليه - اذا رغبت فى عمل شىء جيد تحت هذا النظام فانك بلا شك تبحث عن الطفرة والتى ان حققتها فقد فعلت معجزة.

لايقف الامر على نظام التشغيل بل يمتد ليشمل ادوات البرمجة فكم من اداة برمجية ضعيفة تسبب العناء للمبرمجين مما يجعلهم يفشلون فى تحقيق اهدافهم ومن ثم ينصرفون بعيدا عن تلك الادوات

ان ماارت قوله عليك ان تحدد البيئة جيدا من نظام تشغيل ولغة البرمجة والادوات الاخرى من برامج ومكتبات لان فشلك فى تحقيق اهدافك من الحصول على برامج قوية عالية الاستقرار قد يعود الى البيئة وليس الى خطأ منك.

إذا واجهتك مشكلة اثناء تطوير نظامك فهناك احتمال بنسبة ٥% ان يكون الخطأ ليس من نظامك وانما من لغة البرمجة وهناك احتمال بنسبة ٢% ان يكون الخطا من برامج اخرى تعمل على النظام وهناك احتمال ٢% ان يكون الخطا من نظام التشغيل الذى تعمل عليه - وهذا بالتأكيد اذا كنت تطور انظمة معقدة تحتاج الى موارد كثيرة.

التعامل مع المشاكل الصعبة

عادة قد يصاب الانسان بشيء من خيبة الامل عندما يتعرض لمشكلة صعبة - ورد الفعل حينها يختلف من شخص لآخر فمنا من يهرب من المشكلة ويقول "وانا مالى - مش مهم" وكلمات اخرى لاينبغى ان تذكر فى الكتب - ومنا من ينفعل وكان

المسالة مسالة شرف وتراه مكبا على العمل وهو غاطب واحيانا يقول (طيب - ماشيء) وايضا كلمات اخرى لاينبغى ان تذكر فى الكتب - بصراحة النوع الاول بارد جدا وغالبا ما لايحقق النجاح لان الاهمال احد اسباب الفشل - والنوع الثانى غير متزن وقد ينجح فى حل المشكلة ولكنه غالبا عندما يصل الى الحل لايعرف كيف وصل ! - نعم يحدث ذلك بالفعل - وربما بعد حل المشكلة يقضى اوقاتا اخرى ليحلل ويفهم كيف وصل الى الحل - وياليت الامر يقف عند هذا الحد وخاصة ان الكثير يهتم بالحل ذاته وليس كيف - ولكن الامر يمتد ليشمل ان يقع الشخص فى مشاكل اخرى وهكذا فبعد ان كانت مشكلة اصبحت المسالة ٢ مشاكل - ثم ٥ ثم ١٠ ثم ١٠٠ وهكذا - وهذا الشخص المندفع بدلا من ان يتراجع عما فعل ليعود لمشكلته الاصلية - نجده يستمر فى خلق المشاكل وكأنه يقود سيارة (عربية) بدون فرامل وكل منا يقدر خطورة ذلك.

وهناك نوع متزن من البشر ويجدر بنا جميعا ان نكون من هذا النوع حيث عندما يواجه مشكلة فانه يسلك المنهج النفسى السليم والاسلوب المنطقى الراقى لحل المشكلة.

والسؤال الان : ما علاقة ذلك بالبرمجة ؟

يجدر بى ان اقول انى اعلم ان هذا السؤال هذه المرة لن يساله على الاطلاق اى مبرمج محترف لان المبرمجين ليس فقط يعرفون معنى المشاكل - وانما يقتنعون جيدا بان البرمجة اساسا كلها مشاكل - فى الحقيقة انك اثناء تعلم للبرمجة او لاي شى جديد فيها ستكون المشاكل Problems هى رسالة الاستقبال بدلا من مرحبا Welcome ولهذا تنتشر صفحات الاسئلة الشائعة واجاباتها FAQ داخل تعليمات لغات البرمجة وادوات التطوير وايضا البرامج.

لهذا تجد المبرمجين دائما يقولون ان البرمجة بحاجة الى (صبر) مستمر ولهذا قد تسمع عن ابتعاد كبار السن وخاصة بعد سن (٤٥) عاما عن البرمجة ويفضلون الاعمال الاخرى مثل الادارة لانه فى ذلك العمر بصراحة يكون الصبر قد نفذ - وكل من لديه والدين ويرى انهم ينفعلون بسرعة يقدر ذلك - ونحن دائما ننصح ابائنا بعدم الانفعال وهم ينصحوننا بعدم التسرع.

تختلف المشاكل فى النظم عنها فى التطبيقات - فغالبا مشاكل التطبيقات تنتج نتيجة نقص فى المعلومة - اما مشاكل النظم فتنتج نتيجة

✚ خطأ فى تطبيق التصميم

✚ خطأ فى التصميم

✚ خطأ فى بيئة العمل

والنوع الاول والثالث الى حد ما يحدث نادرا لان مبرمجين النظم غاية فى الاحتراف - كما ان نظم بيئة العمل غالبا ماتتمتع بالاستقرار عن اى نظم اخرى وبخصوص الخطا فى التصميم فهو الخطا الشائع وكما ذكرنا فى الجزء الاول من الكتاب تصميم النظم عملية فنية تمر بمراحل عديدة من الابحاث والتجارب والاختبارات.

الخلاصة

درب نفسك على التعامل مع مشاكل البرمجة باتزان نفسى وتذكر انك ان كنت مبرمج محترف فان تطوير التطبيقات يبعثك كثيرا عن المشاكل - لكن فى تطوير النظم الامر مختلف وتاكد ان المشاكل ستكون حياتك اليومية تستيقظ عليها وتسهر معها.

البحث عن الشهرة

ان هناك اناس على وجه الارض تعشق الجمال واناس تعيش من اجل المال واخرون ينكبون على العلم ومن الناس من يتمنى ان يحقق الشهرة وبالاخص من هم فى سن الشباب حيث يتوهمون ان تحقيق الشهرة نوع من انواع البطولة - مع ان الشهرة فى الواقع هى حساب فى البنك بدون رصيد - وان كان الناس يلتفون حول المشهورين - فان اغلبهم يفعلون ذلك من اجل مصالحهم والقليل جدا من الناس من استفاد من كونه مشهور.

اذا كنت عالما مشهورا ويعرف الناس كيف يصلون اليك فسوف تقضى عمرك لكى ترد عليهم وربما يقل مستواك كثير لانشغالك عن العلم باختلاط مع الناس الذين يسالون اسئلة ربما يعرف اجابتها من هم دونك من العلم بكثير

واذا كنت غنيا مشهورا ويعرف الناس كيف يصلون اليك فانتظر ان تستيقظ دائما على صياح المحتاجين اون كنت تخرج صدقتك وزكاتك بانتظام كما ينبغى فربما ترد الكثير منهم وتسيء الى صورتك فى المجتمع - وربما تعطيهم خوفا من ذلك ويضيع منك الاجر

واذا كانت فنان مشهور تجيد الرسم او كتابة الشعر او القصص او الغناء او التمثيل ويعرف الناس كيف يصلون اليك فانك سوف تعاني كثيرا - ممن يسالونك عن كيفية الوصول لما وصلت اليه - وممن يرون ان فنك غير شرعى ويطلبون منك التوقف - وغير ذلك

اما النجم المشهور فهو الشخص المشهور ولا يعرف الناس كيفية الوصول اليه لانه نجم بعيد - وفى نفس الوقت برون اعماله - اذا اردت الشهرة فكن كالنجم تسعد فى حياتك.

والسؤال الان : ما علاقة ذلك بالبرمجة ؟

ان البرمجة ليست الطريق المناسب لتحقيق الشهرة وان تمكن البعض من ذلك - والشواهد تقول انك اذا اردت الشهرة فعليك بان تكون فنان او لاعب كرة - اما البرمجة فالشهرة فيها نادرة وان تحققت فتكون فى مستوى ضيق ينصب على المهتمين بالمجال

الخلاصة

اذا اردت ان تكون مبرمج مشهور فعليك بطرح الاكواد المجانية المتميزة فى المشاريع مفتوحة المصدر - وعليك بكتابة المقالات والكتب- والمشاركة فى المنتديات المليئة بالاعضاء - وتذكر ان البرمجة ليست المجال المناسب للشهرة

ولعلها فرصة لذكر أسماء بعض المبرمجين المشهورين

- **Clifford Adams:** [strn](#), [UseModWiki](#)
- **Tim Berners-Lee:** [HTTP](#), [HTML](#), [www](#) (the first [Web browser](#))
- **Dan Bricklin:** [VisiCalc](#), [Demo](#)
- **John Carmack:** [Doom](#), [Quake](#)
- **Alan Cox:** Made significant improvements to [Linux](#)
- **Ward Cunningham:** [WikiWiki](#)
- **Dave Cutler:** [Windows NT](#), [VMS](#)
- **Miguel de Icaza:** [GNOME](#)
- **Bill Gates:** [Altair BASIC](#), [Apple II BASIC](#); founded [Microsoft](#).
- **James Gosling:** [Java](#), [Gosling Emacs](#), [NEWS](#)
- **Grace Hopper:** [Navy Mark I computer](#)
- **Bill Joy:** [BSD](#), [vi](#); founded [Sun Microsystems](#)
- **Mitch Kapor:** [Lotus 1-2-3](#); founded [Lotus Development Corporation](#)
- **Brian Kernighan:** [AWK](#) (co-author)
- **Donald E. Knuth:** [TeX](#), [CWeb](#), [The Art of Computer Programming](#)
- **John McCarthy:** [LISP](#)
- **Bertrand Meyer:** [Eiffel](#), [Object-oriented Software Construction](#), [Design by contract](#)
- **Eric S. Raymond:** [fetchmail](#), [The Cathedral and the Bazaar](#)
- **Dennis Ritchie:** [C](#), [Unix](#)
- **Ken Thomson:** [Unix](#), [C](#)
- **Guido van Rossum:** [Python](#)
- **Richard Stallman:** [Emacs](#), [GCC](#), [GNU](#)
- **Guy Steele:** [Common Lisp](#), [Scheme](#)
- **Linus Torvalds:** Originated the [Linux](#) kernel
- **Andrew Tridgell:** [Samba](#), [Rsync](#)
- **Larry Wall:** [rn](#), [patch](#), [Perl](#)
- **Niklaus Wirth:** [Pascal](#)
- **Steve Wozniak:** [Breakout](#), [Apple Integer BASIC](#); founded [Apple Computer](#) (with [Steve Jobs](#))
- **Jamie Zawinski:** [Lucid Emacs](#), [Netscape](#), [Mozilla](#), [XScreensaver](#)

المحاكاة

كما نعلم ان هناك فرق كبير بين المحاكاة والتقليد - حيث ان التقليد هو محاولة اعادة صنع الشئ نفسه مرة ثانية - وهناك اشياء تستطيع تقليدها واشياء تعجز عن تقليدها - اما المحاكاة فهى محاولة تمثيل وتصوير الشئ الذى يصعب او نعجز عن تمثيله - فمثلا نجد العديد من الالعاب التى تقوم بعمل محاكاة لعالم الطيران والسفر فى البحر وكرة القدم وغيرها العديد من الالعاب - وغالبا ماتكون المحاكاة مرئية ولكن الشئ الذى تحاكيه قد يكون مرىء او غير مرىء - بالنسبة للعين المجردة.

والسؤال الان ما علاقة ذلك بالبرمجة ؟
هل تقصد المعنى المباشر - برمجة الالعاب - انظمة الذكاء الاصطناعى ؟

بالتاكيد لا اقصد المعنى المباشر - وانما اقصد المضمون حيث ان المحاكاة كانت وسيلة لتحقيق الاهداف فمثلا محاكاة النظم العسكرية والجيش فى لعبة Red Alert او Age of empire كانت تهدف الى المتعة اولا كلعبة - وزيادة فهم اللاعب للعسكرية.

ويمكنك كمبرمج ان تستغل المحاكاة كوسيلة لتحقيق اهداف - بلى انى اعتدت ان ابحث عن حلول لمشاكل الصعبة فى البرمجة عن طريق المحاكاة.

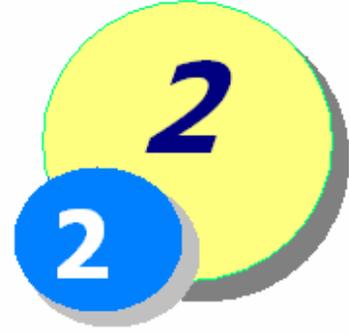
مثلا سطح المكتب Desktop فى اغلب نظم التشغيل التى تدعم الواجهة الرسومية للمستخدم GUI هو محاكاة للمكتب فى العالم الحقيقى حيث تمثل النواقد Windows الاوراق التى يمكن تحريكها وتبديلها واحدة مع الاخرى.

ولكن هنا استخدمت المحاكاة لصالح المستخدم User
اما المحاكاة لصالح المبرمج فتكمن فى واقع اسلوب برمجة النظام - ان المحاكاة فن راقى من فنون البرمجة فمثلا البرمجة الهيكلية تحاكي المباني المعمارية - الوراثة فى برمجة الكائنات تحاكي الوراثة فى الكائنات الحية - ادارة تطبيقات الزبون الخادم - تحاكي التفاعل الانسانى - الالعاب ونظم ادارة الاحداث هى عبارة عن محاكاة لردود الفعل نتيجة تغير الاحداث

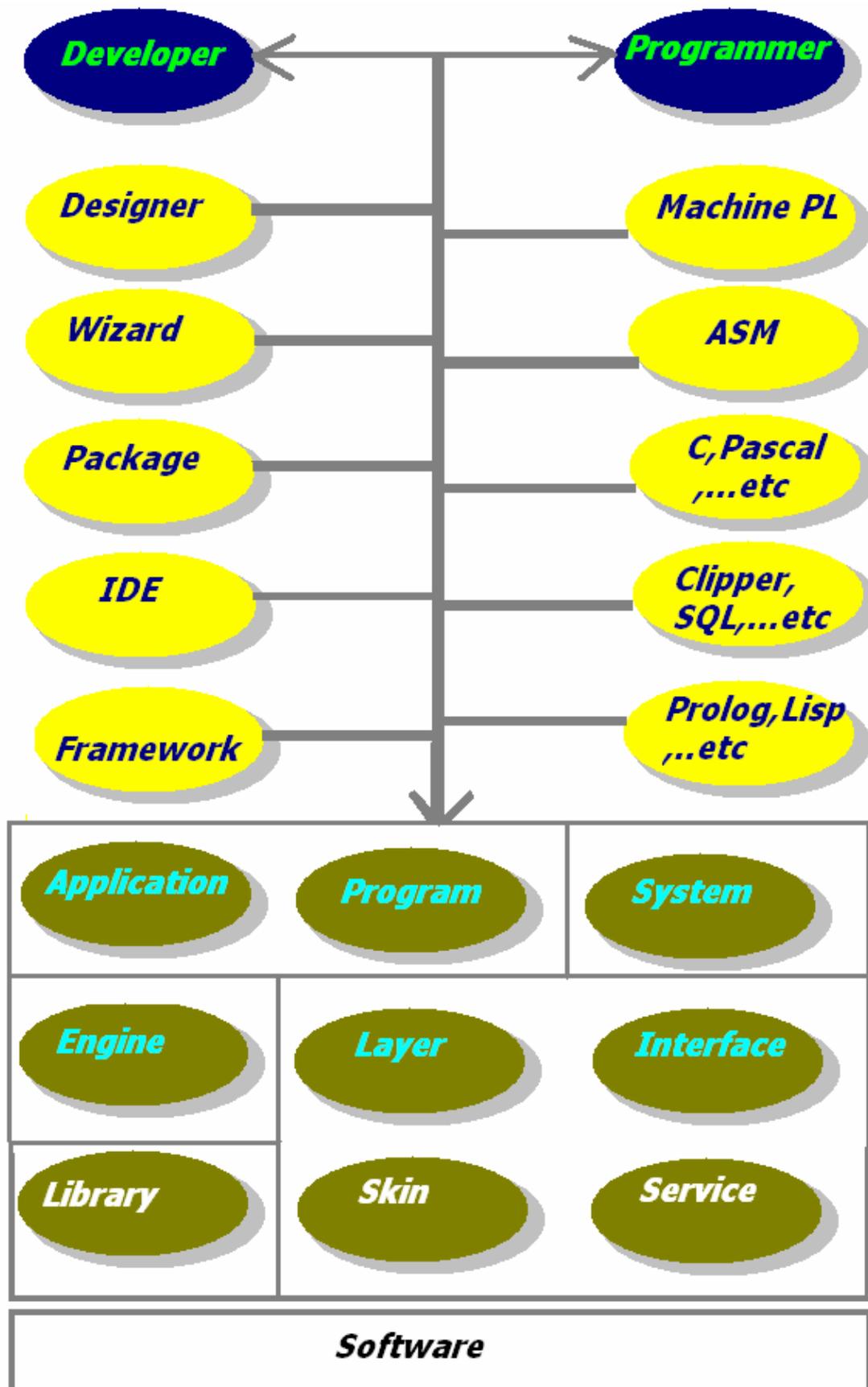
- ويمكن تصورها بالدوائر الكهربائية - وهيكل البيانات تحاكي
النظم الكيميائية حيث يمكن تصور البيانات كالكثرونات داخل
مدارات بدلا من المصفوفات ووجود ذرات تشملها وهكذا.
ومحاكاة العدل كما فى تقسيم موارد الحاسب بين البرامج من
قبل نظام التشغيل - ومحاكاة تقسيم المهام لادائها بكفاءة كما
فى النظم الموزعة Distributed Systems ومحاكاة تنفيذ مهمة
على التوازي كما فى Grid Computing ومحاكاة الرياضيات كما
فى الدوال Functions ومحاكاة اللغات الطبيعية كما فى الذكاء
الاصطناعى

المحاكاة غالبا ماتكون حل لكثير من المشاكل فى النظم - وقد
تساعد على ايجاد الحل - لذلك عليك ان تتامل فى الكون جيدا
فربما تجد فيه حلا لمشاكلك.

تكنولوجيا معاصرة



الشكل التالي يبين احدث ماوصلت اليه تكنولوجيا تصنيع البرمجيات حيث نجد ان البرمجة تطورت على عدة مراحل فمنذ قديم الازل كانت برمجة العتاد من خلال لغة الالة Machine Language ثم لغة التجميع - الاسمبلى ثم لغات الجيل الثالث وهى عديدة مثل لغة سى وباسكال وكوبول وفورتران وغيرها - ثم لغات الجيل الرابع الخاصة بانظمة قواعد البيانات مثل كليب واوراكل وغيرها فهم كثر - ثم لغات الجيل الخامس الخاصة باللغات الطبيعية وانظمة الذكاء الاصطناعى على الصعيد الاخر كان الاتجاه نحو تسهيل عملية انتاج البرمجيات من خلال ادوات التطوير التى يستخدمها المطورون Developers لانتاج برمجيات جيدة بسهولة ويسر وبسرعة عالية ومر ذلك بمرحل مختلفة مثل ادوات التصميم designers والمعالجات Wizards التى تعتمد على عدة قوالب Templates بالاضافة الى خيارات المستخدم - وايضا البرمجيات الخاصة Package مثل برنامج PowerPoint الشهير وبرنامج فلاش الغنى عن التعريف وغيرها الكثير من البرمجيات الخاصة - وايضا ظهرت بيئات التطوير المتكاملة IDE التى تشمل كم هائل من المصمات القوية والمعالجات وتسهل عملية ادارة المشاريع وغيرها من الامكانيات واخيرا محيط التطوير Framework الذى يتيح انشاء تطبيقات بسمات خاصة ويشمل ادوات تطوير تعمل بصورة متشابهة وعلى اسس مشتركة ثابتة وهو قابل للامتداد بسهولة بحيث تحمل الادوات الجديدة سمات مشتركة بينها وبين الادوات التى سبق التعرف عليها مما يسهل عملية التعلم والاستخدام فهو عبارة عن (مطبخ) بمجرد التعرف على تركيبه ومحتوياته يسهل اضافة محتويات جديدة اليه.



والجدير بالذكر ان ادوات البرمجة ليست قاصرة فقط على المبرمجين فقد يلجا اليها المطورون للتعامل مع المشاكل العصبية - كما ان ادوات التطوير ليست قاصرة على المطورين فقد يلجا اليها المبرمجين لتسهيل عملهم واختصار الوقت والحصول على نتائج جيدة بسرعة.

وكما نلاحظ فى الشكل فان ماينتج من عمليات التطوير او البرمجة او كلاهما معا هو برمجيات Software ولاتى تصنف الى

- تطبيقات Applications
- برامج Programs
- نظم Systems
- محرك Engine
- طبقة Layer
- رابط Interface
- مكتبة Library
- غلاف Skin
- خدمة Service

اولا : التطبيقات Applications

هى نتيجة استخدام ادوات التطوير

○ التطبيقات البسيطة تعتمد على Package

- البساطة من حيث سهولة التطوير
- لايعنى ذلك انها ضعيفة فقد يعجز المبرمج او

المطور العادى على برمجتها بدون Package

○ التطبيقات القوية تعتمد على لغات تشمل ادوات التطوير الاخرى

ثانيا : البرامج Programs

هى نتيجة استخدام ادوات البرمجة

○ البرامج منخفضة المستوى الى تتعامل مع العتاد تعتمد على لغات الجيل الثانى او لغات من الجيل الثالث تحمل امكانيات الجيل الثانى مثل لغة سى وباسكال

○ البرامج عالية المستوى التى تخدم اغراض السوق بصورة مباشرة مثل برامج انظمة قواعد البيانات يتم برمجتها بلغات الجيل الرابع

○ برمجيات انظمة الذكاء الاصطناعى تطور بلغات الجيل الخامس

ثالثا : النظم Systems

➤ هى البرامج Programs المعقدة كبيرة الحجم صعبة البرمجة

- سبق التحدث عنها فى الجزء الاول من الكتاب
- غالبا مالاتعمد على ادوات التطوير المتاحة - حيث يتم بنائها من الصفر حفظا على مستقبل النظام وعدم توقف تطويره او وجود مشاكل بسبب جهات اخرى
- اذا اعتمد النظام على ادوات اخرى فانها غالبا ادوات شهيرة شائعة الاستخدام - يثق بها الجميع
- يمكن للافراد انتاج البرامج والتطبيقات - بينما تحتكر الشركات برمجة النظم لانها مهمة صعبة وبحاجة الى فريق عمل متميز وجاد

رابعا : المحرك Engine

➤ هو نظام ذو سمات خاصة

- يودى مهمة محددة قد لا تكون هى النتيجة النهائية
- يعمل مستقلا او داخل نظام اخر كنظام جزئى Sub System او نظام مختبى Embedded System
- عندما يدخل فى نظام اخر فانه يسمى نظام جزئى Sub System اذا كان يمكن التعامل معه من الانظم التى تتعامل مع النظام - ويطلق عليه نظام مختبى Embedded System اذا كان لا يمكن التعامل مع الى من النظام الذى يحتويه

خامسا : الطبقة Layer

➤ هى جزء من النظام له مهمة محددة

- تختلف عن engine فى انها
 - لايمكن فصلها من النظام
 - تحتوى معنى متضامن ولكن قد يكون غير مكتمل بمعنى انها من الممكن ان تعتمد Layer على Layers اخرى داخل النظام

سادسا : الرابط Interface

- هو وسيط بين طرفين او مجموعة من الاطراف
- الاطراف قد تكون عبارة عن طبقات Layers
- الاطراف قد تكون عبارة عن محركات Engines
- الاطراف قد تكون عبارة عن نظم Systems
- الاطراف قد تكون عبارة عن برامج Programs
- الاطراف قد تكون عبارة عن تطبيقات Applications
- احد الاطراف قد يكون عبارة عن Resources اى موارد وهذه الموارد تشمل
 - العتاد Hardware
 - البرمجيات Software بانواعها

سابعا : المكتبة Library

- مجموعة من التعليمات Instructions المصنفة الى دوال Functions او فئات Classes حسب نمط البرمجة - يمكن المشاركة فى استخدامها من قبل البرمجيات التى يتم تطويرها
- المكتبات نوعين
 - نوع يستخدم عن الربط Linking اى عند استخراج ملفات تنفيذية
 - نوع يستخدم اثناء وقت التشغيل Runtime وهو يصنف الى نوعين
 - نوع ثابت Static لبرنامج محدد
 - ونوع ديناميكى Dynamic لاكثر من برنامج

ثامنا : الغلاف Skin

- جزء من النظام او البرنامج او التطبيق يكون وسيط او رابط للتعامل مع النظام او البرنامج او التطبيق
- حيث ان الرابط Interface يسمع باستخدام موارد من العتاد او البرمجيات - فان الغلاف Skin يسمح بكون النظام او التطبيق او البرنامج هو احد الموارد بالنسبة لنظام او برنامج او تطبيق اخر.

تاسعا : الخدمة Service

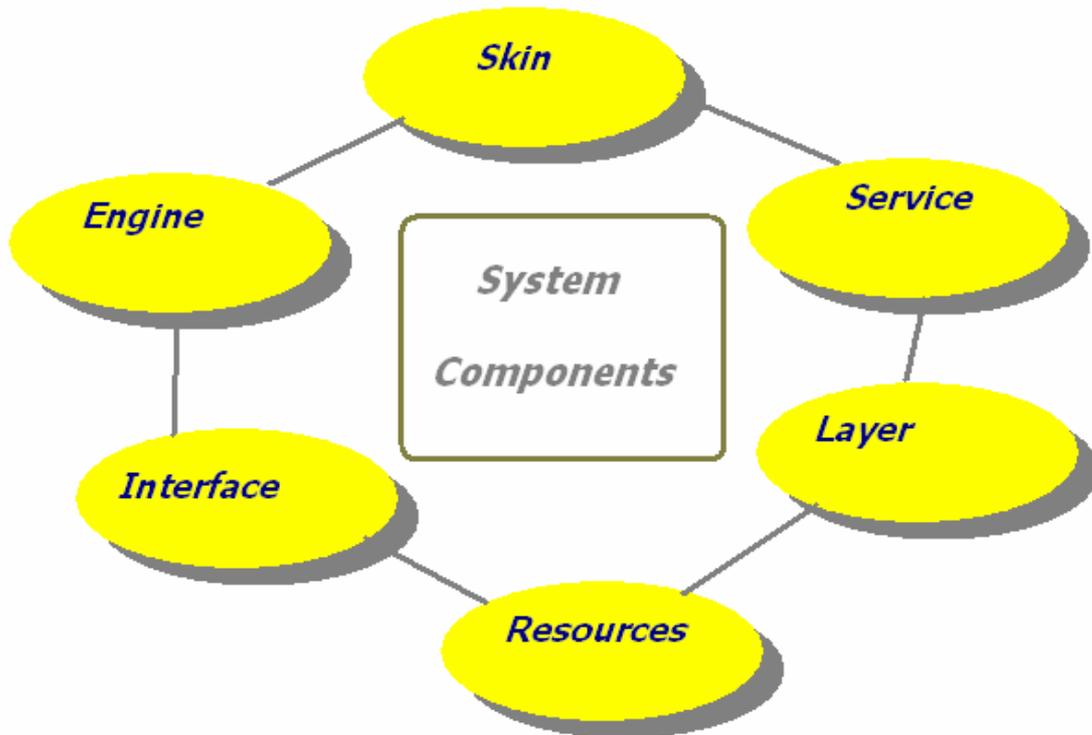
ما يمكن ان يقدمه النظام للعالم الخارجى بالنسبة له - اى البيئة المحيطة

o الفرق بين الخدمة Service والغلاف Skin هى ان خدمة عبارة عن وظيفة Job يقوم بها النظام لطرف اخر - اما الغلاف Skin فهو وسيلة للوصول لتلك الخدمة - ولا تقتصر مهمة الغلاف Skin على ذلك بل تمتد لتشمل امكانية دمج النظام مع طرف اخر وهكذا

والان بعد التعرف على تصنيف البرمجيات Software يتسنى لنا التعرف على مكونات النظم

مكونات النظم System Components

مكونات النظم System Components



كما نرى شمل النظام برمجيات تصنف الى ٦ عناصر اساسية سبق التعرض لها - ولم يذكر البرامج Programs او التطبيقات Applications لان النظام فى حد ذاته بديل لها - واذا تم استخدام برامج او تطبيقات خارجية مع النظام فانها تدخل ضمن

الموارد Resources كما ان المكتبات Libraries لم تذكر ايضا لانه اذا تم بنائها من الصفر تدخل ضمن المحركات Engines او الطبقات Layers واذا تم استخدام مكتبات لم يتم بنائها من الصفر فانها تندرج ضمن الموارد Resources.

- والان نتعرض لمراحل بناء النظام
- ١ - مرحلة الابحاث العلمية Researches
 - ٢ - مرحلة التحليل Analysis
 - ٣ - مرحلة التصميم Design
 - ٤ - مرحلة البرمجة Programming
 - ٥ - مرحلة الاختبار وتصحيح الاخطاء Testing & fixing Bugs
 - ٦ - مرحلة التوثيق Documentation

بخصوص مرحلة الابحاث العلمية فقد تم الاشارة فى الجزء الاول من الكتاب انه سوف يتم التعرض لها فى الجزء السادس من الكتاب باذن الله

اما مرحلة التحليل فهى عملية تخضع لعوامل كثيرة مثل الحاجة من النظام وظروف بيئة العمل والادوات المتوفرة ومدى كفاءة المبرمجين ومقدار ماتوصلت اليه الابحاث العلمية من نتائج مفيدة جاهزة للتطبيق المباشر

وبخصوص مرحلتى الاختبار وتصحيح الاخطاء ومرحلة التوثيق فلا تختلف عن تطوير البرامج والتطبيقات بصفة عامة من حيث المفهوم ولكنها اصبحت اكثر تعقيدا واصبحت عملية فنية ليس فقط فى الاختبار وتصحيح الاخطاء بل ايضا فى التوثيق لانه من الصعب كتابة توثيق مفهوم وشامل ومرن للنظم المعقدة

- اما مرحلة التصميم فيتم تحديدها بالخطوات التالية
- ١ - فهم التحليل جيدا لمعرفة المطلوب بعناية
 - ٢ - تحديد الموارد Resources التى سوف يتم استخدامها
 - ٣ - تصميم الرابط Interface للتعامل مع تلك الموارد
 - ٤ - تحديد المحركات Engines التى سوف يشملها النظام
 - ٥ - تحديد وتصميم طبقات النظام Layers
 - ٦ - تحديد وتصميم الخدمات Services التى سوف يقدمها النظام
 - ٧ - تصميم الغلاف Skin

ام بخصوص مرحلة البرمجة فهي ان نقوم تنفيذ ما يطلبه التحليل ملتزمين بالتصميم حيث ان التحليل هو المطلوب والتصميم هو خريطة العمل وقيود العمل فى نفس الوقت

وبالتاكيد البرمجة تعنى استخدام التكنولوجيا الخاصة ببيئة العمل من لغات البرمجة (يمكن استخدام اكثر من لغة برمجة فى تطوير النظام) ومن نظم التشغيل (يمكن دعم اكثر من نظام اذا كام النظام الذى نظوره ليس نظام تشغيل)

ومن المعلوم انه من الشركات الرائدة فى مجال توفير تكنولوجيا ادوات التطوير و البرمجة على سبيل المثال لا الحصر

- ✚ Microsoft
- ✚ Macintosh
- ✚ Sun
- ✚ Borland
- ✚ Oracle
- ✚ Computer Associates
- ✚ Sybase
- ✚ Alaska

وتتحفنا هذه الشركات دائما بتكنولوجيا جديدة - تاكد دائما ان لاتستطيع الالمام بها كاملة حتى لو تخصصت فى لغة برمجة واحدة لان المسالة لم تعد تقف عند نقطة تعلم لغة برمجة بل امتدت لتشمل ادوات تطوير ومكتبات قام بصنعها الالاف من المبرمجين لتخدم مختلف مجالات التطوير ماسمعت عنها ومالم تسمع ولعلك تعترض على استخدام لفظ "الالاف من المبرمجين" وخاصة ان نظام التشغيل "Microsoft Windows" قام بالعمل به ٣٠٠٠ الاف مبرمج - فاجيب عليك عزيزى القارى بان تطوير لغة البرمجة لايقف عند الشركة المصنعة للغة - بل يمتد الى شركات متعاونة تقدم ادوات ومكتبات - كما يمتد الامر للمبرمجين الذين يستخدمون اللغة ويوميا تفاجىء بادوات جديدة قد غزت العام وقد تكون هذه الادوات مجانية Freeware او مفتوحة المصدر Open Source او يتم بيعها Shareware وقد تحصل على نسخة مجانية محدودة Limited Version من منتجات تجارية.

فانها تقدم

بخصوص شركة مايكروسوفت

Developer Tools

[Visual Studio](#)

[Visual Basic](#)

[Visual C#](#)

[Visual C++](#)

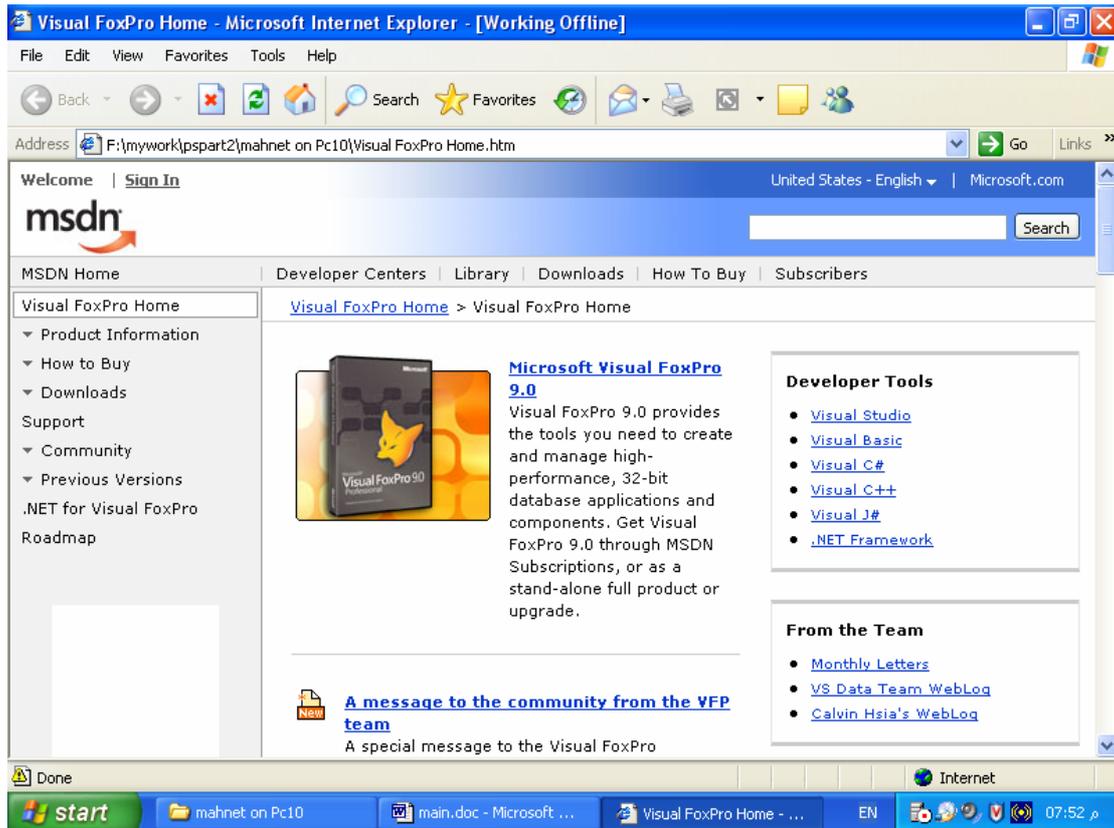
[Visual J#](#)

[.NET Framework](#)

ونلاحظ عدم ظهور Visual FoxPro لان مايكروسوفت كانت تطورها لغرض المنافسة مع شركة CA وشركة Alaska وذلك فى زمن Dos ومع بدايات Windows ولكن فى عصر Windows تالقت لغات اخرى وبدايات لغات xBase فى التراجع بعد ان كانت الاختيار الاول لتطوير التطبيقات التجارية - اصبحت الان لغات نادرة ولا تذكر الا نادرا - ولكن ذلك لا ينفى تميز هذه الفئة من اللغات باعتبارها مدير قاعدة بيانات DBMS ولغة برمجة Programming Language فى منتج واحد

والجدير بالذكر ان كلمة xBase تعود للغة مقابلة للغة SQL اى لغة بديلة لها للتعامل مع قواعد البيانات العلاقية - ولكن الفرق بينهما يكمن فى ان SQL ذات عدد قليل من الاوامر التى تتعامل فى الامر الواحد مع جدول او عدة جداول محددة - على العكس مع لغة xBase التى لها اوامر عديدة وتتعامل مع جدول او عدة جداول بدون اشتراط معرفة الجدول الذى تعمل معه - لهذا فهى ملائمة جدا فى تطبيقات Desktop Applications التى تعمل على جهاز واحد - وايضا هى جيدة فى التطبيقات التى تعمل على شبكة

Network Client Sharing يتم عليها المشاركة فى ملفات قاعدة البيانات اما SQL فتجد مكانها المناسب فى تطبيقات Server ونظرا لان لكل منهما مميزات وعيوبه فان لغة مثل فيجوال فوكس برو Visual FoxPro التى تعتبر احد لغات xBase تدعم SQL ايضا بنفس الدرجة كاحد اوامر اللغة بدون استخدام ادوات مثل ADO وذلك لاينفى دعمها لتلك الادوات فهى لغة متطورة.



والجدير بالذكر انه لن يكون هناك اصدار جديد من اللغة ولكن دعمها مستمر حتى عام ٢٠١٥ وسوف تجد فى موقع شركة مايكروسوفت كتاب من ٥٠٠ صفحة مجانى للانتقال من فيجوال فوكس برو الى الدوت نت.

وبالتاكيد ان اللغة الاولى فى الدوت نت هى الفيجوال سى شارب والتى تختصر بالسى شارب يليها بالطبع فيجوال بيسك دوت نت.

وكما يعلم الجميع فان السى شارب متشابهة الى حد كبير مع لغة الجافا Java ولكن ليس تماما

والجدير بالذكر ان الدوت نت يشمل فيجوال سى ++ والتى تتيح عمل تطبيقات مستقلة لاتحتاج للدوت نت لكى تعمل.

Visual C# Developer Center - Microsoft Internet Explorer - [Working Offline]

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites

Address <F:\mywork\pspart2\mahnet on Pc10\Visual C# Developer Center.htm> Go Links

Welcome | [Sign In](#)



MSDN Home | [Developer Centers](#) | [Library](#) | [Downloads](#) | [How To Buy](#) | [Subscribers](#)

[Visual C# Home](#)

- Learn C#
- Reference
- Downloads
- Support
- Community
- Webcasts
- Virtual Labs
 - Product Information
 - Previous Versions
 - Future Versions

[Visual C# Home](#) > Visual C# Developer Center

Visual C# Developer Center

Visual C# is a simple, type-safe, object-oriented, general-purpose programming language. It provides code-focused developers with powerful tools and language support to build rich, connected Web and applications on the .NET Framework.

Now Available: The Visual Studio Orcas Beta 1 Download.
The March CTP is now available for download as a regular install and as a virtual PC. A video that serves as an introduction to the CTP is also available.

Custom Iterators
Learn from top selling author Bill Wagner how to create building blocks for your program using custom iterators and the yield statement.

Done, but with errors on page. Internet

start mahnet on Pc10 main.doc - Microsoft ... Visual C# Developer ... EN 07:56

Visual C++ Developer Center - Microsoft Internet Explorer - [Working Offline]

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites

Address <F:\mywork\pspart2\mahnet on Pc10\Visual C++ Developer Center.htm> Go Links

Welcome | [Sign In](#) United States



MSDN Home | [Developer Centers](#) | [Library](#) | [Downloads](#) | [How To Buy](#) | [Subscribers](#)

[Visual C++ Home](#)

- Learn Visual C++ 2005
- Reference
- Downloads
- Support
- Community
- Product Information
- Previous Versions

[Visual C++ Home](#) > Visual C++ Developer Center

Visual C++ Developer Center

Visual C++ provides a powerful and flexible development environment for creating Microsoft Windows-based and Microsoft .NET-based applications. It also enables developers to build Web applications, smart-client Windows-based applications, and solutions for thin-client and smart-client mobile devices.

Desktop Security: Create Custom Login Experiences with Credential Providers for Windows Vista
Credential Providers in Windows Vista allow you to customize the logon experience and integrate the authentication methods that best meet your organization's needs. With this article, learn how to write your own.

Misplaced your list of C++ compiler options?
Bookmark this page for quick access to the Visual C++ compiler options. They are listed both alphabetically and by category.

Done Internet

start mahnet on Pc10 main.doc - Microsoft ... Visual C++ Develop... EN 08:01

وبخصوص مبرمجين فيجوال بيسك ٦ الذين انتقلوا الى الدوت نت فهم دائما فى حيرة من امرهم التشابه بين الاصدارين تقريبا لايتخطى كونه تشابه فى الاسماء ومع تفوق الدوت نت فى الامكانيات الهائلة التى يقدمها الى ان فيجوال بيسك سهلة ومرنة وتجد مكانها فى قلب كل من استخدمها - ومن منا لايعشق فيجوال بيسك ٦ - وخاصة ان معظم المبرمجين تعلموا البرمجة لويندوز من خلال فيجوال بيسك

وقد انقسم المبرمجين الى عدة فرق

١ - فريق ضائع لا محالة قرر اساسا عدم التطور وتعلم الدوت نت واكتفى فيجوال بيسك ٦ - نقول لهذا الفريق تقدم للامام وتعلم الدوت نت لانه اذا كان به صعوبة الان - فانه سيصبح اسهل من الذى هو عليه مع كل اصدار جديد - ولا تنسى ان تطوير تطبيقات الانترنت والشبكات والموبايل هو جائزتك الكبرى.

٢ - فريق قرر استخدام فيجوال بيسك ٦ لتطوير التطبيقات القديمة التى سبق وان قامو بعملها واستخدام الدوت نت فى المشاريع الجديدة بصفة مطلقة

٣ - فريق قرر إعادة كتابة مشاريع فيجوال بيسك ٦ مرة ثانية بالدوت نت مع الاستعانة بتقنية COM القديمة لتسهيل عملية النقل - واستخدام تقنية الـ COM/DCOM/COM++ لايفضل مع الدوت نت لانه يعنى Unmanaged Code - وبالطبع من درس الدوت نت يعلم الفرق بين Managed code و Unmanaged Code وكذلك الفرق بين Safe code و Unsafe Code

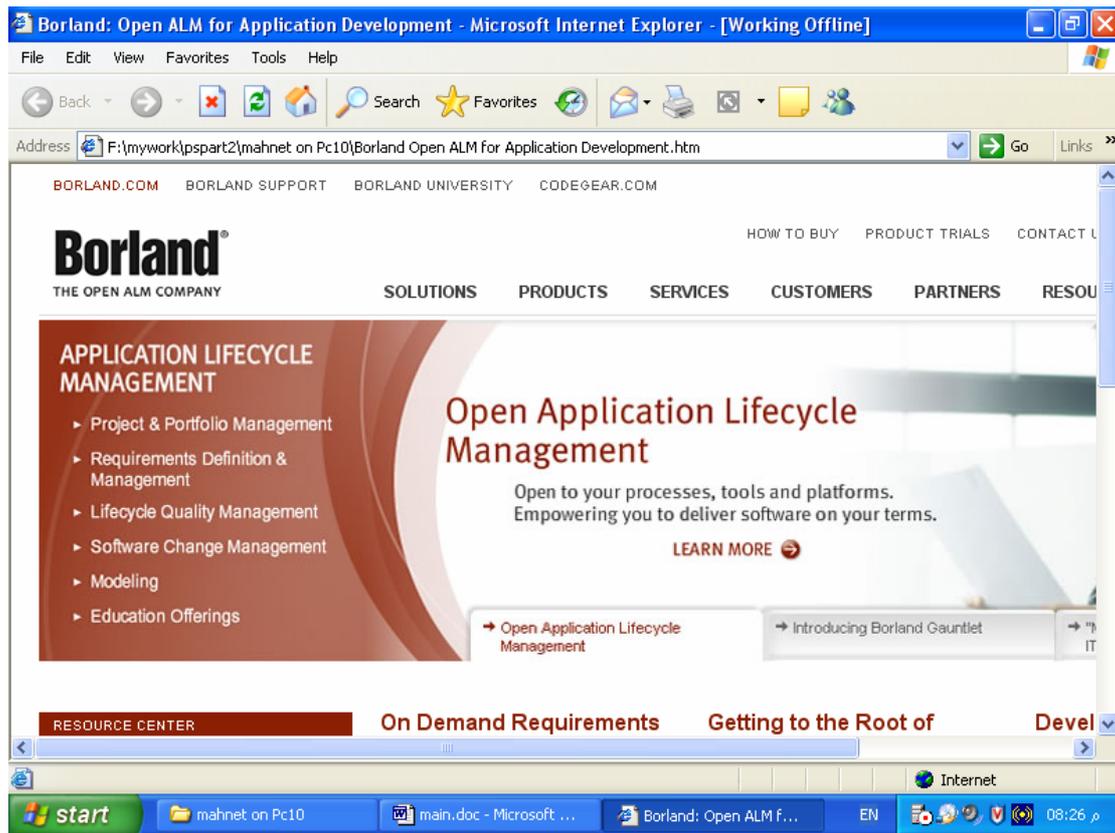
وبالتاكيد هذا الفريق سوف يطور المشاريع الجديدة باستخدام الدوت نت

٤ - فريق قرر عمل التطبيقات الجديدة من النوع Desktop باستخدام Visual Basic 6 وغالبا مع Access وعمل تطبيقات الشبكات متوسطة الحجم باستخدام الدوت نت مع SQL Server والكبيرة الحجم مع Oracle

اما مبرمجين لغة Visual C++ فانهم محظوظين لتوقر اللغة فى الدوت نت بدون صدام - ومع توفر سى شارب كاختيار لابد منه عند الرغبة فى التعمق الكامل فى العالم الجديد.

الما بخصوص شركة بورلاند

تقدم تكنولوجيا مستقرة تماما - او لا تقدمها من البداية ومن منا كان له باع في عالم البرمجة تحت DOS وبالاخص باستخدام لغة سي فانه يعلم المنافسة الشرسية التي كانت بين مايكروسوفت وبورلاند في اصدار تلك اللغة وفي الواقع كانت بورلاند صاحبة الغنيمة الكبرى حيث ان معظم تطبيقات دوس المطورة بلغة سي كانت باستخدام Borland C ولم يقتصر الامر على ذلك فقد كانت الشركة تقدم اللغة الرائعة Pascal والتي كان يعشقها اساتذة الجامعات (اظن هناك جامعات متاخرة مازالت تعشقها حتى الان) وكنا نستمتع بـ Turbo C و Turbo Pascal ولاحظت مايكروسوفت ذلك فانتجت Quick C و Quick Basic والتي عادة ماتكون اول لغة نتعلم بها البرمجة في عصر DOS نظرا لسهولةها.



وتعمل الان بورلاند في مشروع يدعى CodeGear وكلمة Gear تعني "ترس"

Developer Tools Group Becomes



Borland has formed CodeGear - a separate, new operation focused exclusively on maximizing developer productivity.

اي ادوات التطوير تجمعت تحت مسمى CODEGEAR والذي يهدف الى زيادة إنتاجية المطورين (هذا كلام عام لانه اساسا هدف ادوات التطوير دائما هو زيادة الانتاجية - ولاينفى ذلك حقيقة الامر لان CODEGEAR بالفعل يزيد الانتاجية ومن يستخدمه سوف يلاحظ ذلك).

ويشتمل الـ CODEGEAR على المنتجات التالية

CodeGear Products

C++Builder®

The ultimate C++ visual development environment for the Windows® platform

Delphi® 2007 for Win32®

The RAD visual development environment for Windows®

Delphi® for PHP

The RAD visual PHP development environment

JBuilder®

Making collaborative development fast and reliable for Java™, open source and the web.

C#Builder®

A complete solution for enterprise C# development

Delphi 2006®

The premier rapid application development solution for Windows

InterBase®

A high-performance, cross-platform, Unicode, SQL 92 compliant database

Turbo™

Everything you need to start real development today, the fun and easy way!

ولعلك تسال الان - ما الفرق بين Delphi 2006 و Delphi 2007 ؟ for win32

ان Delphi 2007 for WIN32 يتيح عمل تطبيقات WIN32 تعمل على **Windows 2000, XP, AND Vista** لا تحتاج الى الدوت نت ويوجد منها اصدارين

Delphi® 2007 for Win32® Professional Edition

للافراد والمؤسسات الصغيرة

للمؤسسات الكبيرة
اما بخصوص Interbase

InterBase®

Ensure durability and reliability leveraging the powerful database for embedded and enterprise applications



High performance durable database solution
Scalable and portable; designed for flexibility as demands grows
Powerful, high availability database for embedded and enterprise applications

فهي عبارة عن قاعدة بيانات قوية قابلة للدفن Embeddedable
ويوجد منها ثلاث اصدارت

- ✚ Server Edition
- ✚ Desktop Edition
- ✚ InterBase Developer Edition

اما بخصوص Delphi for PHP

Delphi® for PHP

The RAD visual PHP development environment

Accelerate PHP web development with Delphi's proven visual (RAD) framework for PHP
Simplify PHP programming with Delphi's comprehensive PHP for VCL libraries
Sophisticated editing and integrated debugging speeds up the process of tracking down bugs and errors in the code
Integrated development environment streamlines code management and navigation so managing and organizing PHP code has never been easier.

فهي عبارة عن توفير بيئة تطوير متكاملة تدعم اللغة PHP الشهيرة في تطوير صفحات الانترنت الديناميكية - وهي لغة غالبا على قلوبنا وعلى قلب كل من يحترف تطوير صفحات الانترنت بالادوات المجانية.

ولعل ذلك يلفت الانتباه الى ان الشركات العملاقة بدأت تلتف حوال المشاريع المجانية مفتوحة المصدر التي اخذت مكانها فى السوق وحققت النجاح.

ولعلك الان تتامل المنتج Turbo وخاصة ان له جذور قديمة جدا

Turbo™

The fast and easy way to start Windows development!

High productivity single language development environments
Develop powerful apps for the web and the desktop
Quickly and easily access all your data
Source code, games, trivia and more at turboexplorer.com

فى الحقيقة ان حل شركة بورلاند لمن يرغب فى الحصول على اداة تطوير سهلة الاستخدام وسريعة – وبالتاكيد تطوير تطبيقات وندوز – وهى تدعم تطبيقات الويب وسطح المكتب Desktop

وهو يشمل

-  Turbo Delphi
-  Turbo Delphi for .NET
-  Turbo C++
-  Turbo C#

ويتميز بالتاكيد بانه يمكن ضبضه بالصورة الملائمة لنا كما انه قابل للامتداد – ويشتمل على اكثر من ٢٠٠ عنصر جاهزين للاستخدام مباشرة Drag & Drop لتسهيل برمجة وندوز

اما بخصوص JBuilder 2007

JBuilder® 2007

Making collaborative development fast and reliable for Java™, open source and the web



Easily move existing JBuilder projects to the new Eclipse-based JBuilder 2007 IDE
Accelerate installation and configuration of a complete developer tool stack
Blend Open Source and commercial software, plug-ins, tools and frameworks
Boost collaborative development with integrated project portal tools

فهو هدية شركة بورلاند لمبرمجين لغة جافا حتى تزيد سرعة تطوير التطبيقات.

اما بخصوص C++ Builder

C++Builder® 2006

Updates and refines the popular C++Builder IDE with the latest RAD features for mission-critical C++ and C development



Simplify coding with new IDE productivity features and refactorings
Leverage a wide variety of 3rd party components with the latest VCL
Accelerate application delivery through seamless IDE integration with version control, bug tracking, and team collaboration

فهو منتج غنى عن التعريف - وهذا الاصدار يسهل عملية كتابة الاكواد ويحتوى ادوات تطوير إضافية كما يدعم الفريق (مجموعة من المبرمجين يعملون معا) بصورة افضل

ويوجد منه اصدارين

- 🚀 C++ Builder Professional
- 🚀 C++ Builder Enterprise

اما بخصوص C# Builder 2006

C#Builder® 2006

- Bring Enterprise Development Capabilities to .NET
- O/R Mapping, transparent object persistence, and transactions
- UML modeling, two-way code-model synch, audits and metrics
- Integrated requirements management, version control, bug tracking, and team collaboration

فانه يدعم التطبيقات الموسعة Enterprise و لغة التوثيق الموحدة UML ودعم فرق العمل وسهولة ادارة المشاريع

ويوجد منه ٣ إصدارات

- C# Builder Professional
- C# Builder Enterprise
- C# Builder Architect

ولعلنا نلاحظ ان تكنولوجيا شركة بورلاند بدأت تلتف حول المنتجات الاخرى مثل Java و PHP و .NET. وهذا امر يشكل خطورة على مستقبل بورلاند ومنتجاتها ويدل على تميز تلك المنتجات الاخرى

اما بخصوص شركة Sun

لقد امتعتنا تلك الشركة بلغة البرمجة Java والتي الان تعتبر اللغة الاولى فى العالم من حيث عدد المستخدمين والمهتمين بتلك اللغة التى تجد لها مكان فى كل انظمة التشغيل المشهورة والمتطورة.

ويوجد منها ثلاث إصدارات

- Java SE (Standard Edition)
- Java EE (Enterprise Edition)

Java ME (Mobile Edition)

ولعل الاصدار الخاص باجهزة الموبايل علامة بارزة فى منتجات شركة سن يحق لها بالفعل ان تفتخر به

ملحوظة :

تعد بيئة التطوير المتكاملة [NetBeans IDE](#) اداة جيدة يمكن استخدامها لتطوير تطبيقات Java ولكن العيب الذى لمستته لهذه الاداة هو البطء الشديد اثناء تحميل المكونات التى تعتمد عليها الاداة وذلك يحدث فى كل مرة يتم تشغيلها فيها - ولايقف الامر عند ذلك بل يستمر البطء اثناء عملها - وهذا يزعجنى كثيرا - وخاصة انها ابطا من الدوت نت - علاوة علا ان بطء تطبيقات Java بصفة عامة بطء لا تحسد عليه - ولهم العذر فى ذلك نظرا لاعتماد اللغة على Virtual Mahchine تقوم احيانا بمهام يفترض ان يقوم بها نظام التشغيل مثل متابعة العمليات Processes.

أما بخصوص شركة CA

لا اعرف وضعها الحالى بالتحديد ولكنها كانت فى يوم من الايام ثانى اكبر بائع برمجيات فى العالم - حيث انها عبارة عن اكثر من ٥٠٠ شركة على مستوى العالم اتحدت معا لتكون شركة عالمية واحدة

كان لهذه الشركة منتج قديم يدعى Clipper (قبل ان تشتريه من شركة نانتوكيت الامريكية) ثم تحول الى CA-Clipper عندى اشتريته الشركة - هذا المنتج الساجر نشا منذ عام ١٩٨٤ حتى عام ١٩٩٧ وهو عبارة عن لغة برمجة من الجيل الرابع بدات كلغة برمجة متخصصة فى انظمة قواعد البيانات ثم تحولت الى لغة عامة - ورغم ذلك التميز وخصوصا انها فى عصرها كانت لغة البرمجة الاولى لنظام التشغيل Dos - كانت شركة CA تعامل هذه اللغة ك Package وليس كلغة برمجة - وعند سيطرة MS-Windows على السوق توقفت الشركة عن تطوير المنتج بنفس الاسم - واصدرته تحت اسم CA-Visual Objects مع وجود اختلافات بين المنتجين نظرا للتطور الذى حدث - ولم يكن المنتج الجديد منافسا قويا للمنتجات الاخرى مثل فيجوال بيسك و دلفى

Visual Objects®

*The Simpler C++, The Smarter Basic, The Easier Pascal,
The More Intelligent Xbase.*

For Microsoft Windows 2000 and XP Operating Systems



Designed to help individual and corporate developers rapidly deliver Windows applications, as well as extend those applications to the Internet, Visual Objects radically simplifies the integration of Windows and browser clients, web servers, middleware and back-end database systems. Visual Objects includes support for Internet client services, Internet server applications, ActiveX control creation, Jasmine multimedia applications and much more.

كما نلاحظ من تعرييق المنتج فانه يساعد المبرمجين الذى يعملون منفردين او فى فرق على تطوير تطبيقات Windows بسهولة - مع دعم الانترنت بشتى الوسائل المتاحة.

ونلاحظ ان الشركة تلوح بان اللغة تحاكي لغة C++ مع البساطة (حيث اشتهرت لغة C واسلافها بالتعقيد - وذلك فقط اذا تم مقارنة اللغة باللغات الاخرى - لكنها لغة جيدة وسلسة ومن يتقنها يعرف ذلك).

تلوخ الشركة ايضا بان اللغة نحاكى لغة ال BASIC مع وصفها بانها اكثر ذكاء حيث نظرا لسهولة لغة البيسك التى كانت قديما تخصص للمبتدئين ثم تحولت الى لغة برمجة قوية يجذب اليها المبتدئين والمحترفين على حد السواء كان يصفها مستخدميهما بالذكاء

وتلوح الشركة ايضا بانها تحاكي لغة Pascal التى طورت من اجل الجامعات فى البداية والتعليم الاكاديمى ثم صارت اللغات التجارية فى السوق - وتتمتع باسكال بالسهولة

وتلوح الشركة ايضا بان اللغة تحاكي لغات xBase وهذا امر موكد لانها اساسا احدى لغات xBase وتم وصفها بالادراك او الذكاء لانها لغة تتلائم مع التطبيقات التجارية بمرونة لايمكن وصفها ولا تدرك الا بالاستخدام

الامثلة البسيطة لا تميز بين اللغات

وبالتأكيد ان عدد لغات البرمجة كثير واحيانا تجد ان من الصعب التماس الفروق الجوهرية بين هذه اللغات حيث انها لاتظهر عادة الا عند تطوير التطبيقات الكبيرة - اما فى التطبيقات الصغيرة والامثلة البسيطة فان لغات البرمجة رغم تعددها تبدو وجوه مختلفة لعملة واحدة

وسوف ننظر لبعض الامثلة البسيطة التى توضح ذلك

اولا : برنامج لطباعة الشكل الهرمى

```
/* Author : Mahmoud Fayed
   Date   : 14/11/2006
   Job    : C/C++ Report, PROBLEM [1]
   */
#include<iostream.h>
#include<iomanip.h>
void main(void)
{
    int x,y,z,w,t ;
    x = 4; w = 1;
    for (y=1;y<=x;y++)
    {
        z = (x-y)+1; cout<< setw(z) ;
        for(t=1;t<=w;t++) cout<< "*";
        w=w+2; cout << setw(z) << '\n' ;
    }
    return ;
}
```

ونتيجة تنفيذ البرنامج نراها كما بالشكل

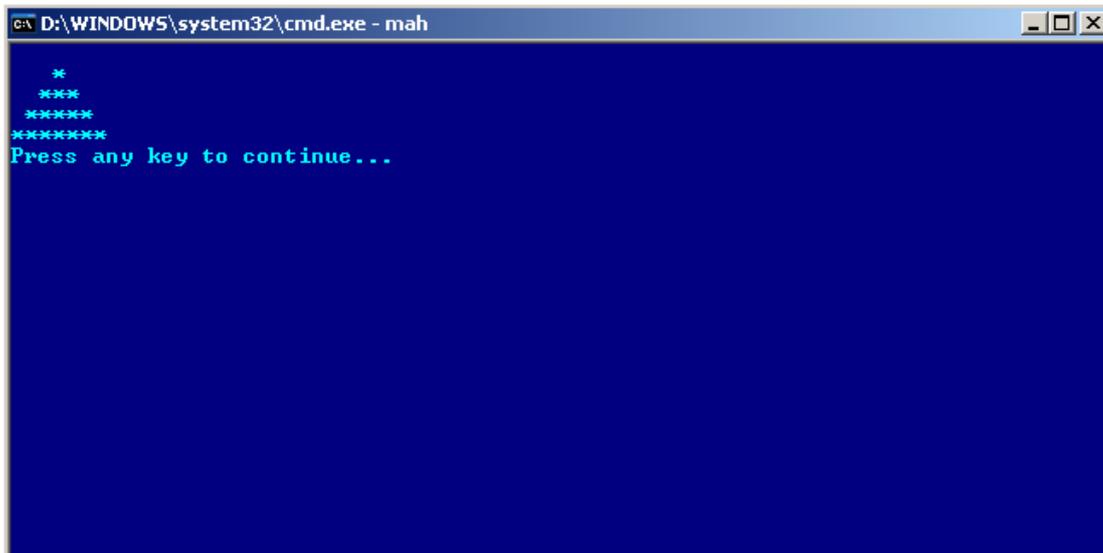
```
ca\ D:\WINDOWS\system32\cmd.exe
C:\MSUC\BIN>mah.exe
*
****
*****
*****
C:\MSUC\BIN>_
```

برنامج بلغة سى بلس بلس لرسم الشكل الهرمى

ويمكن كتابة نفس البرنامج باستخدام لغة Clipper كالتالى

```
*-----*
* Author : Mahmoud Fayed
* Date   : 14/11/2006
* Job    : Report Problem[1]
*-----*
set color to bg+/b,gr+/rb
clear
n=4
s=1
for x = 1 to n
  ? space(n-x)
  for y = 1 to s
    ?? "*"
  next
  s = s + 2
next
Wait
```

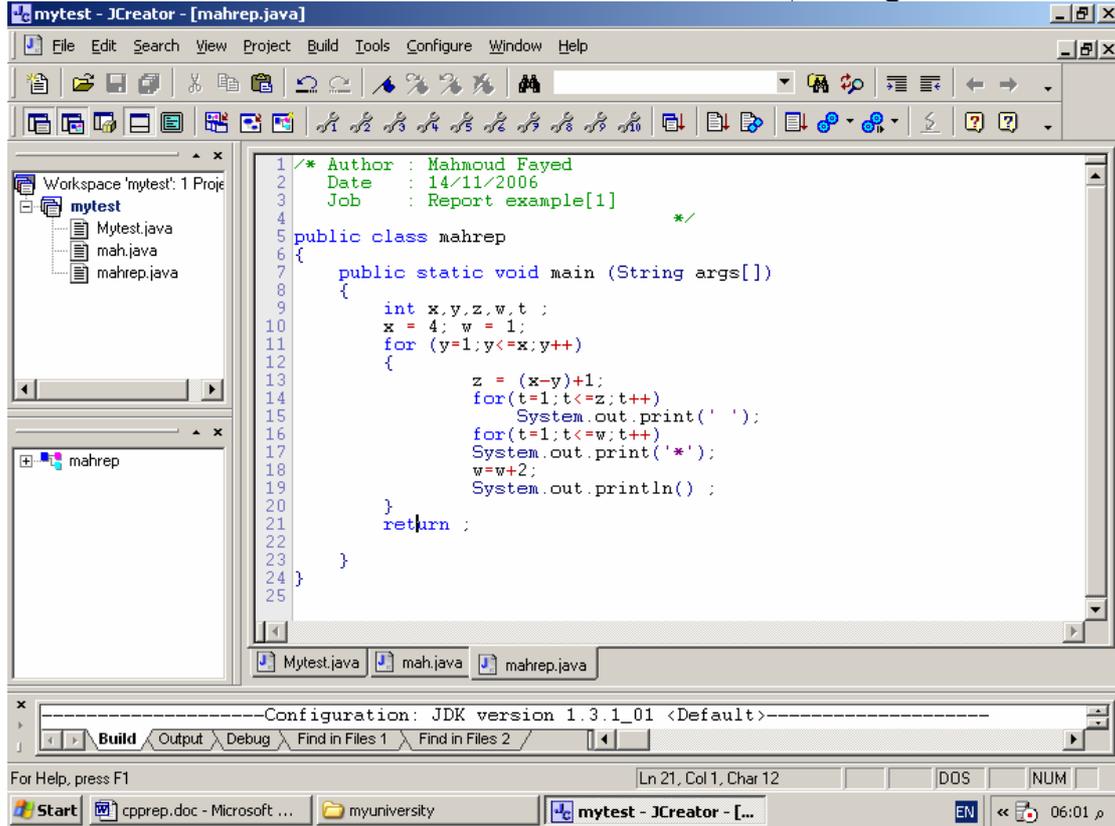
ونتيجة تنفيذ البرنامج كما بالشكل



```
C:\ D:\WINDOWS\system32\cmd.exe - mah
*
***
*****
*****
Press any key to continue...
```

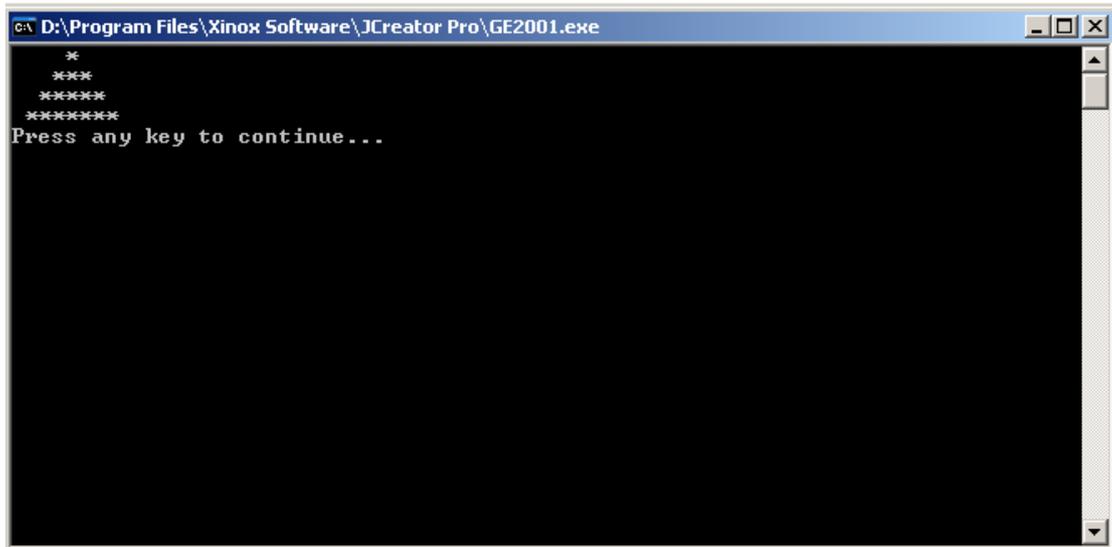
برنامج رسم الشكل الهرمى باستخدام لغة كليبر

وفيما يلي نفس البرنامج باستخدام لغة Java



برنامج رسم الشكل الهرمي باستخدام لغة Java

ونتيجة تنفيذ البرنامج كما بالشكل التالي

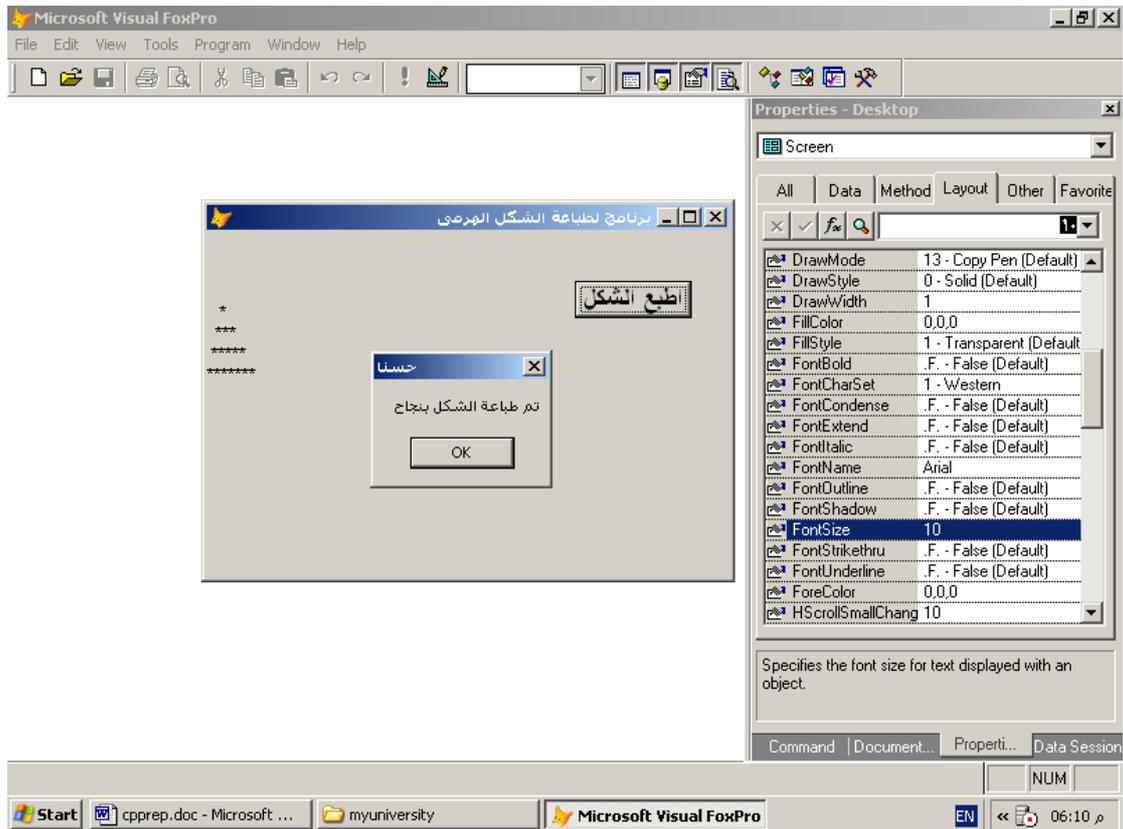


نتيجة تنفيذ برنامج الشكل الهرمي باستخدام لغة جافا

وفيما يلي نفس البرنامج باستخدام لغة Visual FoxPro

```
* Author : Mahmoud Fayed
* Date   : 14/11/2006
LOCAL n,s,x,y
n=4
s=1
for x = 1 to n
  ? space(n-x)
  FOR y = 1 to s
    ?? "*"
  ENDFOR
  s = s + 2
ENDFOR
MESSAGEBOX("حسنا",0,"تم طباعة الشكل بنجاح")
```

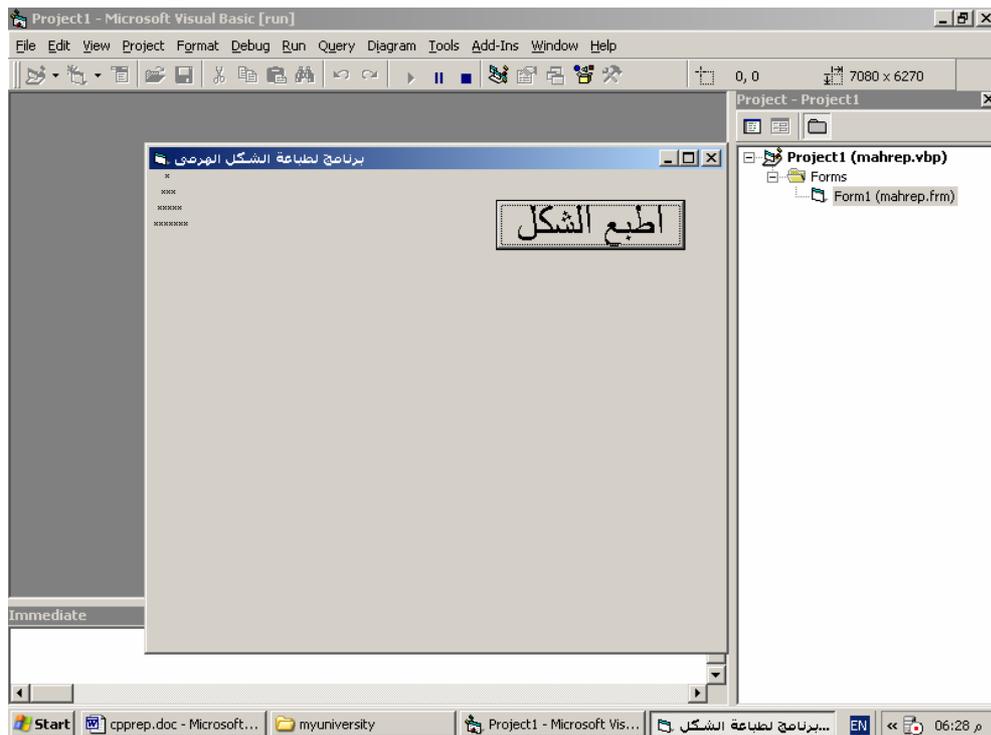
ونتيجة تنفيذ البرنامج كما بالشكل التالي



برنامج رسم الشكل الهرمي باستخدام لغة فيجوال فوكس برو

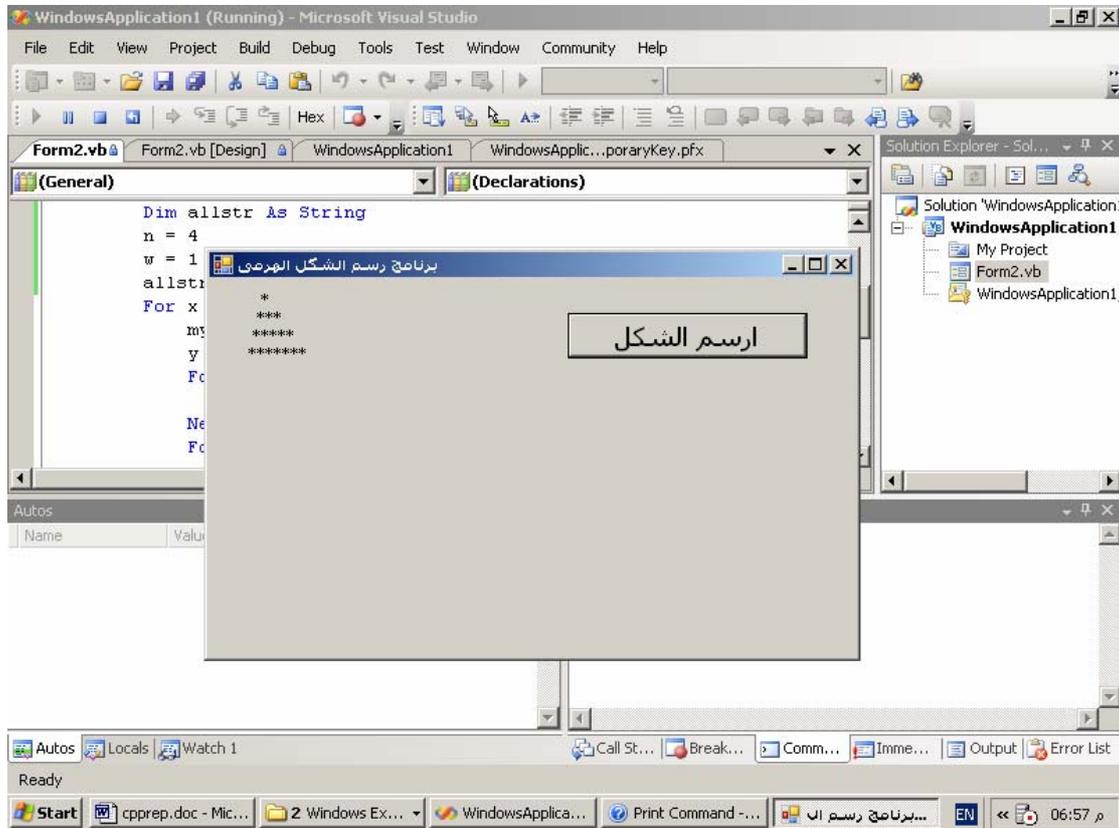
وفيما يلي نفس البرنامج باستخدام لغة فيجوال بيسك

```
Private Sub Command1_Click()  
' Author : Mahmoud Fayed  
' Date : 14/11/2006  
Dim x As Integer  
Dim y As Integer  
Dim z As Integer  
Dim w As Integer  
Dim n As Integer  
Dim mystr As String  
n = 4  
w = 1  
For x = 1 To n  
mystr = ""  
y = (n - x) + 1  
For z = 1 To y  
mystr = mystr + " "  
Next  
For z = 1 To w  
mystr = mystr + "*"  
Next  
w = w + 2  
Print mystr  
Next  
End Sub
```



برنامج رسم الشكل الهرمي باستخدام لغة فيجوال بيسك
وفيما يلي نفس البرنامج باستخدام لغة فيجوال بيسك دوت نت

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    ' AUTHOR : MAHMOUD FAYED
    ' DATE   : 14/11/2006
    Dim x As Integer
    Dim y As Integer
    Dim z As Integer
    Dim w As Integer
    Dim n As Integer
    Dim mystr As String
    Dim allstr As String
    n = 4
    w = 1
    allstr = ""
    For x = 1 To n
        mystr = ""
        y = (n - x) + 1
        For z = 1 To y
            mystr = mystr + " "
        Next
        For z = 1 To w
            mystr = mystr + "*"
        Next
        w = w + 2
        allstr = allstr + mystr + Chr(13)
    Next
    Label1.Text = allstr
End Sub
```

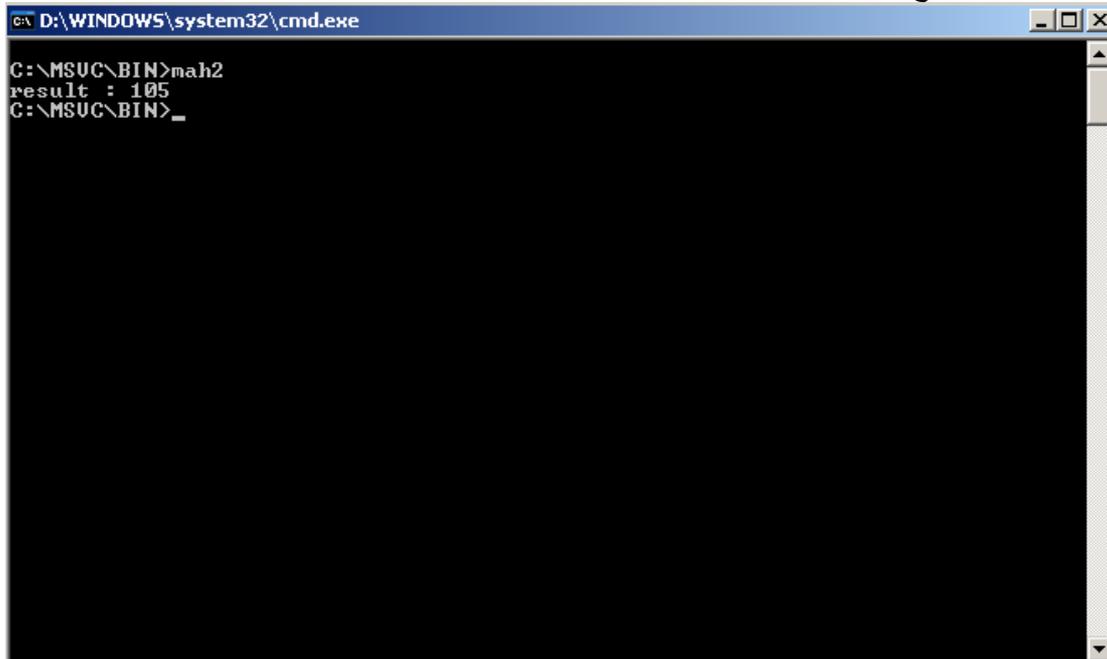


برنامج لرسم الشكل الهرمي باستخدام فيجوال بيسك دوت نت

ثانيا : برنامج لحساب المتوسط للارقام من ١٠٠ الى ١١٠

```
/* Author : Mahmoud Fayed
Date : 14/11/2006
Job : C/C++ Report, PROBLEM [2]
*/
#include<iostream.h>
#include<iomanip.h>
void main(void)
{
    int n1,n2,x;
    float r;
    n1 = 100;
    n2 = 110;
    r = 0 ;
    for (x=n1;x<=n2;x++)
        r = r + x ;
    r = r / ((n2-n1)+1);
    cout<< "result : " << r;
    return ;
}
```

ونتيجة تنفيذ البرنامج كالتالي



```
C:\WINDOWS\system32\cmd.exe
C:\MSUC\BIN>mah2
result : 105
C:\MSUC\BIN>_
```

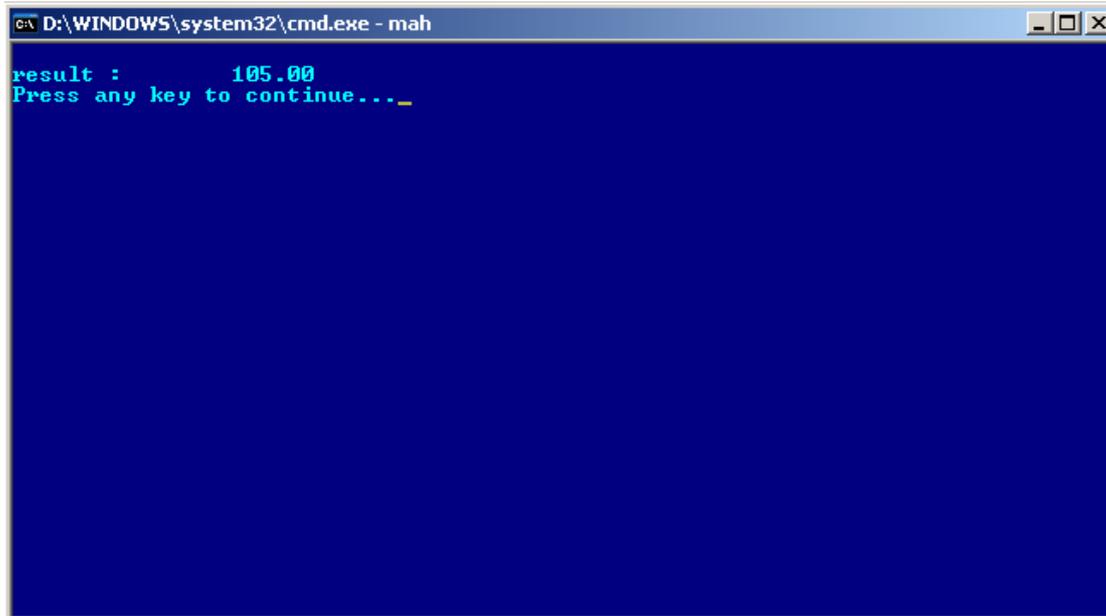
نتيجة تنفيذ برنامج حساب المتوسط باستخدام لغة سي\سي بلس بلس

ويمكن كتابة نفس البرنامج باستخدام لغة كليبر كالتالى

```
* Author : Mahmoud Fayed
* Date  : 14/11/2006
* Job   : Report Problem[2]
*
set color to bg+/b,gr+/rb
clear

n1=100
n2=110
r = 0
for x = n1 to n2
r = r + x
next
r = r / ((n2-n1)+1)
? "result : " + str(r)
wait
```

ونتيجة تنفيذ البرنامج كما فى الشكل



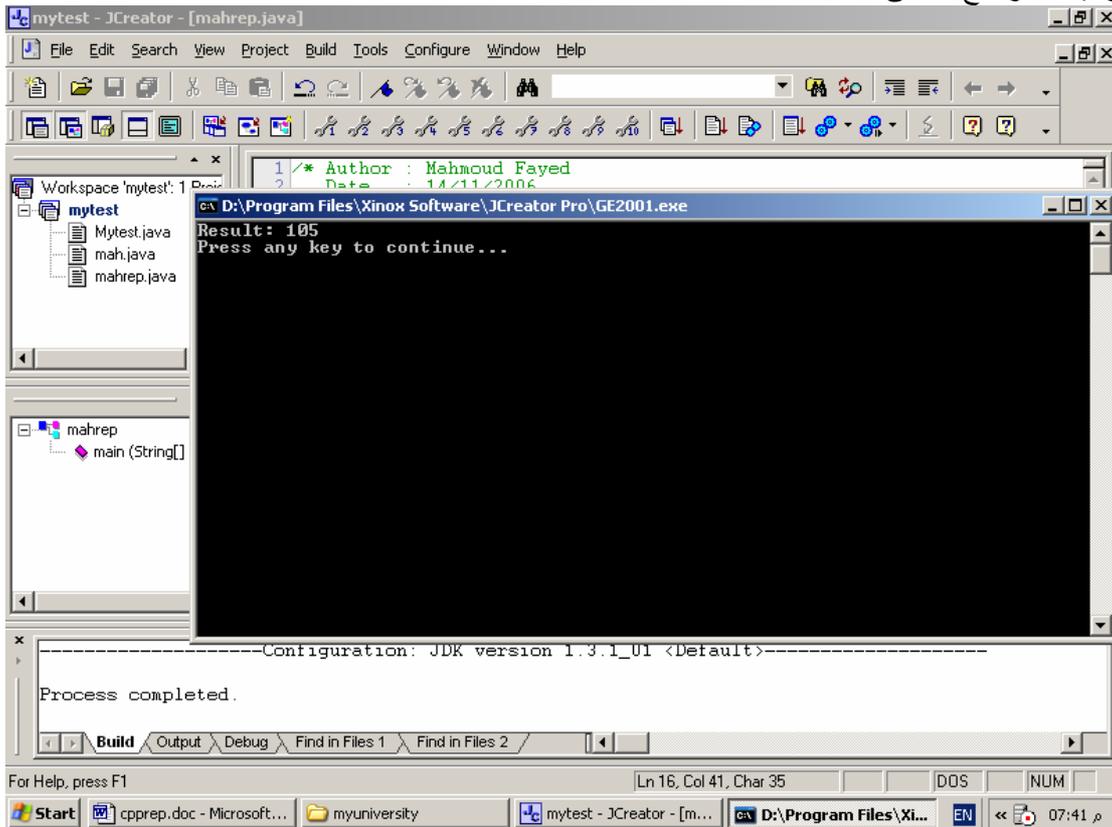
```
cmd D:\WINDOWS\system32\cmd.exe - mah
result : 105.00
Press any key to continue..._
```

نتيجة تنفيذ برنامج حساب المتوسط باستخدام لغة كليبر

ويمكن كتابة نفس البرنامج باستخدام لغة جافا

```
1 /* Author : Mahmoud Fayed
2    Date   : 14/11/2006
3    Job    : Report example[2]
4                                     */
5 public class mahrep
6 {
7     public static void main (String args[])
8     {
9         int n1,n2,x,r;
10        n1 = 100;
11        n2 = 110;
12        r = 0 ;
13        for (x=n1;x<=n2;x++)
14            r = r + x ;
15        r = r / ((n2-n1)+1) ;
16        System.out.println("Result: " + r);
17        return ;
18    }
19 }
20 }
21 }
```

ونتيجة البرنامج كما في الشكل

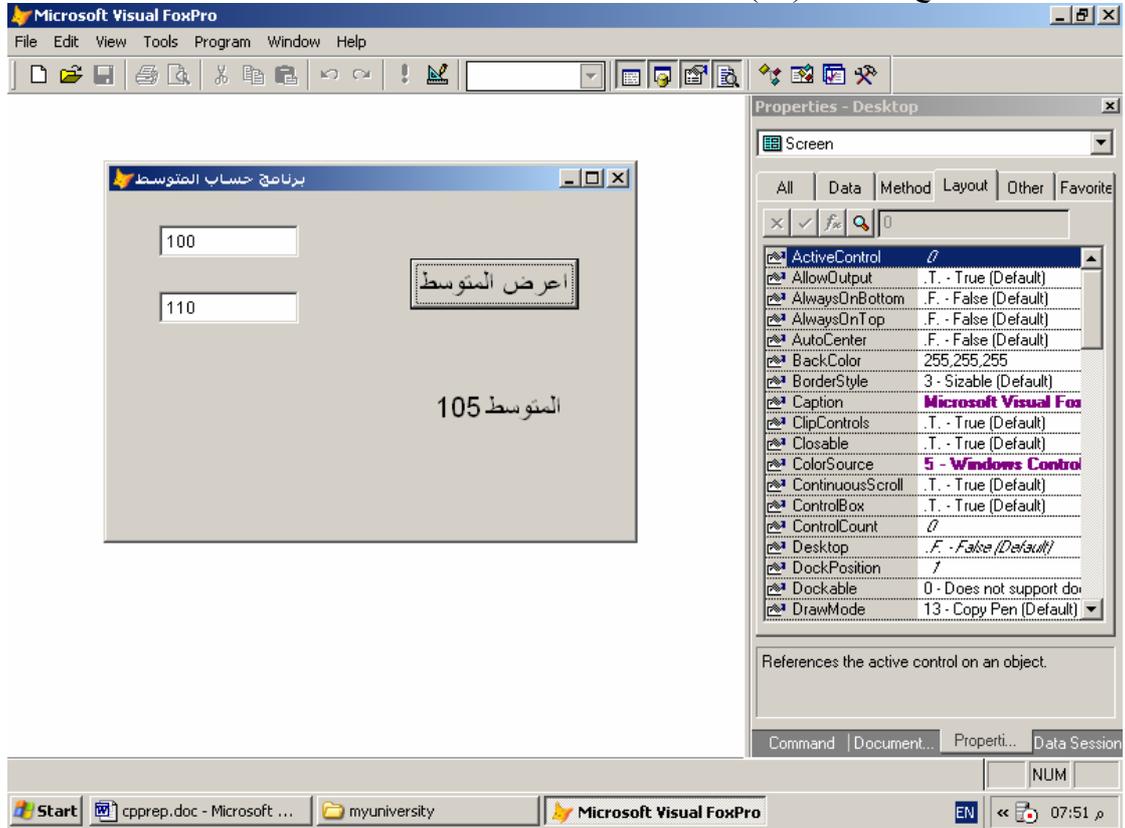


نتيجة تنفيذ البرنامج باستخدام لغة جافا

وفيما يلي نفس البرنامج باستخدام لغة فيجوال فوكس برو

```
* Author : Mahmoud Fayed
* Date   : 14/11/2006
LOCAL x,r
r = 0
n1 = VAL(thisform.text1.value)
n2 = VAL(thisform.text2.value)
FOR x = n1 TO n2
    r = r + x
NEXT
r = r / ((n2 - n1)+1)
thisform.label2.Caption = STR(r)
```

ونتيجة تنفيذ البرنامج كما بشكل (١١).



برنامج حساب المتوسط باستخدام لغة فيجوال فوكس برو

وفيما يلي نفس البرنامج ولكن بلغة فيجوال بيسك

```
Private Sub Command1_Click()  
    ' Author : Mahmoud Fayed  
    ' Date : 14/11/2006  
    Dim n1 As Integer  
    Dim n2 As Integer  
    Dim r As Integer  
    Dim x As Integer  
    n1 = Val(Text1.Text)  
    n2 = Val(Text2.Text)  
    r = 0  
    For x = n1 To n2  
        r = r + x  
    Next  
    r = r / ((n2 - n1) + 1)  
    Label2.Caption = Str(r)  
End Sub
```

ونتيجة البرنامج كما بالشكل



برنامج حساب المتوسط باستخدام لغة فيجوال بيسك

وفيما يلي نفس البرنامج باستخدام لغة فيجوال بيسك دوت نت

```
Public Class Form2

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        ' AUTHOR : MAHMOUD FAYED
        ' DATE : 14/11/2006
        Dim n1 As Integer
        Dim n2 As Integer
        Dim r As Integer
        Dim x As Integer
        n1 = Val(TextBox1.Text)
        n2 = Val(TextBox2.Text)
        r = 0
        For x = n1 To n2
            r = r + x
        Next
        r = r / ((n2 - n1) + 1)
        Label2.Text = Str(r)
    End Sub

    Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

    End Sub
End Class
```

ونتيجة تنفيذ البرنامج كما بالشكل التالي



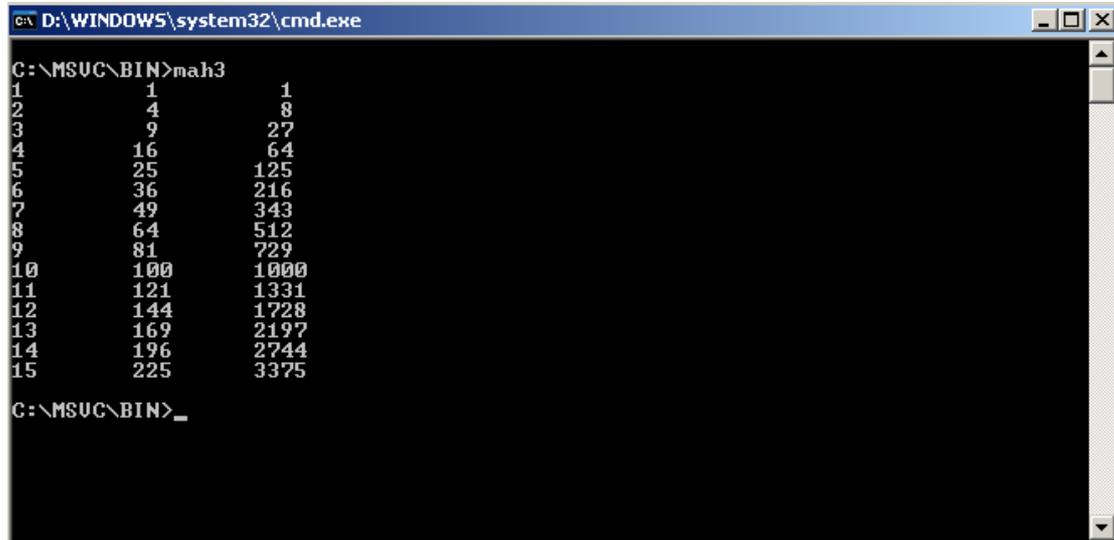
برنامج حساب المتوسط باستخدام لغة فيجوال بيسك دوت نت

ثالثا : برنامج عد الارقام من ١ الى ١٥ وتربيعها وتكعيبها

البرنامج بلغة سي بلس بلس كالتالي

```
/* Author : Mahmoud Fayed
Date : 14/11/2006
Job : C/C++ Report, PROBLEM [3]
*/
#include<iostream.h>
#include<iomanip.h>
void main(void)
{
    int n1,n2,x;
    n1 = 1;
    n2 = 15;
    for (x=n1;x<=n2;x++)
    {
        cout<< x << setw(10) << x*x << setw(10) << x*x*x;
        cout<< '\n' ;
    }
    return ;
}
```

ونتيجة تنفيذ البرنامج كالتالي



```

C:\WINDOWS\system32\cmd.exe
C:\MSUC\BIN>mah3
1          1          1
2          4          8
3          9         27
4         16         64
5         25        125
6         36        216
7         49        343
8         64        512
9         81        729
10        100       1000
11        121       1331
12        144       1728
13        169       2197
14        196       2744
15        225       3375
C:\MSUC\BIN>_
```

برنامج عد الارقام وتربيعها وتكعيبها بلغة سي بلس بلس

وفيما يلي نفس البرنامج ولكن باستخدام لغة كليبر

```
*
* Author : Mahmoud Fayed
* Date  : 14/11/2006
* Job   : Report Problem[3]
*
set color to bg+/b.gr+/rb
clear

n1=1
n2=15
for x = n1 to n2
? str(x) + space(20) + str(x*x) + space(20) + str(x*x*x)
next
wait
```

انظر الشكل لتري نتيجة تنفيذ البرنامج

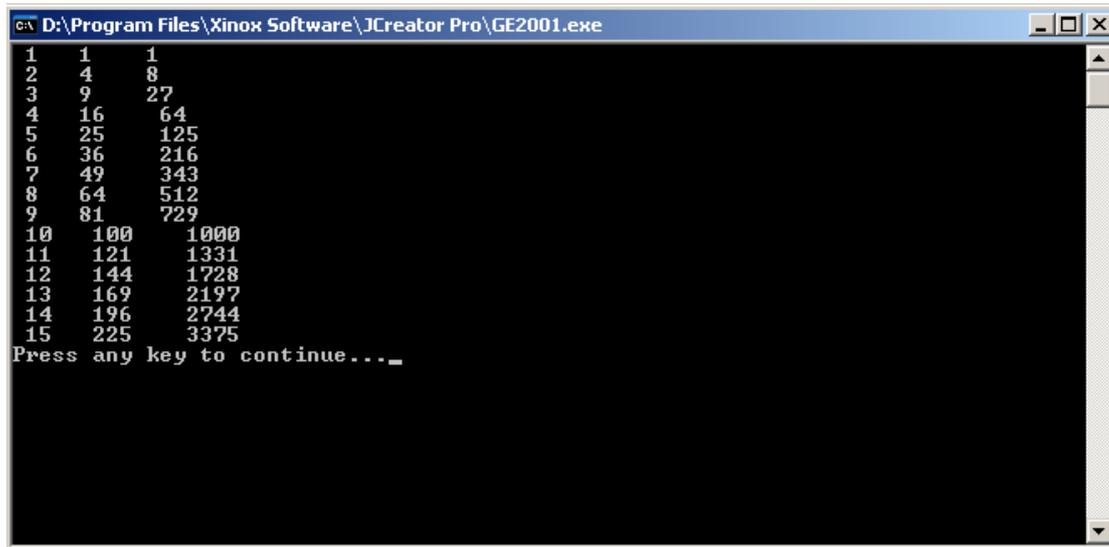
```
1          1          1
2          4          8
3          9         27
4         16         64
5         25        125
6         36        216
7         49        343
8         64        512
9         81        729
10        100       1000
11        121       1331
12        144       1728
13        169       2197
14        196       2744
15        225       3375
Press any key to continue...
```

برنامج التربيع والتكعيب باستخدام لغة كليبر

وفيما يلي نفس البرنامج ولكن بلغة جافا

```
1 /* Author : Mahmoud Fayed
2    Date   : 14/11/2006
3    Job    : Report example[3]
4
5 public class mahrep
6 {
7     public static void main (String args[])
8     {
9         int n1,n2,x;
10        String mystr;
11        n1 = 1;
12        n2 = 15;
13
14        for (x=n1;x<=n2;x++)
15        {
16            mystr = " " + x + "      " + (x*x) + "      " + (x*x*x) ;
17            System.out.println(mystr);
18        }
19
20        return ;
21
22    }
23 }
24
```

ونتيجة تنفيذ البرنامج كما بالشكل



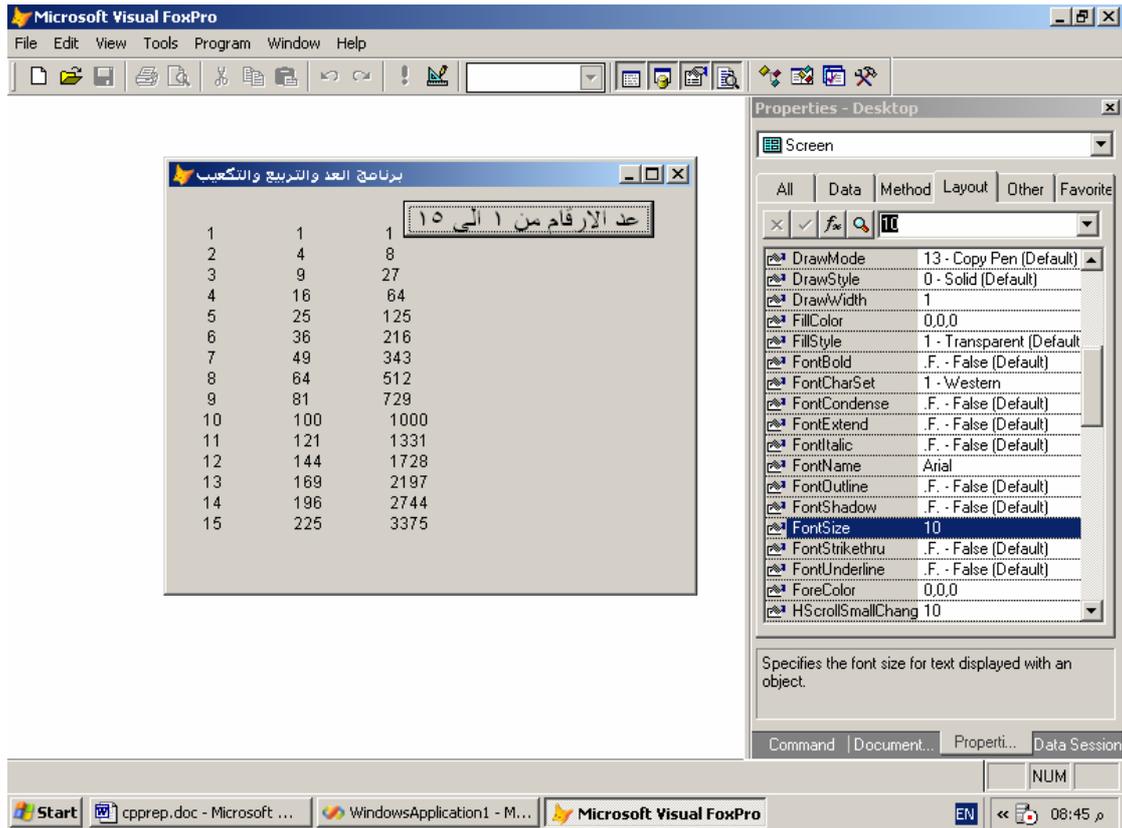
```
1 1 1
2 4 8
3 9 27
4 16 64
5 25 125
6 36 216
7 49 343
8 64 512
9 81 729
10 100 1000
11 121 1331
12 144 1728
13 169 2197
14 196 2744
15 225 3375
Press any key to continue...
```

نتيجة تنفيذ البرنامج

وفيما يلي نفس البرنامج بلغة فيجوال فوكس برو

```
LOCAL x
FOR x = 1 TO 15
? STR(x) + SPACE(10) + STR(x*x) + SPACE(10) + STR(x*x*x)
NEXT
```

ونتيجة تنفيذ البرنامج كما بالشكل

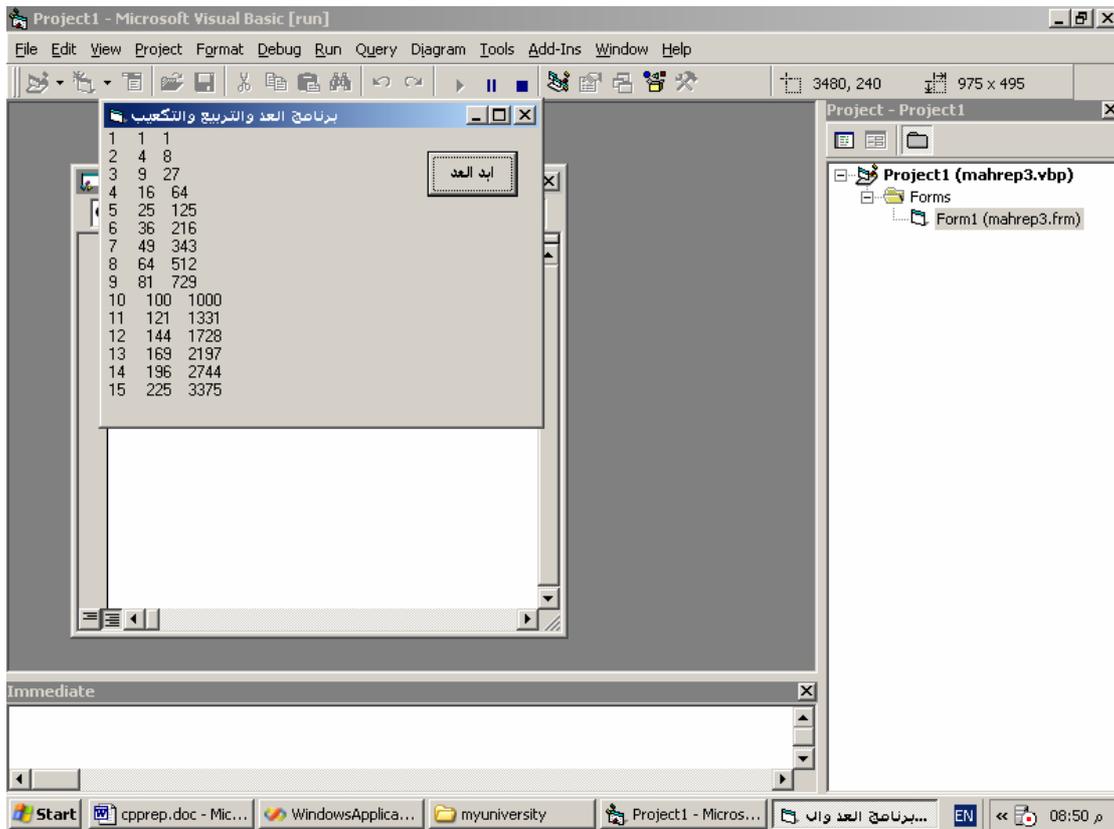


نتيجة تنفيذ برنامج العد والترتيب والتكعيب بلغة فيجوال فوكس برو

وفيما يلي نفس البرنامج باستخدام لغة فيجوال بيسك

```
Private Sub Command1_Click()
Dim x As Integer
Dim mystr As String
For x = 1 To 15
mystr = Str(x) + " " + Str(x * x) + " " + Str(x * x * x)
Print mystr
Next
End Sub
```

انظر الشكل لتري نتيجة البرنامج

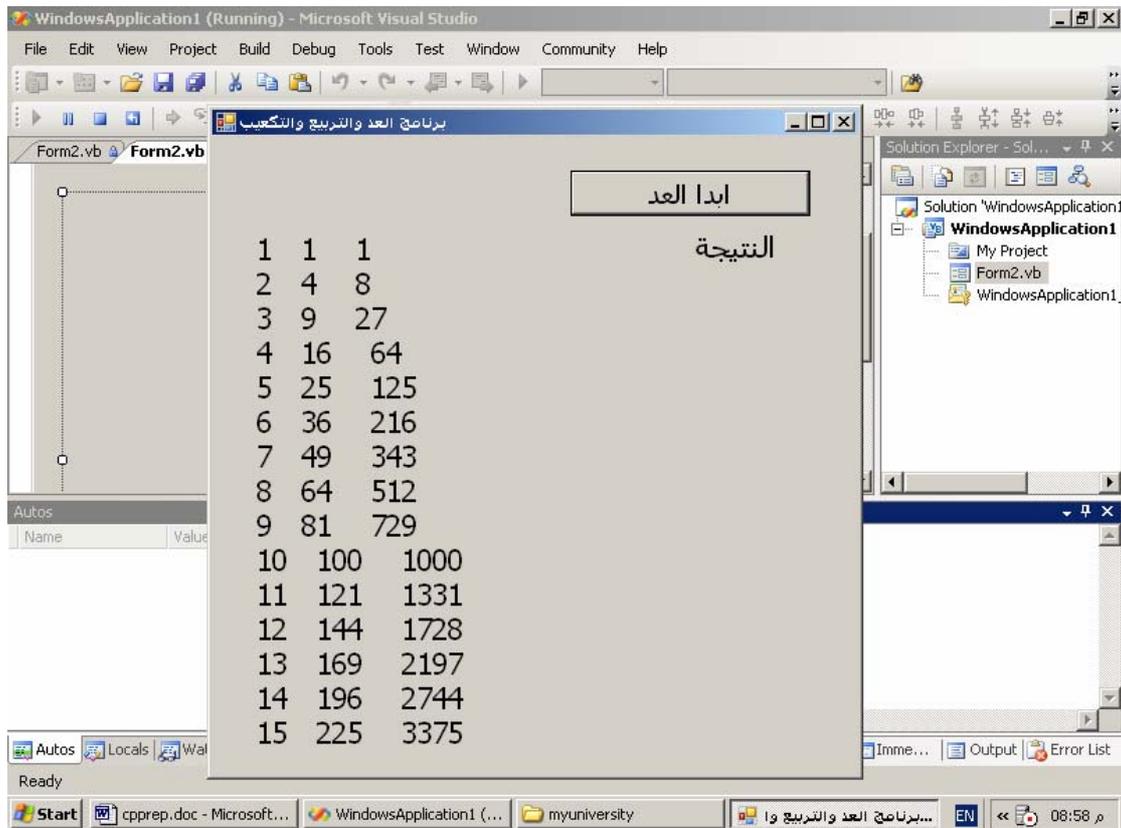


نتيجة تنفيذ برنامج العد والتربيع والتكعيب باستخدام فيجوال بيسك

وفيما يلي نفس البرنامج باستخدام فيجوال بيسك دوت نت

```
' AUTHOR : MAHMOUD FAYED
' DATE   : 14/11/2006
Dim x As Integer
Dim r As String
For x = 1 To 15
    r = r + Str(x) + " " + Str(x * x) + " " + Str(x * x *
x) + Chr(13)
Next
Label2.Text = r
```

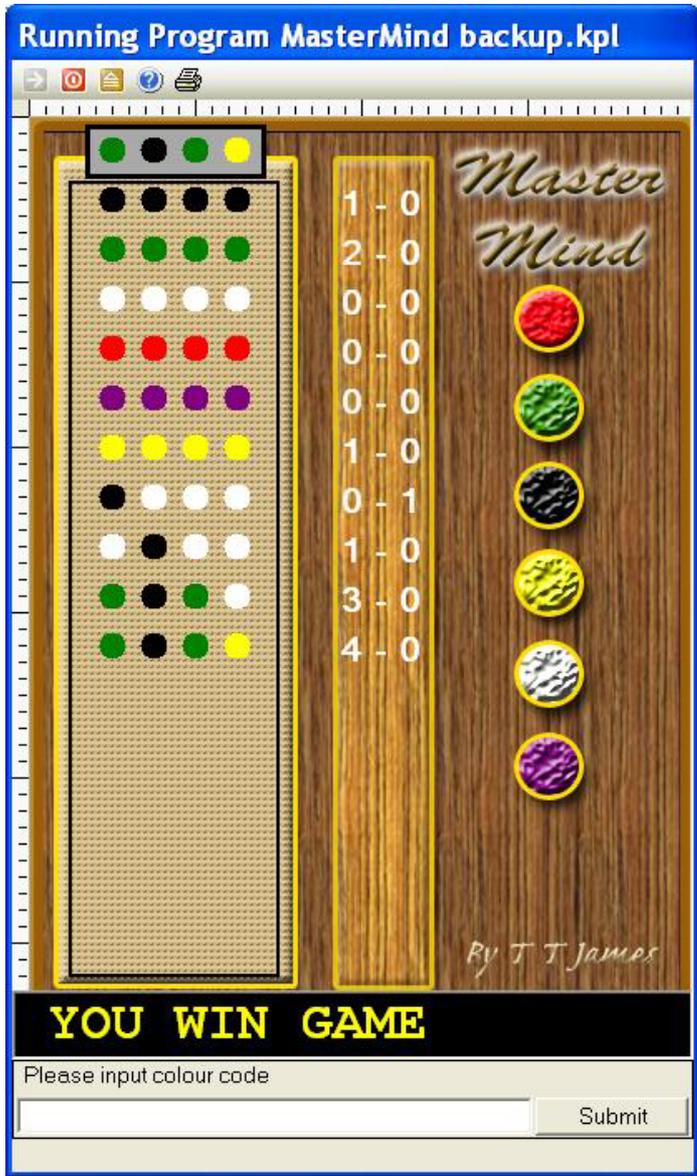
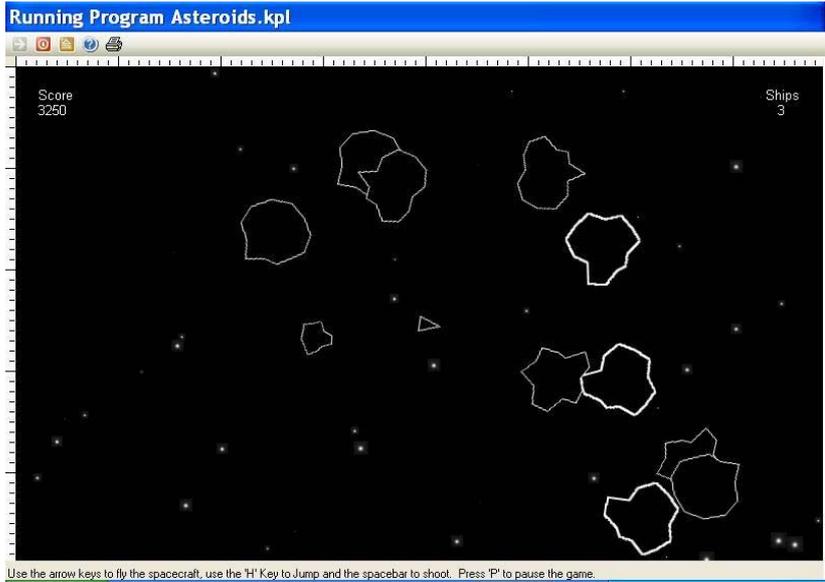
ونتيجة التنفيذ كما بشكل



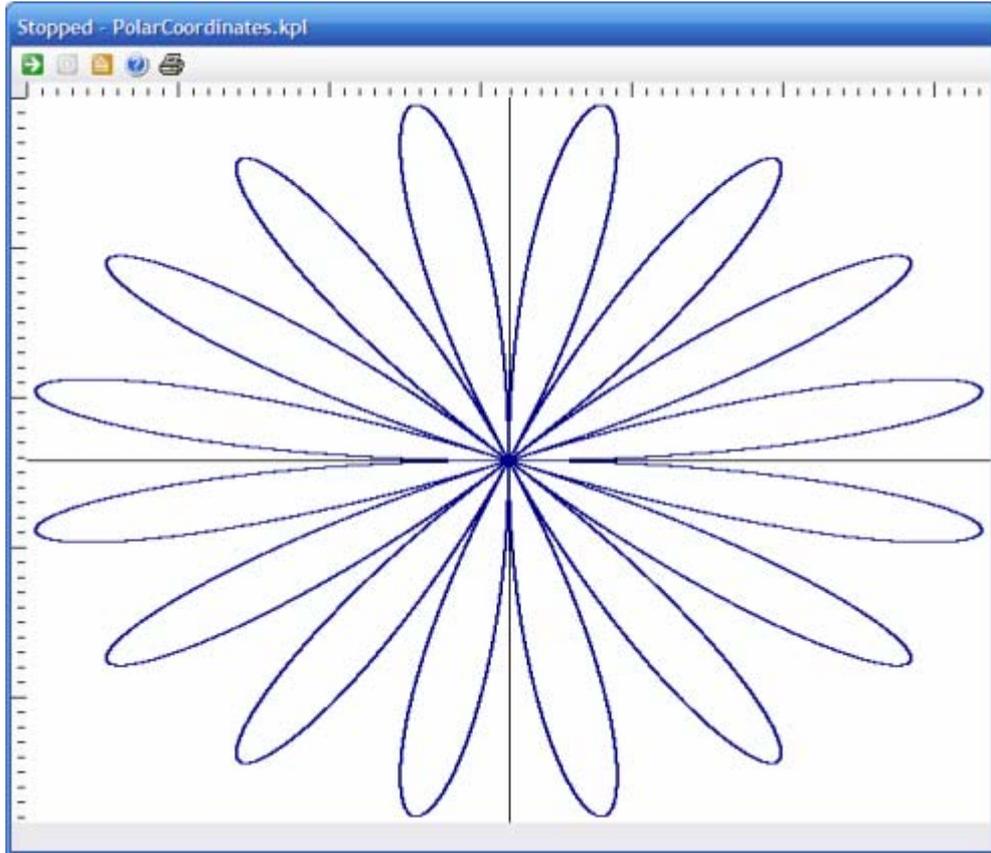
نتيجة برنامج العد والتربيع والتكعيب بلغة فيجوال بيسك دوت نت

أفكار خلف لغات البرمجة الجديدة

والجدير بالذكر ان العالم لا يحتاج لغة جديدة وانما يحتاج افكار جديدة قد تكون اساس للغات جديدة ومن الافكار المثيرة فى هذا العالم فكرة انشاء لغة برمجة مخصصة للاطفال Kids والتي تعتمد على المتعة Fun من اجل التعلم لهذا فهى مخصصة للاعب التي يتفاعل معها الصغار والاشكال التالية عبارة عن العاب تم تطويرها بتلك اللغة KPL







نموذج سير العمليات باستخدام C#.NET

سبق وان تحدثنا فى الجزء الاول عن نموذج سير العمليات ومدى اهميته فى بناء النظم - ولعل البعض يطرح سؤالاً - هل يمكن ان اقوم بكتابة نموذج سير العمليات الخاص بى داخل لغة متطورة مثل سى شارب - بالتأكيد نعم - والمثال التالى يوضح ذلك

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Windows.Forms;

namespace mahmoudlovecsharp
{
    class SS_EventManager
    {
        public bool shutdown;
        const int objscount = 1000; // events count
        object[] myobjs = new object[objscount];
        bool[] myswitches = new bool[objscount];
        int lastobject = 0;
        public event EventHandler ActiveEvent;
        public int NewEvent(EventHandler myevent)
        {
```

```

        if (lastobject < objscount)
        {
            myobjs[lastobject] = myevent;
            lastobject++;
            return lastobject-1;
        }
        return -1;
    }

    public void SetSwitch(int EventID, bool SwitchStatus)
    {
        if (EventID < objscount)
        {
            myswitches[EventID] = SwitchStatus;
        }
    }

    public void StartUp()
    {
        int x;
        shutdown = false;
        for (x = 0; x < objscount; x++)
        {
            myswitches[x] = true;
        }
    }

    public void DoEvents()
    {
        int x;
        while (true)
        {
            Application.DoEvents();
            if (shutdown == true )
            {
                break;
            }
            for (x = 0; x < objscount; x++)
            {
                ActiveEvent = (EventHandler)myobjs[x];
                if (ActiveEvent != null && myswitches[x] == true)
                {
                    ActiveEvent(this, new EventArgs());
                }
            }
        }
    }
}

class mahclass
{
    SS_EventManager myniceobj;
    public void hello()
    {
        int EventID;
        myniceobj = new SS_EventManager();
        myniceobj.StartUp();
        EventID = myniceobj.NewEvent(new EventHandler(myclick));
        // myniceobj.SetSwitch(EventID, false); // switch test
    }
}

```

```

        EventID = myniceobj.NewEvent(new
EventHandler(myclick2));
        EventID = myniceobj.NewEvent(new
EventHandler(CheckExit));
        myniceobj.DoEvents();
    }
    private void myclick(Object sender, EventArgs e)
    {
        if (Form1.ActiveForm != null)
        {
            Form1.ActiveForm.Text = "nice ";
        }
    }
    private void myclick2(Object sender, EventArgs e)
    {
        if (Form1.ActiveForm != null)
        {
            Form1.ActiveForm.Text = "good ";
        }
    }
    private void CheckExit(Object sender, EventArgs e)
    {
        if (Form1.ActiveForm == null)
        {
            myniceobj.shutdown = true;
        }
    }
}
}
}

```

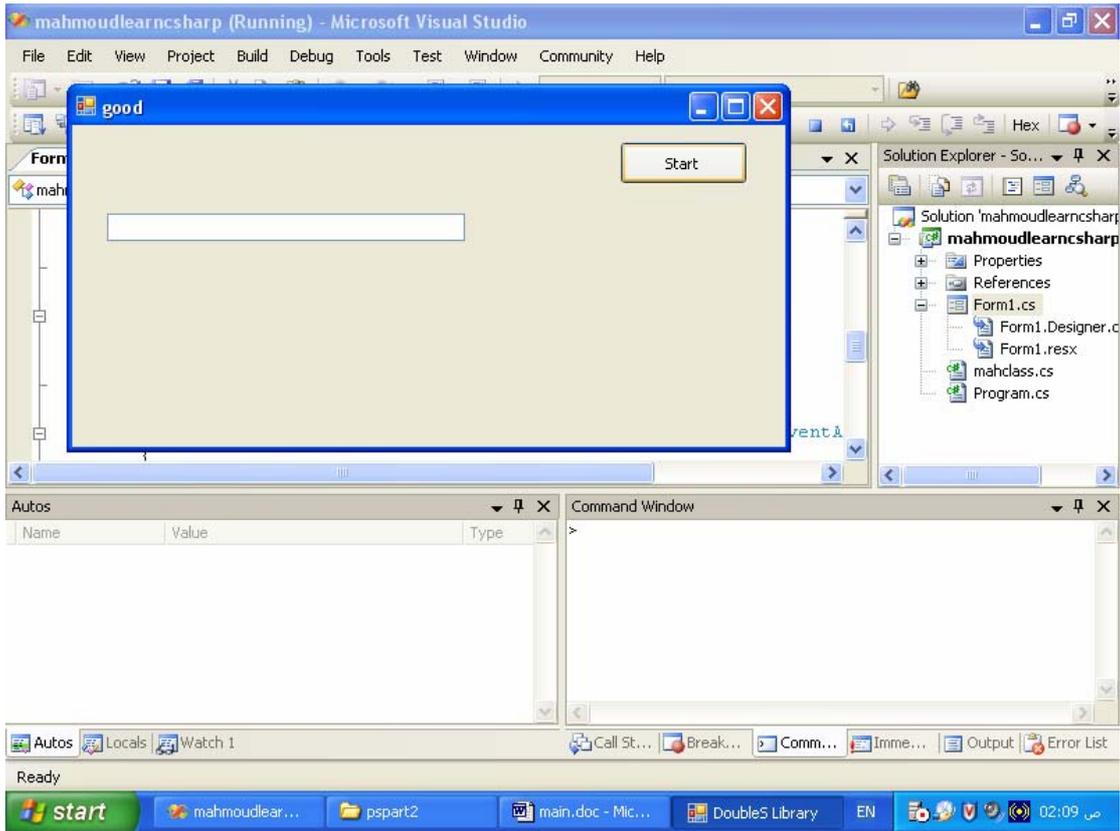
فى هذا المثال نضع فى حدث
Click
الخاص باحد الازار

```

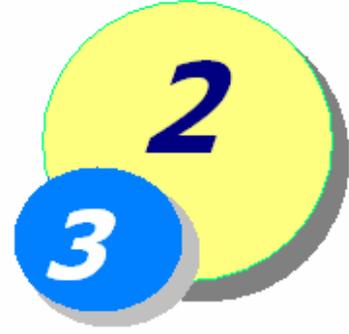
mahclass myobj;
myobj = new mahclass();
myobj.hello();

```

بمجرد التنفيذ سوف يستمر نظام الاحداث فى العمل - حيث يغير
الخاصية caption الخاصة بالنموذج form1 باستمرار وبدون توقف
من "good" الى "nice" ثم الى "good" ثم "nice" وهكذا حتى
يتم اغلاق النموذج



البرمجة بدون كود RPWI



منذ اكثر من عام كتبت مقالا بعنوان (البرمجة بدون كود باستخدام DoubleS) وقد حصل عليه اكثر من عشرة الاف شخص وتلقيت العديد من الرسائل كان مجملها (لم نفهم شىء) ومن القراء من كان لايهتم بالموضوع اساسا - وانما راسلنى كى ارشده الى كيفية تعلم البرمجة - لهذا كنت امام فئتين من المراسلين و قد تحمست ان ارد على كلا الفئتين.

بالنسبة لمن كان يرغب فى تعلم البرمجة فقد قدمت كتيب بعنوان (اسرع طريق لتعلم البرمجة) حتى يكون مقدمة سريعة نحو المجال - وقد نجح هذا الكتيب رغم بساطته وصغر حجمة الى ان وصل عدد القراء اكثر من اثنين وثلاثين الف قارىء.

اما بخصوص الفئة الاخرى التى لم تفهم اى شىء من الموضوع وهذا كان توقعى لان الهدف من ذلك المقال ان ذلك لم يكن الشرح المفصل وانما كان الهدف منه بمنتهى الصدق والامانة عمل دعاية لمشروع مجانى قمت بعمله ورايت فيه الفائدة واحببت ان ينتشر.

ولكن بما اننى اعشق القارى المحب للغة العربية فقد شرعت فى تقديم موسوعة (اعمق اسرار البرمجة) لكى تشمل ماتوصلت اليه فى خلال عشرة سنوات من الكفاح فى عالم البرمجة - فمن الموكد ان تشمل تلك الموسوعة - الشرح الكامل للبرمجة بدون كود - وكن عزيزى القارىء على ثقة فى انك سوف تتعلم شيئا مثيرا للغاية وممتع فى الاستخدام وفعال فى الاداء - ولا اخفى عليك سرا - ان البرمجة بدون كود هى حصاد ماقت بتعلمه طوال حياتى فى البرمجة وانها الثمرة التى اشكر الله عز وجل عليها.

انك حين تتعلم البرمجة بدون كود خلال شهر واحد - فانك تصل فى الاداء الى مستوى مبرمج قضى ١٠ سنوات من عمره منعزلا

تقريبا عن العالم لايفكر سوى فى الحاسب والبرمجة - وفى ان يقدم شيئا جديدا مفيدا ومميزا.
لقد تكلمت كثيرا الان - وفى الواقع ان الوقت ان نبدا وانا اعلم انك متحمس لتلك الفقرة من الكتاب وخاصة ان جزء الفلسفة ليس عملى بصورة مباشرة - وان الجزء الخاص بالتكنولوجيا المعاصرة قد يكون غير شامل - واغلب محتوياته معروفة لديك - ولهذا فان جزء البرمجة بدون كود هو الوجه الحقيقى لهذا الكتاب حتى يستحق اسمه الغالى الثمن (اعمق اسرار البرمجة)

تاريخ البرمجة بدون كود

انه تاريخ ليس بهين - فان هذا المجال تطور عبر الزمن تطور كبير - وقد قدم لنا الادوات التالية

- الحزم المتخصصة Packages
- المصمات Designers
- مولد الاكواد Code Generator
- المحرك Engine
- المعالجات Wizards
- بيئة التطوير المتكاملة IDE
- محيط التطوير Framework

حتى نكون على حذر اثناء استقبالنا لتلك المادة العلمية الصعبة و يمكننى ان اقول بشيء من الثقة والجرأة انها فى منتهى التعقيد بالنسبة للمبرمج المتمرس - لذا علينا ان نعلم جيدا انها قفزة نوعية - تشمل العديد من المفاهيم الجديدة - لذلك لا تتسرع فى استقبالك للمعلومات - لانها ليست مجرد معلومات وانما تخفى ورائها العديد من المفاهيم شديدة التشابك.

ان الحزم المتخصصة Packages هى عبارة عن برمجيات متخصصة فى عمل معين - هذا العمل كان يؤدي قديما من خلال البرمجة - لكن الحزم تقوم باداء هذا العمل بدون الحاجة الضرورية الى البرمجة
o مثل الحزم المتخصصة فى قواعد البيانات وتسمى DBMS اى نظم ادارة قواعد البيانات

○ الحزم المتخصصة فى تطبيقات عرض البيانات Data Show والكتب الالكترونية eBooks المصمات Designers

○ هى عبارة عن مجموعة من البرمجيات تتخصص فى اداء مهمة معينة - وتنجح فى اداء تلك المهمة بدون الحاجة الى البرمجة بنسبة تتخطى ٩٠%

▪ مثل مصمم النماذج Form Designer

• يتخصص فى تصميم نوافذ وشاشات التطبيقات ويقوم باداء تلك المهمة بالتحديد بدون الحاجة للتدخل البرمجى ولكن لا ينجح فى الحالات الخاصة والتي نحتاج فيها لادوات جديدة

▪ مثل مصمم التقارير Report Designer

• يتخصص فى تصميم تقارير تطبيقات قواعد البيانات بدون الحاجة للتدخل البرمجى ولكن لا ينجح فى الحالات الخاصة التى تتطلب عمليات غير متوقعة حسب المشروع تحت التطوير

▪ مثل مصمم القوائم Menu Designer

• يتخصص فى تصميم القوائم ذات المستويات المختلفة ولكن لا ينجح فى الحالات التى يرغب فيها المبرمج فى الابداع واستخدام ملامح جديدة للقوائم.

مولد الاكواد Code Generator

○ هو عبارة عن نوع معين من البرمجيات مهمته تكمن فى توليد مجموعة من الاكواد لتلائم حالات معينة وبالتالي يسهل على المبرمج ويوفر عليه الكثير من الوقت - ويظهر استخدام مولد الاكواد فى الامور التى يصعب تصور حزمة Package تمثلها او مصمم Designer يتخصص فيها

المحرك Engine

○ هو عبارة عن مولد للاكواد ولكن الاكواد التى يولدها لا يستخدمها المبرمج مباشرة - وانما يستخدمها برنامج اخر - وهو الصورة الوحيدة التى تظهر بها البرمجة بدون كود فى وقت Runtime.

المعالجات Wizards

- هى عبارة عن برمجيات تقوم بعرض شاشات او نماذج تتفاعل مع المستخدم - اى تقدم له مجموعة من الخيارات وبناء عليها يتم استخدام مولد اكواد معين لكى يقوم باستخراج الكود المطلوب.

بيئة التطوير المتكاملة IDE

- هى عبارة عن برمجيات تشمل بداخلها العديد من البرمجيات التى توفر فرصة تطوير التطبيقات بدون الحاجة الضرورية لاستخدام ادوات اخرى (على اعتبار انها متكاملة - لكن لا كامل الا الله - ونحن نخرج منها كثيرا)
- من التطبيقات التى تشملها بيئة التطوير المتكاملة برمجيات التصميم والمعالجات - ومدير المشاريع - وغيرها الكثير
- يتم تفسير IDE على انها Integrated Development Environment اى بيئة التطوير المتكاملة فى جميع منتجات مايكروسوفت ماعدا فيجوال فوكس برو والتى تفسرها على انها Interactive Development Environment اى بيئة التطوير المتفاعلة - والتى تطور التطبيقات بناء على التفاعل (من خلال المعالجات الكثيرة) مع المبرمج (لكن ليس دائما فهى ليست متفاعلة الى مالانهاية - والكود وسيلة اساسية للتعبير عما بداخلك)

محيط التطوير Framework

- هو عبارة عن مجموعة من الادوات القابلة لاعادة الاستخدام والمتخصصة فى شىء معين
- كان ذلك سبب مشكلة كبيرة حينما اطلقت مايكروسوفت محيط تطوير الدوت نت حيث ظن المبرمجين ان استخدامه قاصرا على تطوير تطبيقات الانترنت - ولهذا غيرت مايكروسوفت الدعاية الخاصة بمحيط التطوير لتركز على انه مصمم لتطوير مختلف انواع التطبيقات ويتميز فى تطبيقات الانترنت (فى الواقع هذا تعديا على مصلح علمى - ولكن هذا من حق مايكروسوفت - لانها

احد اهم قادة التكنولوجيا - وعلم الحاسوب اصبح
يستقبل من التكنولوجيا اكثر مما يعطى لها).

ما هى البرمجة بدون كود؟

سوال هام جدا . ببساطة شديدة - ان البرمجة هى وسيلة
حتى يقوم الحاسب بعمل ما نريد - وان كتابة الشفرة المصدرية
اي Source Code الكود هى احد الوسائل لكى نقوم بالبرمجة.

ان هناك وسائل اخرى - تتم عن طريق عزل المبرمج عن الكود -
هذه الوسائل - تسمى البرمجة بدون كود.

ما هى اهم وسائل البرمجة بدون كود بالنسبة للمبرمج؟

ان اهم وسائل البرمجة بدون كود بالنسبة للمبرمج هى
المعالجات والمصمات نظرا للكفاءة التى لا بأس بها والسرعة
الخارقة والتاثير المباشر على زيادة انتاجية المبرمج.

ما هى اهم وسائل البرمجة بدون كود بالنسبة للمستخدم؟

بلا شك الحزم التطبيقية الجاهزة Packages مثل برنامج
Microsoft PowerPoint وبرنامج Microsoft Access

ماذا عن RPWI ؟

حسنا - لقد تحدثت عن تاريخ البرمجة بدون كود - وكلمة تاريخ
لم اذكرها لاننى ذكرت سنة ؟؟؟؟ او سنة ؟؟؟؟ فانا لم افعل ذلك
وانما لاعنى اننى اتحدث عن شىء حدث فى الماضى - وبما ان
هناك ماضى وقد عرفناه - فما هو الحاضر ؟

انه الذى تسال عنه - انها تكنولوجيا RPWI هى احدث ما ورد
فى تكنولوجيا البرمجة بدون كود

Real Programming Without RPWI تعنى Instructions (CODE)

اي البرمجة الفعلية بدون كود - لاحظ ان كلمة Instructions اى تعليمات تم استخدامها للدلالة على الكود Code وليس على التعليمات بصفة عامة - لان البرمجة اساسا هى اعطاء تعليمات للحاسب.

وكلمة Real تعنى حقيقى وذكرها هنا يشير الى البرمجة الحقيقية - اى انها برمجة بدون كود - ولكنها حقيقية بمعنى اوضح انها تناسب تطبيقات الواقع وجديرة بالاستخدام من كل مبرمج جاد - وليست لمجرد الاستخدامات البسيطة و فقط.

ان RPWI هى تكنولوجيا مبتكرة من قبل مؤلف هذا الكتاب (اخوك فى الله - محمود فايد) صممت لكى تكون اول تكنولوجيا للبرمجة بدون كود تنجح تماما فى تحقيق الاهداف التالية

- ✚ عزل المبرمج عن قواعد Syntax اللغة
- ✚ تعتمد على اى لغة برمجة فى الخلفية - مثل سى شارب او جافا C# or Java على سبيل المثال.
- ✚ توفر واجهة للمستخدم باى لغة يفهمها الانسان مثل اللغة العربية او الانجليزية على سبيل المثال.
- ✚ ان تكون قابلة للإمتداد بحيث تكون - بلا حدود - وبلا قيود تحرم المبرمج من اداء ما يريد
- ✚ يمكن ان يتم إمتدادها بدون كود ايضا.

كيف تم إبتكار RPWI ؟

بفضل الله عزل وجل يمكن ان يقوم الانسان بعمل اى شىء - فقط كل ما عليك ان تفكر - تتامل - تدعو الله - ولا تخجل ان فشلت - ان هذه التكنولوجيا قبل ان تظهر الى النور مرت بمراحل قاسية وازمات عصيبة.

بدات الفكرة بالتفكير فى البرمجة بدون كود - حينما عاد اخى المهندس محمد فايد (مبرمج كمبيوتر محترف متخصص فى VB6 و VB.NET واخيرا Java) من مؤتمر حضره لشركة مايكروسوفت فى القاهرة - مصر - تحدث فيه الشركة عن فيجوال استيديو

٢٠٠٥ Visual Studio 2005 وفى الواقع لم احضر ذلك المؤتمر لسبب بسيط ان المبلغ الازم لحضور المؤتمر رغم صغره حوال مئة جنيه - الا انه لم يكن معى فى ذلك الوقت (اظن لاننا كنا اخر الشهر) وانا فى الواقع لم اصرف من مدخراتى السابقة لانى لا اقبل ان تكون المصروفات اكثر من الايرادات مهما كان السبب.

تحدث اخى كثيرا عن المؤتمر والذى استمر ثلاثة ايام - وفى الواقع قال جمل - لن انساها مدى حياتى - قال **(لقد سمعت الحضور يقولون انه سوف ياتى يوم - مايكروسوفت هتخلينا نبرمج من غير كود)**

وكان اخى الغالى يقول ذلك ناقلا لى تعبير الحضور عن مدى دهشتهم - ولم يلحظ مدى تاثير تلك الجملة على نفسى - فقد سهرت طوال الليل افكر - ويدور فى راسى

هل ممكن ان يتحقق ذلك ؟
ياترى - كيف تكون البرمجة بدون كود ؟

وبالتاكيد اسئلة بدون اجابات - وانا عندما اجد نفسى فى موقف كهذا - فانى اكشر عن انيابى - ولا يهدأ لى بال - حتى اعمل كل ما بوسعى للوصول الى نتيجة - اما بالاجابة على اسئلة او الاستخدام بطريق مسدود ولكن بعد اكتساب شرف المحاولة (او المحاولات ان كان هناك امل).

انه البحث والتامل والتفكير والعمل والدعاء - خمسة وسائل لا غنى عنها لتحقيق اى نتيجة مرضية - وجميع هذه الوسائل تعمل على التوازى بلا انقطاع.

من خلال البحث توصلت الى التالى

حكمة كبيرة تعلمتها من العالم Alan Kay ان خير وسيلة للتنبأ بالمستقبل هى ان تقوم باختراعه

○ **استفدت من تلك الحكمة فى الاجابة على سؤالى الثانى - كيف تكون البرمجة بدون كود ؟ - طالما انها مستقبل وانا اسأل عنه - فان خير وسيلة للاجابة ان اخترع انا هذا المستقبل - وهذا ما حاولت جاهدا عمله.**

✚ بحثت عن تطور علم البرمجة من الداخل - فوجدت انه يرتكز على تطور نمط البرمجة - وكان ذلك دافعا للبحث فى هذا الموضوع - ومن خلال ذلك البحث تعرفت على نمط برمجة العميل الموجه AOP ونمط برمجة اللغات الموجه LOP وقمت بابتكار نمط البرمجة الخادم الممتاز DoubleS لكى يكون خير دليل بالنسبة لى لإتقانى لما تعلمته

✚ بحثت عن تطور علم البرمجة من الخارج - فوجدت بحثا لاحد العلماء صدر عام ٢٠٠٢ يتنبأ فيه بالبراهين ان مستقبل البرمجة سوف يكون مئة بالمئة بدون كود **وكان ذلك اجابة على سؤالى الاول هل ممكن ان يتحقق ذلك؟.**

✚ وجدت بحث عن رسالة دكتوراه لاحد الأساتذة فى اليابان كانت فكرته فصل البيانات عن التعليمات ومن خلال فهمى للمعنى توصلت لافكار فى غاية الروعة - ولكنى اكتشفت من خلال مراسلتى لذلك الدكتور - والذى اهتم بعملى فى تكنولوجيا البرمجة بدون كود - وكان حريصا على متابعتى - ان مافهمته فى البداية كان فى منتهى الخطأ - ولكنى فزت بافكارى التى وردت الى وان كان عن طريق الخطأ فى الفهم.

من خلال التأمل فى التكنولوجيا المعاصرة

✚ رسمت صورة شاملة لما وصلت اليه البرمجة بدون كود وتخيلت ماالمدى الذى يمكن ان تصل اليه - وبالفعل ايقنت انها يمكن فى يوم من الايام ان تحل محل البرمجة بواسطة الكود بصورة كاملة

✚ استوعبت مدى قوة وتأثير المعالجات والمصمات واكتشفت ثغرة فى منتهى الخطورة ان الرابط الوحيد بين هذه الادوات انها - يمكن ان تستخدم فى نفس التطبيق - ولكن هذه الادوات لا تعمل معا - انها كائنات منعزلة لاتستند على بعضها البعض - نعم هناك تعاون فى تطوير التطبيق - ولكنها فيما بينها لا يوجد تشابك يربط هذه الادوات معا - حيث فى الواقع انها ادوات لا تاتر على بعضها البعض - اى انها ليست جسد واحد وان تم

تجميعهما معا فى بيئة تطوير متكاملة واحدة IDE وهذا
فى خلاف الانسان الذى تتشابك اعضاءه - فلا يعيش
انسان بلا قلب (من الناحية العضوية - فانا اعلم انا هناك
اناس بلا قلوب) او بلا عقل.
توصلت الى ان محيط التطوير Framework هو خطوة نحو
التشابك بين ادوات التطوير - وذلك احد مميزاته التى
ينفرد بها عن بيئة التطوير المتكاملة IDE

من خلال التفكير والعمل والدعاء توصلت للاتى

ان هدفى هو البرمجة بدون كود - هدف قابل للتحقيق - ولكى
يتحقق انا بحاجة الى محيط تطوير Framework متخصص فى
هذا الامر - لان الهدف مجسم ويحاجة الى ادوات متشابكة ليس
فقط تتعاون معا وانما تركز على بعضها البعض.

وبالفعل كتب للفكرة ان توجد على الورق وان تكون منطقية
وسهلة التطبيق (بالنسبة لى على الاقل) وسهلة الاستخدام
(من قبل كل مبرمج) وان كانت صعبة التعلم (حسب راي من
اطلع عليها - وفى الواقع تلك كانت صدمة - ومن هنا تعلمت ان
مخترع اى فكرة ليس من السهل ان يدرك مدى تعقيدها الا اذا
عرضها على الاخرين).

الأفكار خلف إبتكار RPWI ؟

- فكرة فصل المعطيات X من الدالة $F(X)$
- فكرة حصر الاحتمالات الكثيرة
- فكرة تطور واجهة المستخدم
- فكرة مستويات ومتطلبات البرمجة بدون كود
- فكرة التفكير المنطقي فى حل المشاكل
- فكرة عناصر الخطوة
- فكرة مصمم الاهداف
- فكرة التفاعل لتحديد المعنى
- فكرة شجرة المجال
- فكرة التطبيق المحلل للبيانات
- فكرة الخادم
- فكرة الناقل
- فكرة مصمم الصفحات التفاعلية
- فكرة مصمم الناقل
- فكرة مستخرج الكود
- فكرة الحقيبة

فكرة فصل المعطيات X من الدالة $F(X)$

احد اهم افكار البرمجة بدون كود - مثلا تخيل الكود الوهمى
Pseudo Code التالى

```
Function test
  X = 1
  Print x on screen
  For t = 1 to 10 step 1
    Print expression (t+x-1) on screen
  End of for loop
End of function
```

ان هذا الكود الوهمى يعرف دالة تسمى test - هذه الدالة تعلن متغير يسمى X وتسنده له القيمة واحد (اي انه متغير رقمى) ثم تقوم بطباعة هذا المتغير على الشاشة - ثم تقوم بعمل حلقة تعد الارقام من 1 الى 10 بشرط ان قيمة X تم اعدادها لتكون واحد وبشرط ان تظل قيمة X تساوى واحد اثناء عمل الحلقة (شرطين غير ضروريين - يمكن ببساطة تعديل الكود الوهمى للاستغناء عنهما - وتلك ليست القضية - هذا مجرد مثال).

عند تطبيق هذا الكود الوهمى باى لغة (وقد سبق وانا وضحنا باكثر من مثال ان الامثلة البسيطة لاتفرق بين لغات البرمجة) فان الاختلاف سوف يكون فقط تبعا لاختلاف اوامر اللغة - ولكن الهدف واحد - والتسلسل المنطقى فى التفكير واحد - وايضا البيانات X قد تكون واحدة - اى ان الذى يختلف من لغة للغة هو التوضيح من خلال الوظيفة F() وربطها مع البيانات X لكى نكون الحلقة F(X) والتى لها ثلاث ابعاد - هما البيانات X والدالة F() والربط بينهما.

ان البيانات X فى المثال السابق هى

```
Test
X
1
T
1
10
1
T+X-1
```

والدوال او العمليات هى

```
Function
=
print
for
end of loop
end of function
```

ولا ينبغى الخلط بين استخدام كلمة البيانات X وبين المعطيات للدالة فى البرمجة - ولا ينبغى الخلط بين العمليات F() وبين

الدوال فى البرمجة - اننا نناقش مفهوم اشمل يتعلق بالكود بصفة عامة بعيدا عن تفاصيل قواعد علم البرمجة.

ان البرمجة بدون كود تستفيد من تلك الفكرة - بانها تطلب من المبرمج تحديد البيانات X فقط - ولا تطلب منه ان يقدم البيانات مرتبطة بالوظيفة التى تتم عليها.
اى انها تطلب X بدلا من $F(X)$.

نلاحظ ان البرمجة بواسطة الكود تشمل تكرار للبيانات - مثل تكرار الرقم واحد اكثر من مرة - لكن فى البرمجة بدون كود لا تكون الحاجة الى تكرار احد البيانات ضرورى.

فكرة حصر الاحتمالات الكثيرة

بعد فصل البيانات عن العمليات التى تتم عليها - تظهر لنا مشكلة كبيرة كيف يمكن للحاسب تحديد العملية - اى انها ليست مشكلة كبيرة - فى الواقع انها كارثة - ان الكمبيوتر لا يفهم بالاشارة (هى لغة ليست سهلة حتى على الانسان)

هل تتوقع ما حجم هذه الاحتمالات ؟
ما حجم العمليات التى يمكن ان يتم عملها على البيانات؟

لحسن الحظ ان هذه الاحتمالات التى يظن البعض انها لا نهائية وهى فعلا كذلك - لكن يمكن حصرها - كما سنرى - ان العمليات وعددها يتوقف على لغة البرمجة تحت الاستخدام - اذا تم ربط عدد العمليات بامكانيات لغة البرمجة فاننا نصل لعدد محدود - فى لغة قديمة مثل للغة السى والتى لدى اصدار منها يضم ٥٥٠ دالة فقط بالاضافة الى قواعد اللغة - فاننا نصل الى احتمالات عددها بسيط اقل من ١٠٠٠ احتمال

لكن فى لغات متطورة مثل لغات الدوت نت والتى تضم الالاف من الفصائل - بها العشرات والمئات من العمليات Methods فاننا نقف امام الملايين من الاحتمالات
ولكن سوف نصل لعدد محدد - اى تم الحصر بناء على لغة البرمجة المستخدمة فى الخلفية.

فكرة تطور واجهة المستخدم

ان تطور واجهة المستخدم من الواجهة الخطية Command Line المعتمدة على CUI الى Character User Interface الى الواجهة الرسومية GUI = Graphical User Interface ملفت للنظر

لقد كان استخدام نظام دوس DOS القديم يعتمد على عدد معين من الاوامر ذات القواعد المحددة فى الكتابة - ثم جاء نظام النوافذ Windows ليقدم البديل الاسهل من خلال واجهة المستخدم

انه حدث خطير - واذا تمعنا ونظرنا بعمق - ان قواعد كتابة اوامر دوس الى حد ما تشبه قواعد كتابة الكود (مع الفارق الكبير طبعا) ومع ذلك من خلال واجهة المستخدم وجد البديل

ومن هنا ان تطوير التطبيقات من خلال كتابة الكود - يعنى المواجهة الخطية - لذا فان البرمجة بدون كود تعلمت الكثير من هذا الدرس - وتقدم المواجهة الرسومية.

فكرة مستويات ومتطلبات البرمجة بدون كود

ان البرمجة بواسطة الكود - كانت عدة مستويات - اهمها المستوى المنخفض Low Level Programming حيث يصل المبرمج الى التعامل مع العتاد Hardware بصورة مباشرة عن طريق التعامل المباشر مع المسجلات Registers الخاصة بالمعالج CPU مع امكانية عمل مقاطعة Interrupt للمعالج فى اى وقت - بالاضافة الى التعامل المباشرة من الذاكرة العشوائية Memory حيث يمكن الكتابة او القراءة فى اى عنوان متاح التعامل معه - ويمكن التعامل مع وحدات الادخال والاخراج من خلال البوابات المخصصة لذلك I/O Ports

وهناك برمجة المستوى الرفيع High Level Programming التى تتسم باستخدام كلمات اشبه للغة الانجليزية - وبذلك تكون عملية كتابة وصيانة البرامج اسهل بكثير - كما انها تعلق قليلا الى فكر الانسان من خلال نمط البرمجة - مثل البرمجة الهيكلية

وبرمجة الكائنات - وتقدم سبل افضل لادارة الذاكرة من خلال هياكل البيانات Data Structure وملامح افضل للتحكم بالمعالج من خلال تركيبات التحكم Control Structure .

كذلك ان البرمجة بدون كود بها مستويات - وهى
Low Level RPWI - ١

حيث تكون لدينا واجهة رسومية بلغة يفهمها الانسان (عربى - انجلىزى) عبارة عن بديل مباشر للغة برمجة معينة - واستخدام البرمجة بدون كود حينها يشترط الالمام بقواعد البرمجة وبالاخص قواعد اللغة تحت الاستخدام - حيث ان العزل عن الكود يكون غير كامل.

High Level RPWI - ٢

هنا تعلقو الواجهة الرسومية المصممة بلغة يفهمها الانسان وتعتمد على لغة برمجة فى الخلفية بحيث تكون ماتوفره اللغة من خيارات يعزل مستخدمها عن مفاهيم البرمجة - ولا تشترط الالمام بقواعد لغة البرمجة.

فكرة التفكير المنطقى فى حل المشاكل

ان حل اى مشكلة يتم من خلال عدة خطوات - وقد تشكل كل مجموعة من الخطوات خطوة عامة او هدف محدد - كذلك فان البرمجة بدون كود - وحدة البناء الاساسية بها هى الهدف Goal ويكون كل هدف مقسم الى مجموعة من الخطوات Steps حيث انه من الممكن ايضا ان تشمل اى خطوة على خطوات فرعية - ولا ينفى ذلك كون هذه الخطوة - هى ايضا خطوة - فمثلا اذا كانت الخطوة A تشتمل على خطوتين B و C فان ذلك لا ينفى كون A هى خطوة ايضا - ومن ثم يكون لدينا ثلاث خطوات هما A و B و C

وحيث ان الهدف Goal يتفرع منه عدة خطوات - فان كل هدف يشمل مايسمى بنقطة البداية Start Point وهى ليست عبارة عن خطوة - كما ان خطوات الهدف يتم تمثيلها من خلال شجرة تسمى Goal Tree

ان لكل خطوة اسم - لدى المبرمج الحرية الكاملة فى تحديده باى لغة يريدتها (انجلىزى - عربى ...إلخ)

فكرة عناصر الخطوة

بما ان الحرية لدى المبرمج فى تحديد اسم الخطوة - فمن المؤكد انه لابد من وسيلة - لتحديد معنى الخطوة بالنسبة للحاسب - يتم ذلك من خلال عناصر الخطوة - وهم خمسة عناصر

1 - Data

○ هى X منفصلة عن العملية التى تتم عليها ولدى المبرمج الحرية فى تحديدها

2 - Analysis

○ هى عملية تتم على X لتهيئتها قبل الحصول على معلومات منها - وليس للمبرمج علاقة بذلك

3 - Information

○ هى معلومات يتم الحصول عليها مستندة على التحليل وعلى امور اخرى - وليس للمبرمج علاقة بها

4 - Result

○ هو الكود المباشر الذى تفهمه لغة البرمجة وليس على المبرمج كتابته - ولكن يمكن الاطلاع عليه

5 - Info

○ هى معلومات يكتبها المبرمج كملاحظات - اى comments

نلاحظ وجود تحليل - وذلك يتم من خلال تطبيق معين يسمى Analysis Applications - كما يوجد معلومات - وهى يتم استنتاجها اما من خلال الخادم Server كنتيجة لعمليات تمت على البيانات بعد تحليلها - او كنتيجة لاستقبال بيانات من المبرمج بصورة مباشرة - اما بخصوص النتيجة Result فهى مهمة الخادم Server - كما انه يوجد مكون اخر على النظر مع الخادم يسمى الناقل Transporter

فكرة مصمم الاهداف Goal Designer

حيث ان البرمجة بدون كود تتركز على الهدف Goal بدلا من الكود - فان هناك مصمم متخصص فى ذلك مهمته تصميم وصيانة

الاهداف - وبالتالي هذا المصمم - هو البديل الاساسى لمحرف الكود Code Editor مع الفارق الكبير.

فكرة التفاعل لتحديد المعنى

الخطوة فى حد ذاتها اذا لم نحدد معناها للحاسب - فانه ليس لها معنى - ويتم تحديد ذلك من خلال التفاعل Interaction مع الحاسب - ومعنى التفاعل الحوار المتبادل - ويتم على ٢ مراحل

١ - نقصر المسافة على الحاسب بان نختار مايسمى بالمجال Domain حيث يشمل هذا المجال مجموعة من الاحتمالات التى بها مانريد ان يحدث من خلال الخطوة.

٢ - يشتمل كل مجال على مجموعة من المكونات Components يجب ان نختار فيما بينها حيث تكون هذه المكونات هى المسئولة عند تحديد العملية التى نريدها - وتنقسم المكونات الى نوعين - النوع الاول يسمى خادم Server والاخر يسمى ناقل Transporter

٣ - اذا كان المكون عبارة عن خادم Server فانه يحصل بصورة تلقائية من قبل مصمم الاهداف على نسخة من البيانات Data اى X الخاصة بالخطوة بعد تحليلها من التطبيق المخصص لذلك - حيث ان الخادم Server عبارة عن تطبيق مستقل - عندما يعمل فان نتيجة عمله هى توليد شغيرة مصدرية بناء على البيانات X بعد تحليلها - وبصورة تلقائية يتم الحصول على التحليل الخاص بالبيانات ووضعه فى المكان المخصص له التابع للخطوة - وكذلك المعلومات التى توصل لها الخادم - وكذلك الكود الذى صدر منه - وبما ان الخادم عبارة عن تطبيق مستقل فمن الممكن ان يصمم بصفة مستقلة بحيث يكون له ملامح مميزة مثل دعم اكثر من لغة مواجهة يفهمها الانسان (عربى - انجليزى..الخ) ودعم اكثر من لغة برمجة يولد الكود لها - وكذلك من الممكن ان يوجه الخادم اسئلة للمستخدم - وتكون اجابات المستخدم عبارة عن معلومات يتم وضعها فى المكان المناسب تبعا للخطوة.

٤ - اذا كان المكون عبارة عن ناقل Transporter فانه شبيه بالمعالج Wizard حيث يقدم مجموعة من الصفحات والاختيارات ثم يولد الكود - ويختلف عن الخادم فى انه ليس تطبيق مستقل

وانما تطبيق بسيط يتم عمله بسهولة من خلال ادوات تصميم مخصصة كما انه يختلف عن المعالج فى ان الاستخدام الاساسى له هو عمل واجهة للغات البرمجة لذلك فهو مناسب لـ Low Level RPWI وسمى ناقل Transporter لانه ينقل اللغة من الكود Code الى RPWI.

فكرة شجرة المجال

سبق وان ذكرنا انه قبل تحديد المكون Component الذى سوف يستقبل بيانات الخطوة - فانه يتم تحديد المجال Domain اولا - وشجرة المجال Domain Tree ماهى الى شجرة تشمل جميع المجالات او التصنيفات للمكونات بحيث يمكن اضافة مجال الى الشجرة - او حذف مجال منها - او تعديل اسمه وهكذا كما يحتاج لنا اضافة او حذف او تعديل اى مكون Component داخل المجال.

فكرة التطبيق المحلل للبيانات

البيانات قبل وصولها للمكون Component سواء كان خادم Server او ناقل Transporter فانه يتم تحليلها - ويمكن احيانا الاستغناء عن هذه العملية حيث تكون البيانات بعد تحليلها هى نفسها قبل تحليلها - ومن امثلة التحليل - وضع كل كلمة فى سطر مستقل - او كل حرف فى سطر مستقل وهكذا.

وحيث ان المكونات Components قد تتشارك فى انها تتعامل مع نمط محدد من البيانات المحللة فانه يمكن استخدام تطبيق موحد للتحليل وعندها يسمى التطبيق المحلل للبيانات.

فكرة الخادم Server

تعرضنا لها - هو عبارة عن تطبيق مستقل يستقبل بيانات من مصمم الاهداف بعد مرورها على التطبيق المحلل - وتكون مهمته توليد الشفيرة المصدرية ويمكن ان يتفاعل مع المستخدم - وميزة الخادم انه يمكن برمجته باى لغة برمجة - ويفضل

استخدام لغة البرمجة xHarbour مع نمط البرمجة DoubleS (Super Server) المتخصص فى برمجة الخادم.

فكرة الناقل

تعرضنا لها - هو عبارة عن تطبيق غير مستقل ولايحتاج للبرمجة بواسطة الكود - حيث يتم عمله وتطويره بدون كود من خلال ادوات التصميم - والناقل شبيه بالمعالج حيث يقدم شاشات يتفاعل بها مع المستخدم - تسمى هذه الشاشات الصفحات التفاعلية اى Interaction Pages ويتم تصميمها من خلال Interaction Designer اى مصمم التفاعل - كما يتم الربط بين هذه الصفحات وبين الكود الذى يتم توليده من خلال الناقل Transporter والذى يتم تصميمه من خلال Designer Transporter و على الرغم من ان الناقل يشمل الصفحات التفاعلية الا انها يتم تصميمها بصفة مستقلة لانه يمكن استخدام نفس الصفحة فى اكثر من ناقل.

فكرة مصمم الصفحات التفاعلية

استخدام هذا المصمم كما ذكرنا لتصميم صفحات الناقل - وهو عبارة عن مصمم Designer شبيه بمصمم النماذج Form Designer الا انه ابسط بكثير - كما ان الهدف مختلف تماما - حيث هنا يتم تصميم صفحات بسيط لاتتخطى عناصرها العناوين ومربعات النص ومربع الفحص وقائمة العناصر.

فكرة مصمم الناقل

هو عبارة عن مصمم يتم من خلال تصميم ناقل والذى يتحدد بمجموعة من الصفحات التفاعلية - بالاضافة الى الكود الذى يتم توليده - مع الربط بينهما - حيث تشمل الصفحة التفاعلية مجموعة من المتغيرات - تحمل القيم التى يحددها المستخدم من خلال مربع النص Text Box او مربع الفحص Check Box او القائمة ListBox - بينما يشمل الكود الذى يولده الناقل مجموعة

من المتغيرات - لذلك يتم الربط بين متغيرات الصفحات التفاعلية والمتغيرات الموجودة بالكود من خلال عملية تسمى Matching

ولعله الجدير بالذكر ان كتابة الكود الذى سوف يتم توليده فى الناقل يمكن ان تشمل كود وصفى Script للتحكم فى عملية توليد الكود - مثل استبعاد جزء من الكود - واستخراج جزء معين فى حالة معينة.

فكرة مستخرج الكود

بعد عملية التفاعل Interaction لتحديد الغرض من الخطوة - فانه يصبح لدينا شفرة مصدرية تابعة للخطوة - وهكذا مع بقية الخطوات - وعندها لايبقى لنا الا استخراج ملف يشتمل على الشفرة المصدرية للهدف كاملا - او لمجموعة من الاهداف معا - ويتم ذلك من خلال تطبيق مخصص يسمى مستخرج الكود اى Code Extractor حيث يستقبل من مصمم الاهداف كل من الخطوات والاكواد الخاصة بكل خطوة - ويقوم بتوليد الكود

فكرة الحقيقية

المبرمج بتكنولوجيا البرمجة بدون كود لديه الحق فى تصميم شجرة المجال Domain Tree الخاصة به - ويمكنه تصميم المكونات Components التى يشملها كل مجال على حده سواء كانت مجموعة من الخوادم Servers او مجموعة من الناقلين Transporters - ولكى يمكن عملية نقل هذا المجهود من جهاز لآخر او من شخص لآخر - تظهر الحقيقية - التى تجمع شجرة المجال بحيث يمكن نقلها وتحميلها على جهاز اخر.

التطبيق العملى

تكنولوجيا البرمجة بدون كود RPWI Technology توجد ضمنا ضمن محيط التطوير DoubleS Framework والذى يمكنك الحصول عليه من الموقع

<http://www.sourceforge.net/projects/doublesvsoop>

ويمكنك من خلال هذا الموقع الحصول على
DoubleS Framework for Windows
وهو عبارة عن ملفين مضغوطين
FAYED_SSF_STABLE_R5.ZIP - ١
يشتمل على محيط التطوير
FAYED_SSF1_R5_SP4.ZIP - ٢
يشتمل على اخر Service Pack

بخصوص الملف المشتمل على محيط التطوير فان محتوياته
كالتالى



قم بنسخ المجلدين CodeEX و SSRPWI الى المسار C:\
وهذا ضرورى حتى يعمل مستخرج الكود ونظام التفاعل

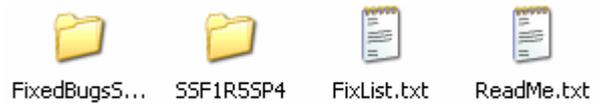
قم بفتح مجلد Framework



setup.exe

قم بتشغيل برنامج الاعداد Setup.exe واتبع التعليمات

والان بخصوص الملف المضغوط الاخر والذي يشمل اخر Service
Pack فان محتوياته كالتالى



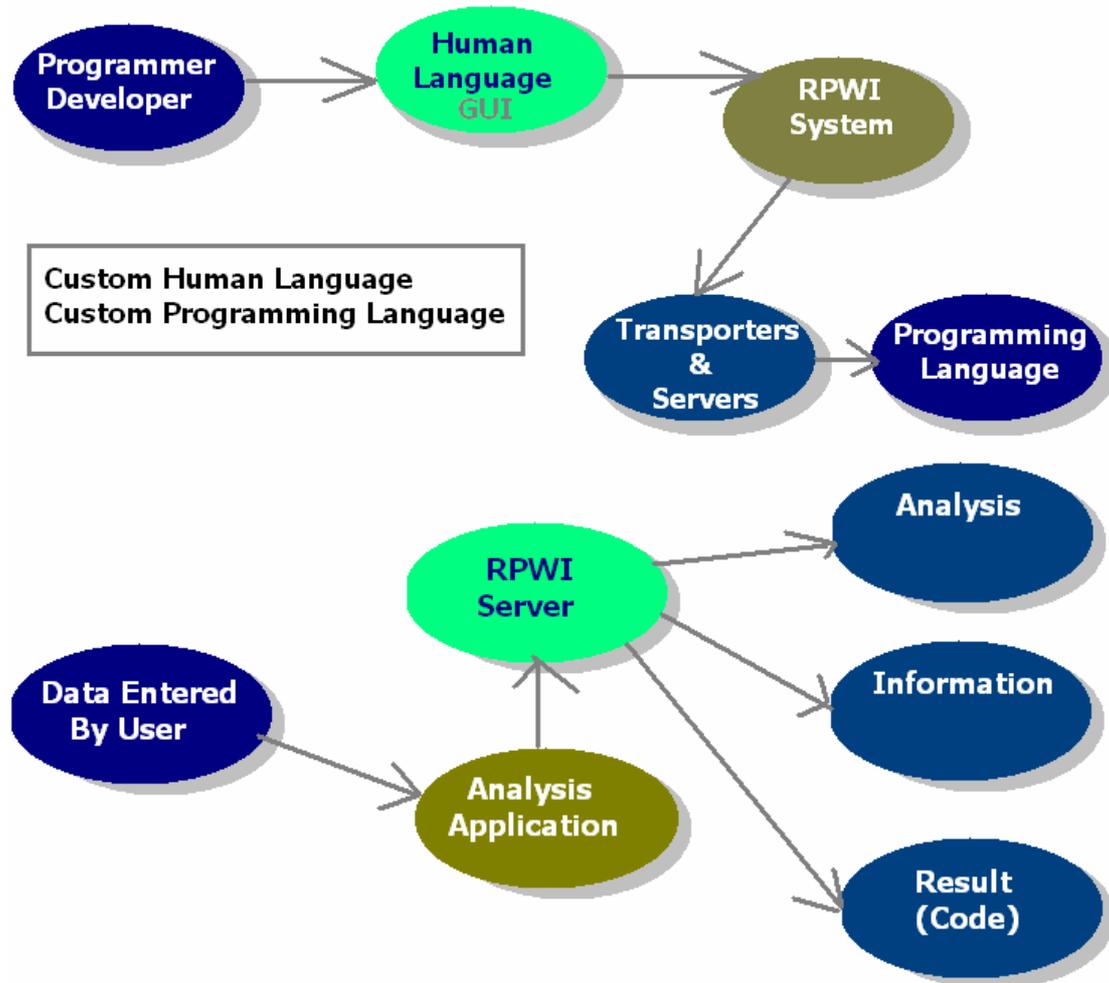
قم بفتح المجلد SSF1R5SP4



قم بنسخ هذه الملفات الى المسار الذى تم تحميل محيط
التطوير به - مع الموافقة على استبدال الملفات.

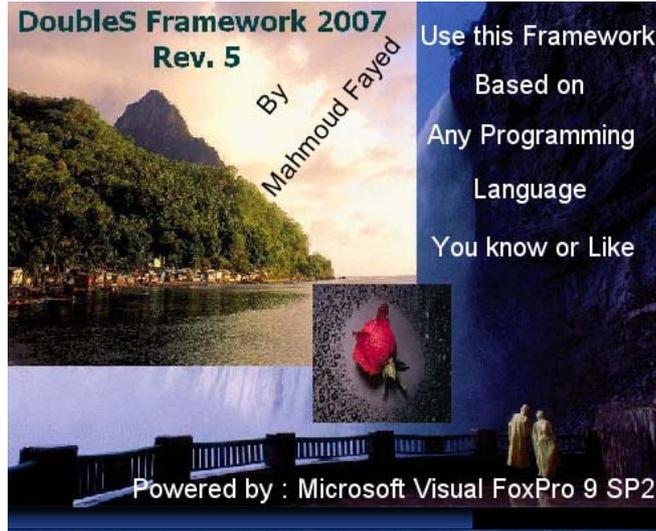
وهكذا نكون قد انتهينا من تحميل واعداد التكنولوجيا الخاصة بالبرمجة بدون كود.

والشكل التالي يبين التصور العام لكيفية عمل النظام



1- Run DoubleS Framework for Windows.

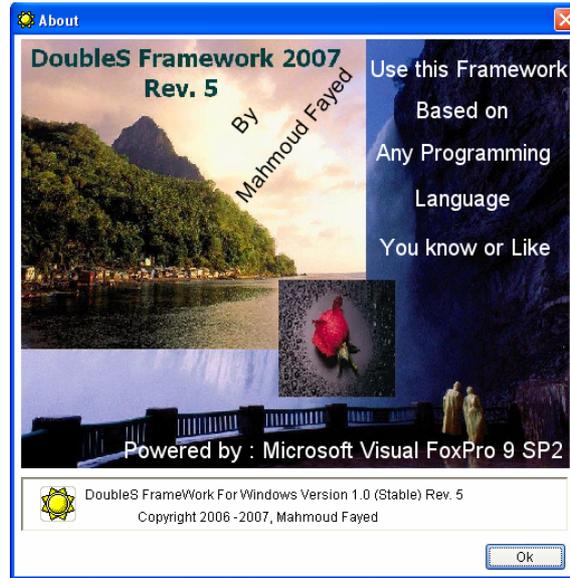
قم بتشغيل محيط التطوير – تحت ويندوز



شاشة البداية – المقدمة - الشعار Splash Screen

Here you should wait. ينبغي عليك الانتظار هنا.

2 – Welcome Screen, Click Ok. شاشة الترحيب – اضغط زر.

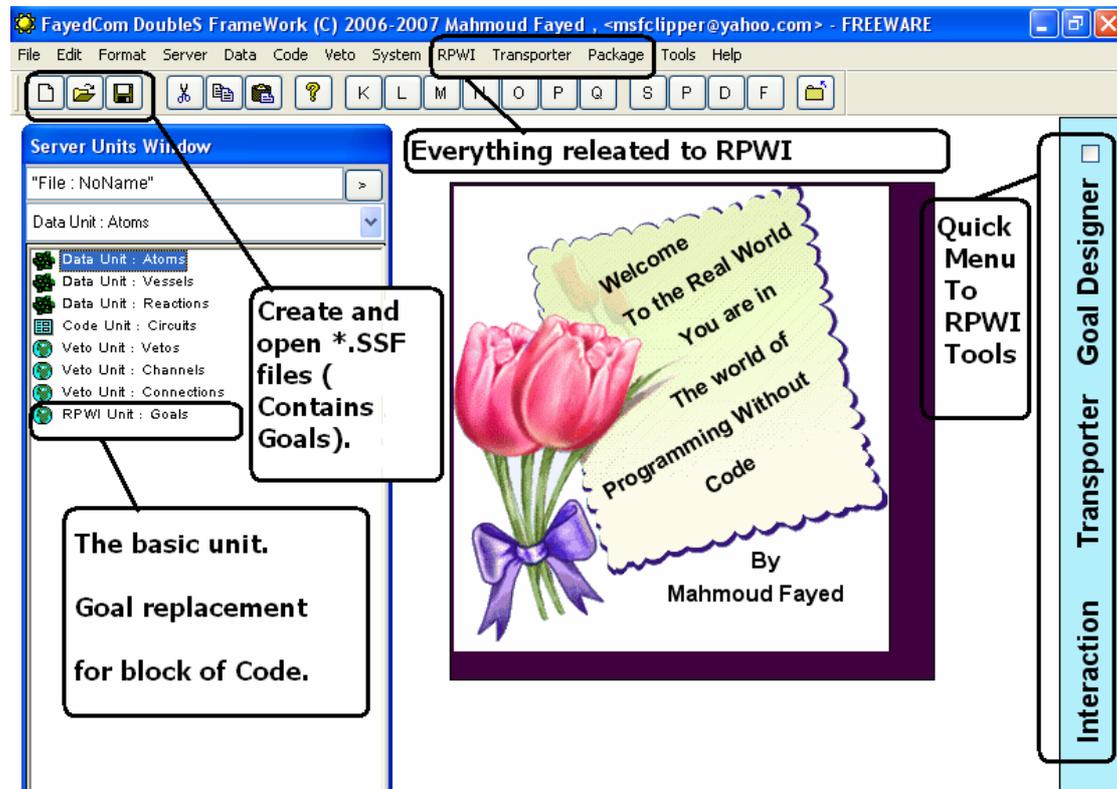


Now we have the framework, it's 2 projects in one, so not every thing is related to our purpose

الان اصبح لدينا محيط التطوير – انه مشروعين فى مشروع واحد – لذلك ليس كل شىء يتعلق بالعرض الذى نهتم له.

Every thing is important for RPWI is selected as in the next figure

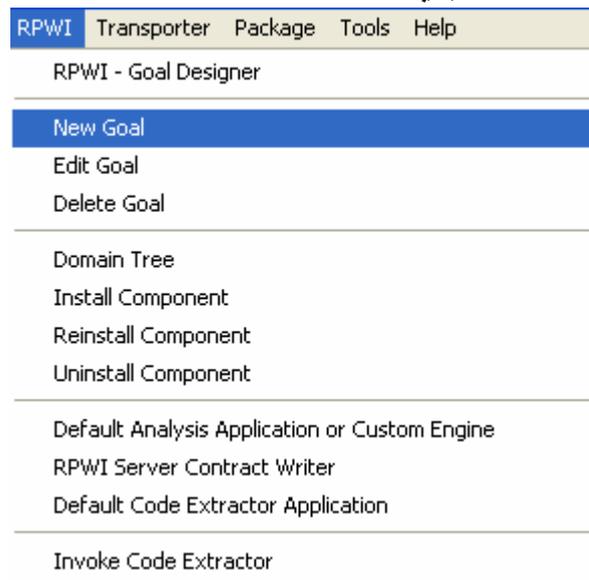
كل شيء يتعلق بتكنولوجيا البرمجة بدون كود تم تحديده كما في الشكل التالي



DoubleS Framework محيط تطوير الخادم الممتاز

Click (RPWI) menu, then chose (New Goal)

اختار قائمة RPWI ثم اختار New Goal لإنشاء هدف جديد

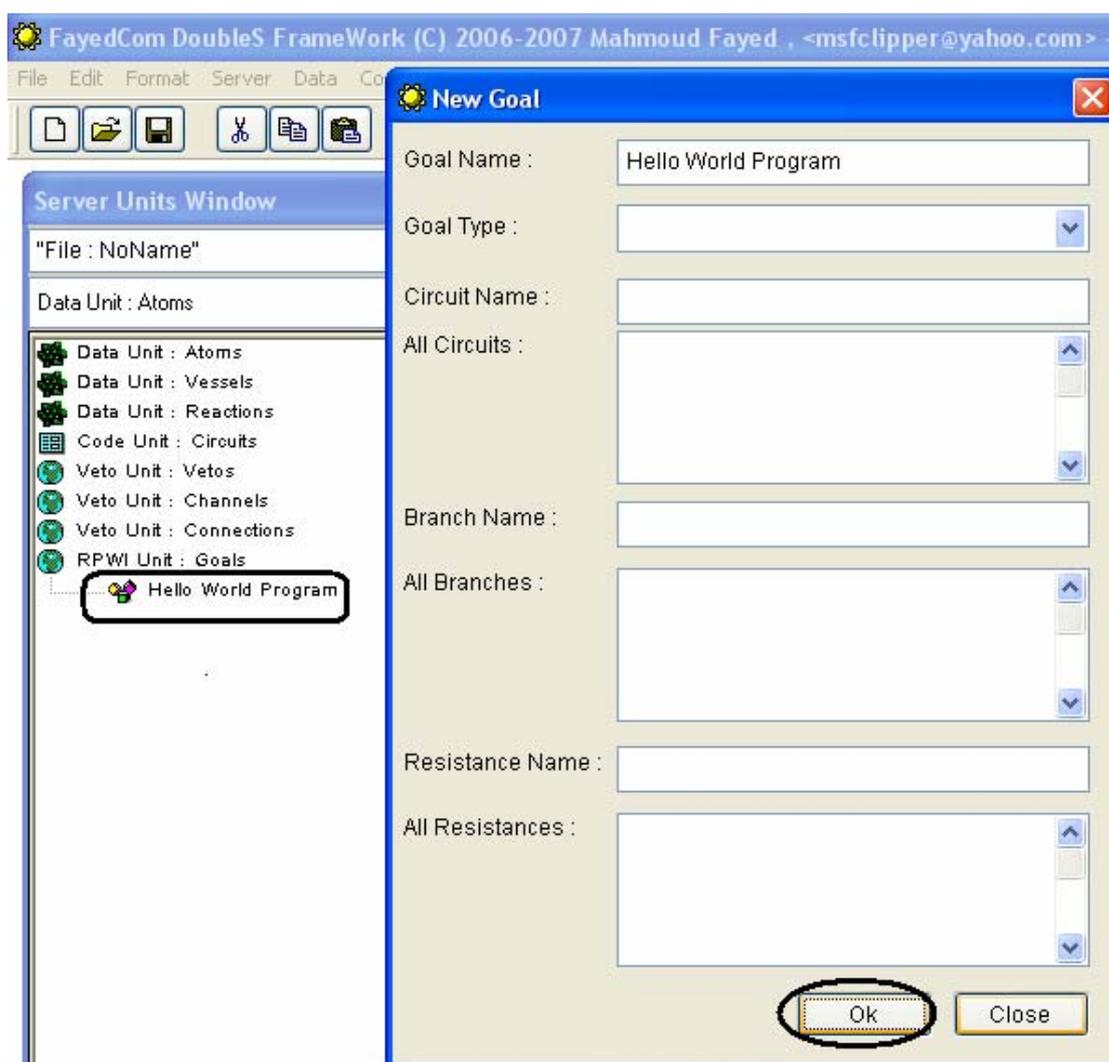


at goal name, write any name you like, for example "Hello World Program", then click (ok) button, you will notice that the goal is added to the (RPWI:UNIT – GOALS) in the tree of (Server Units Window).

عند اسم الهدف – اكتب اى اسم يناسبك وليكن مثلا Hello World Program ثم اضغط على الزر ok سوف تلاحظ ان الهدف تم إضافته الى وحدة RPWI – الاهداف فى شجرة الخادم – التابعة لنافذة وحدات الخادم

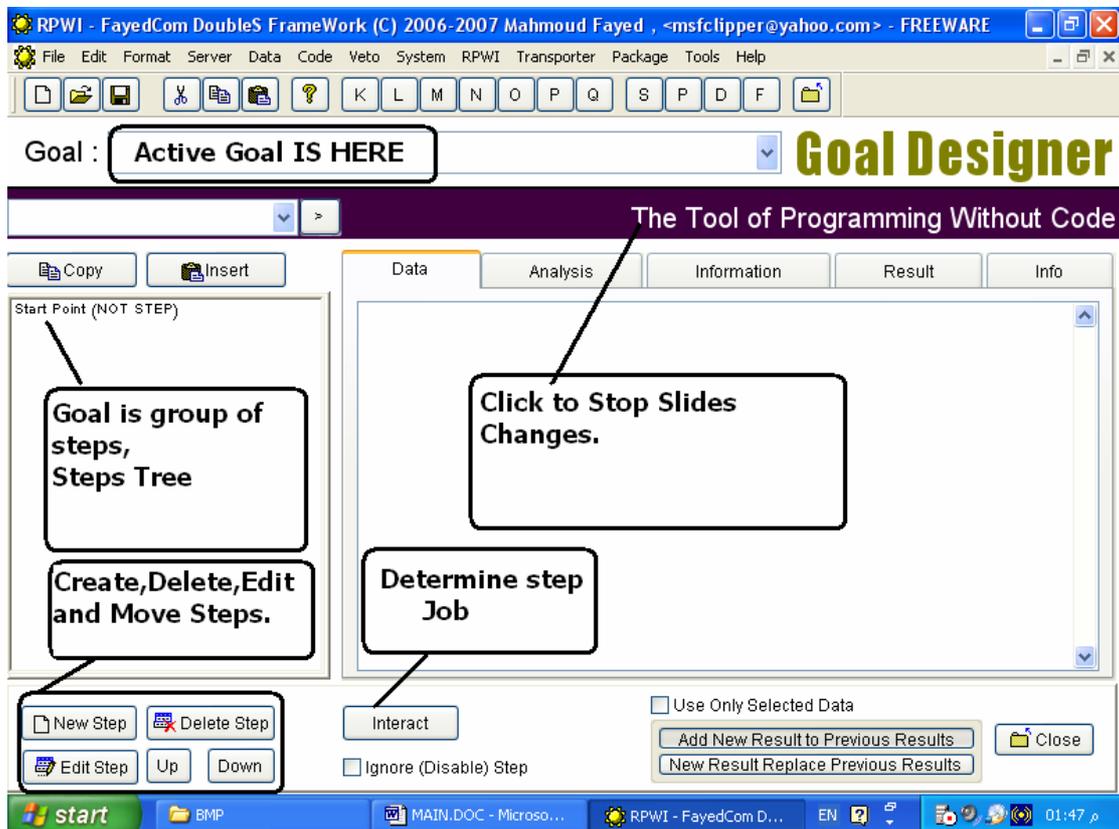
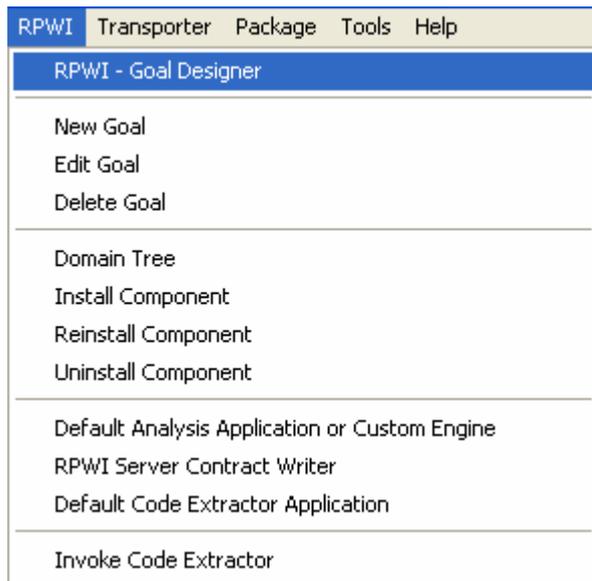
Ignore any other details (Goal type ,Circuit name,...etc) it's not important for us (it's important only for DoubleS Paradigm Users).

اهمل بقية التفاصيل الاخرى الخاصة بتعريف الهدف – انها ليست هامة بالنسبة لنا – انها مهمة فقط لمستخدمى نمط البرمجة الخادم الممتاز



Click Close اضغط على

Now open the goal designer الان افتح مصمم الاهداف



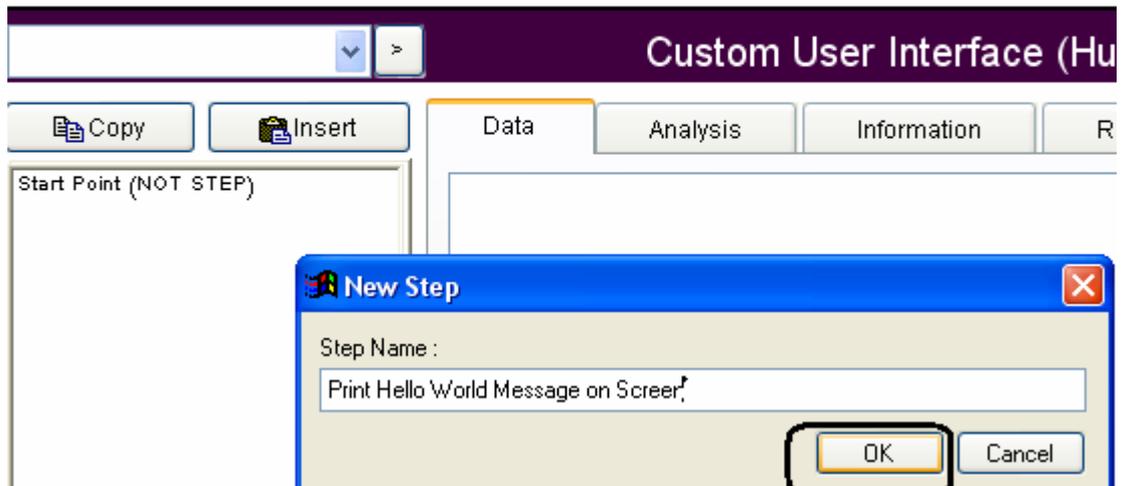
From steps tree, select "Start Pint (Not Step)"

Click "New Step" Button

من شجرة الاهداف قم بتحديد نقطة البداية (ليست خطوة) ثم اضغط على زر New Step

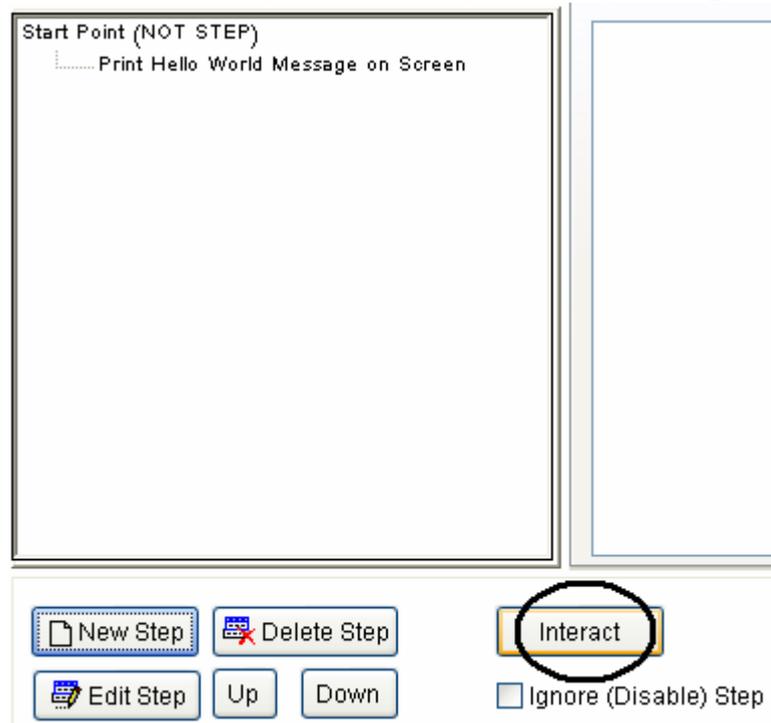
Write the step name, then click ok زر اضغط على زر ok

Goal : Hello World Program Goal D



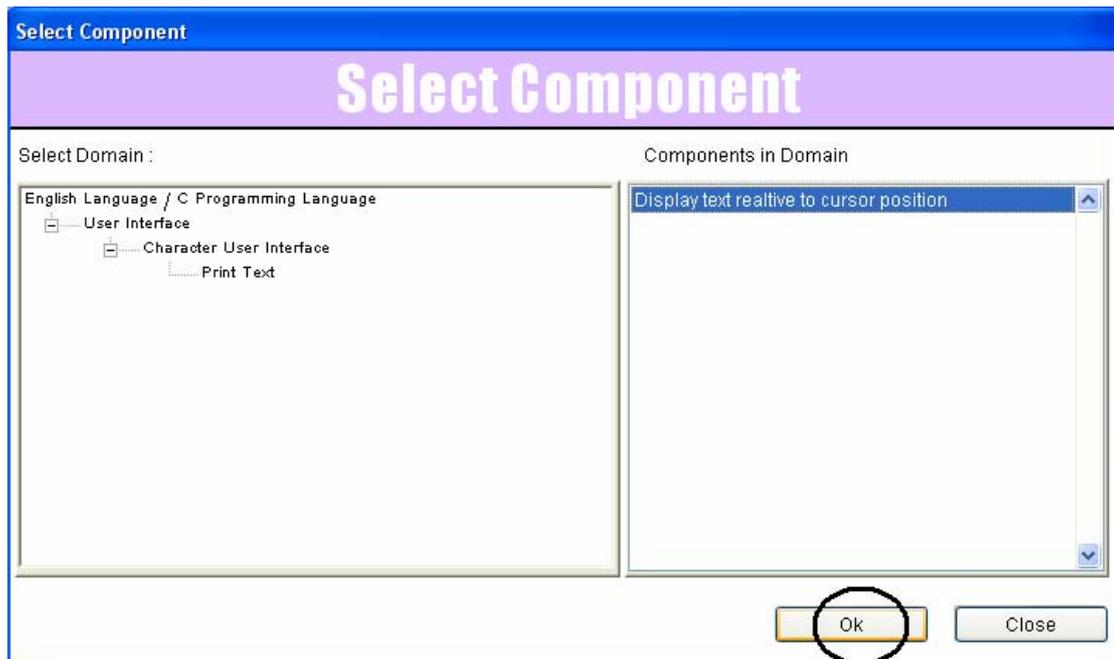
now click "Interact" to determine what is the job of this step.

الآن اضغط على زر Interact لكي تحدد الوظيفة المطلوبة من هذه الخطوة



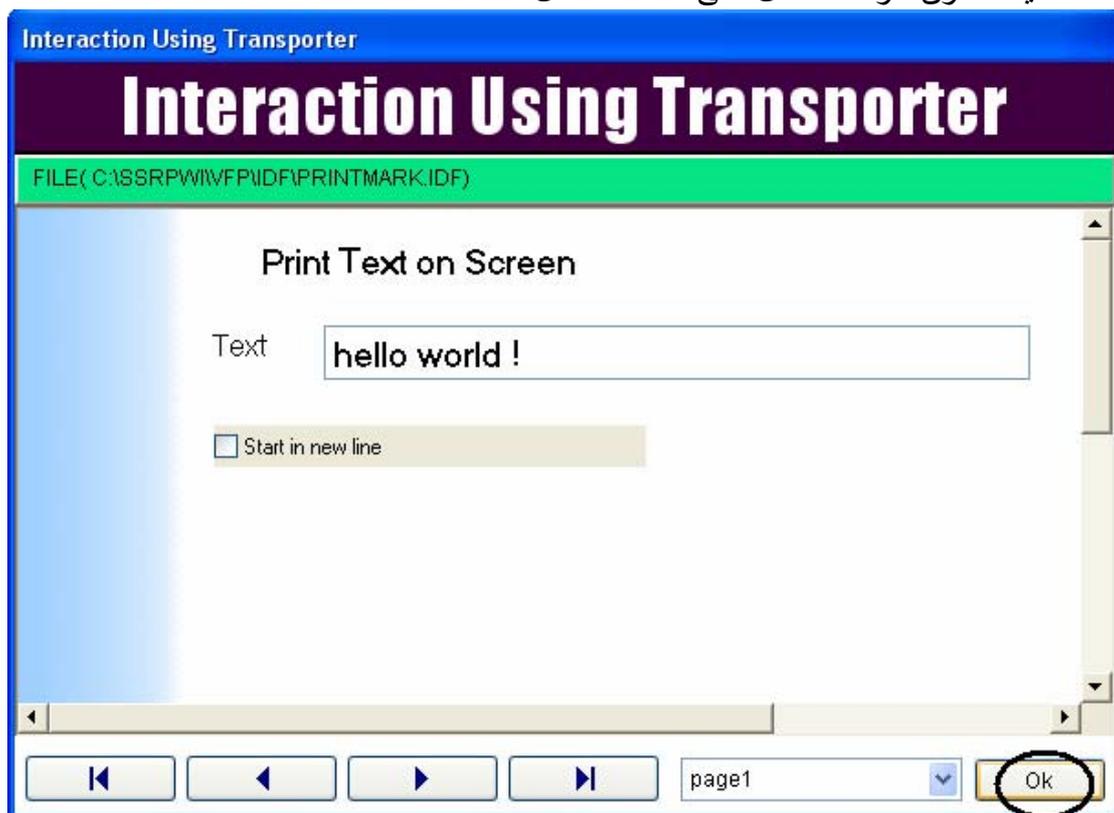
From Domain Tree, select the domain which is related to what you want to do, then select the component you want

من شجرة المجال حدد المجال الذي يتعلق بما تريد عمله ثم اختار المكون الذي تريده



after selecting the component you will have Interaction Window

بعد تحديد المكون سوف تحصل على شاشة التفاعل



use the interaction page to determine what you want exactly then click ok button to continue

استخدم صفحة التفاعل لكي تحدد ماذا تريد بالتحديد ثم اضغط زر ok لكي تستمر

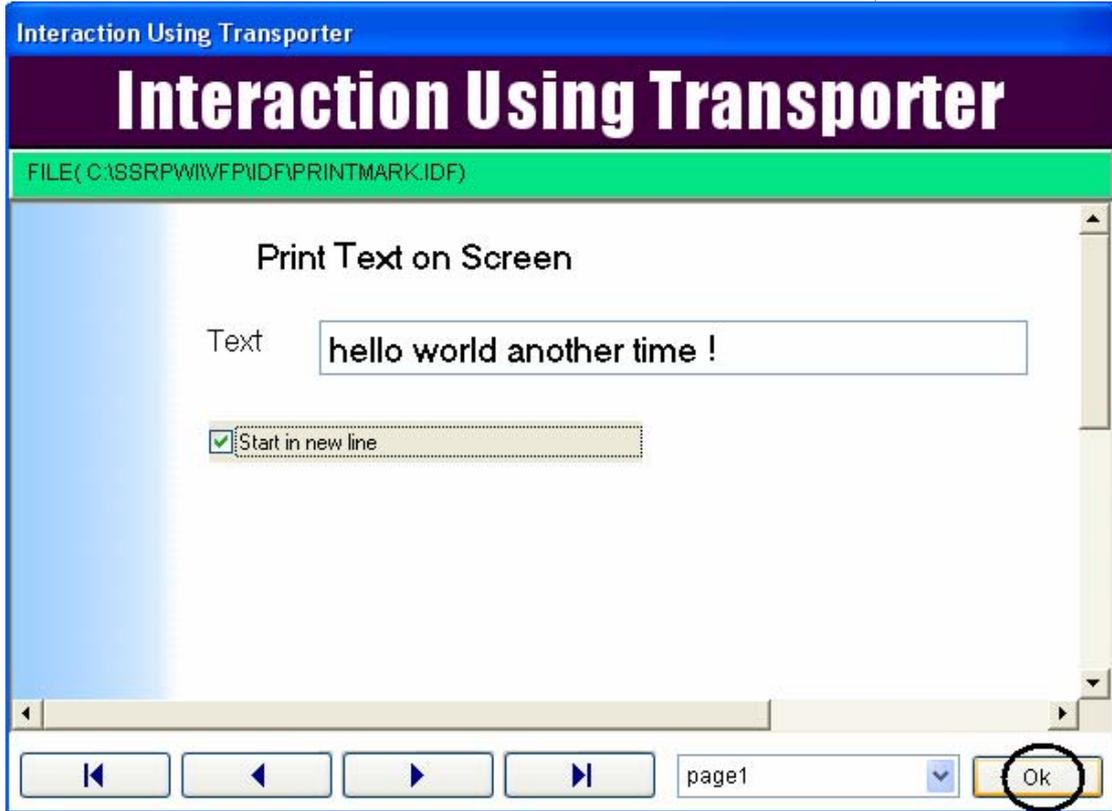
you will have message ask you if you want to repeat this process another time now , click ok

سوف تحصل على رسالة تسالك اذا كنت تريد تكرار هذه عملية مرة اخرى – اضغط Ok



write another message then click ok

اكتب رسالة اخرى ثم اضغط ok



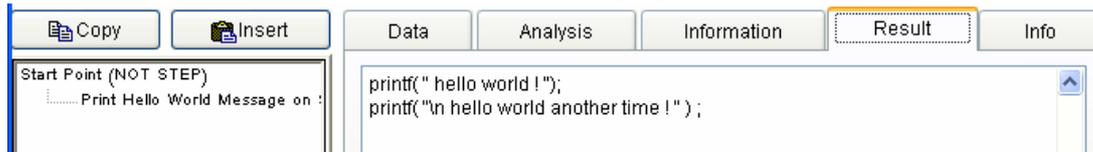
now when you have the message click no

الان عندما تحصل على الرسالة التي سبق التعرض لها اضغط زر no



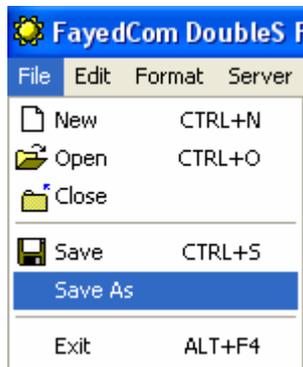
in goal designer, at result tab you can see the generated code.

في مصمم الهدف – عند الجزء الخاص بالنتيجة result يمكنك رؤية الشفرة المصدرية code التي تم توليدها



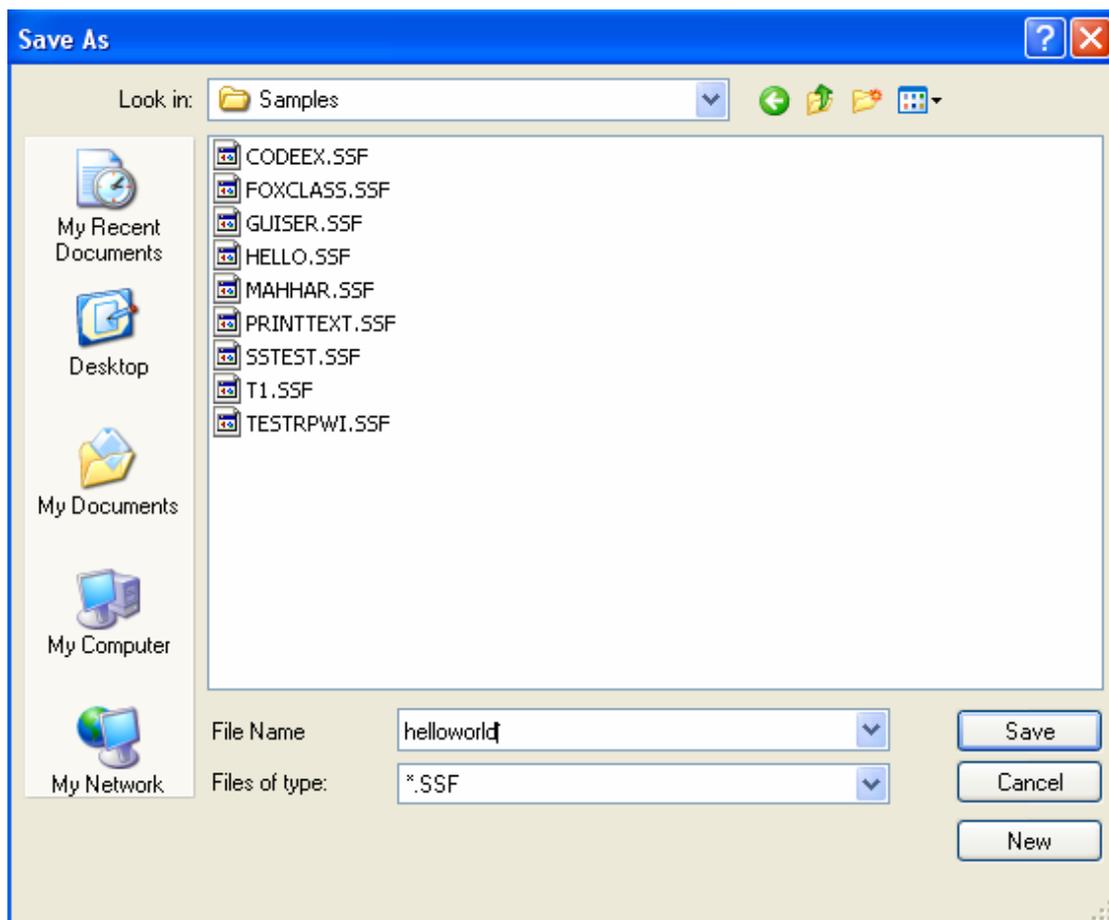
now it's the time to save all of our work in file.

الان حان الوقت لكي نحفظ كل عملنا في ملف خاص



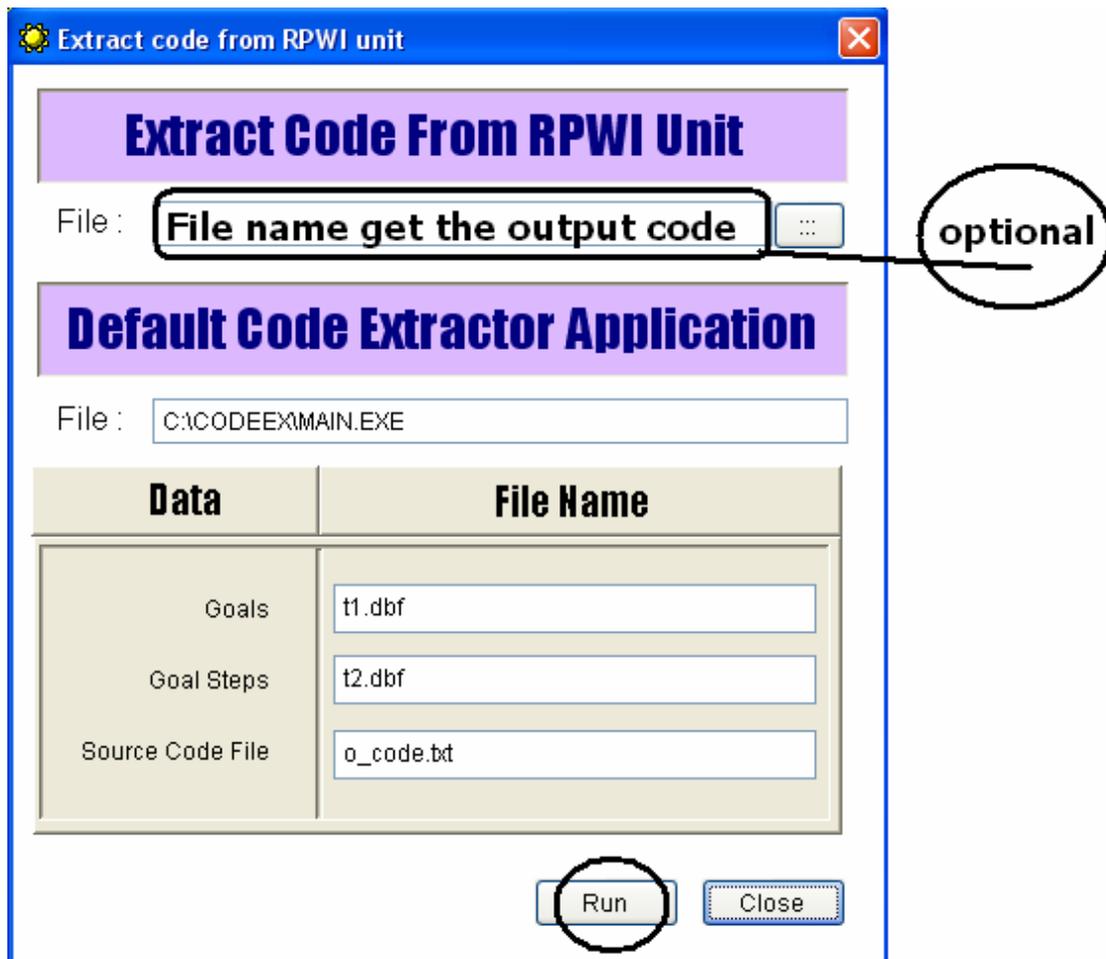
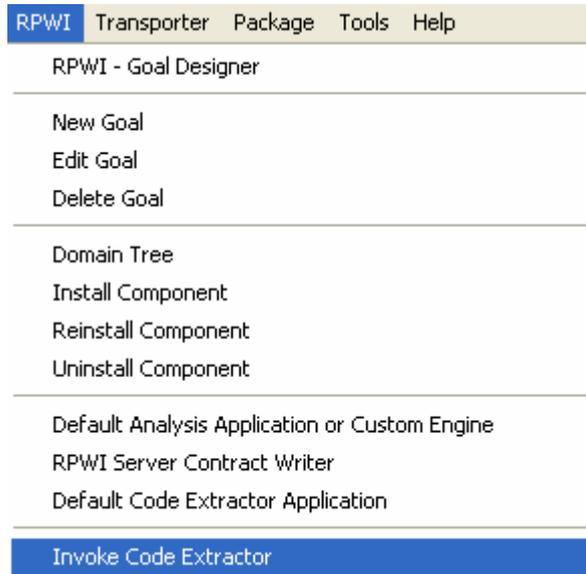
determine the file name *.SSF

حدد اسم الملف بحيث يأخذ الامتداد SSF



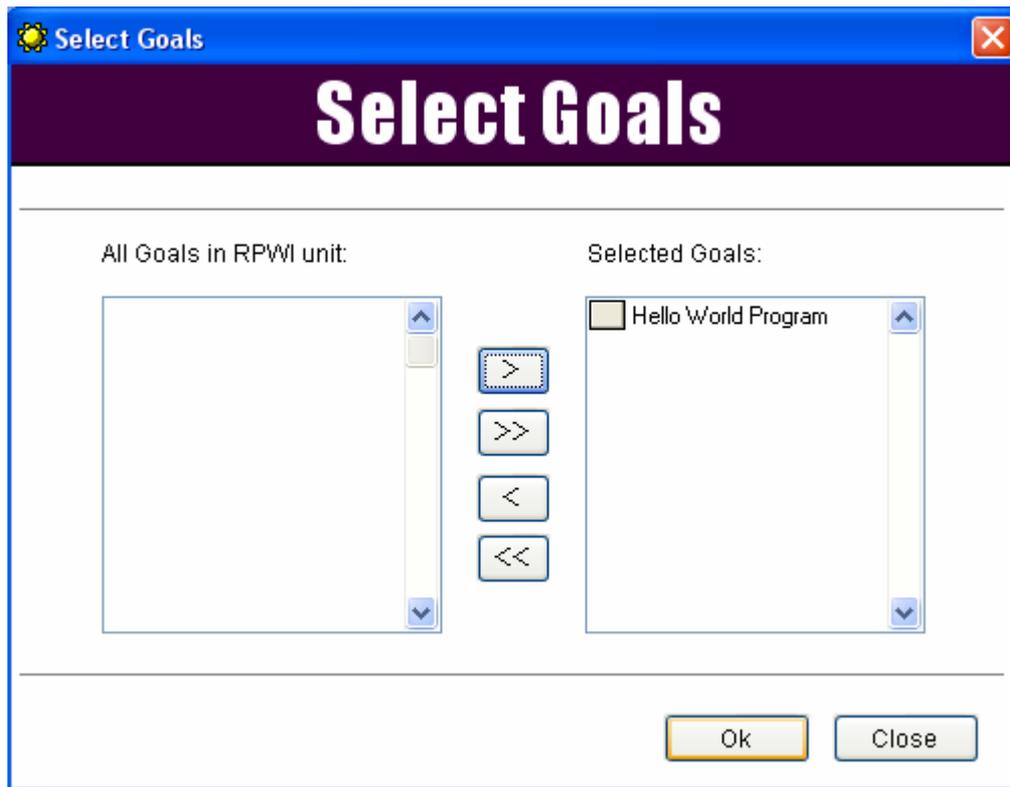
You can extract the code from the RPWI Unit at any time using the code extractor application

يمكنك استخراج الكود من وحدة البرمجة بدون كود RPWI Unit في اي وقت باستخدام تطبيق مستخرج الكود



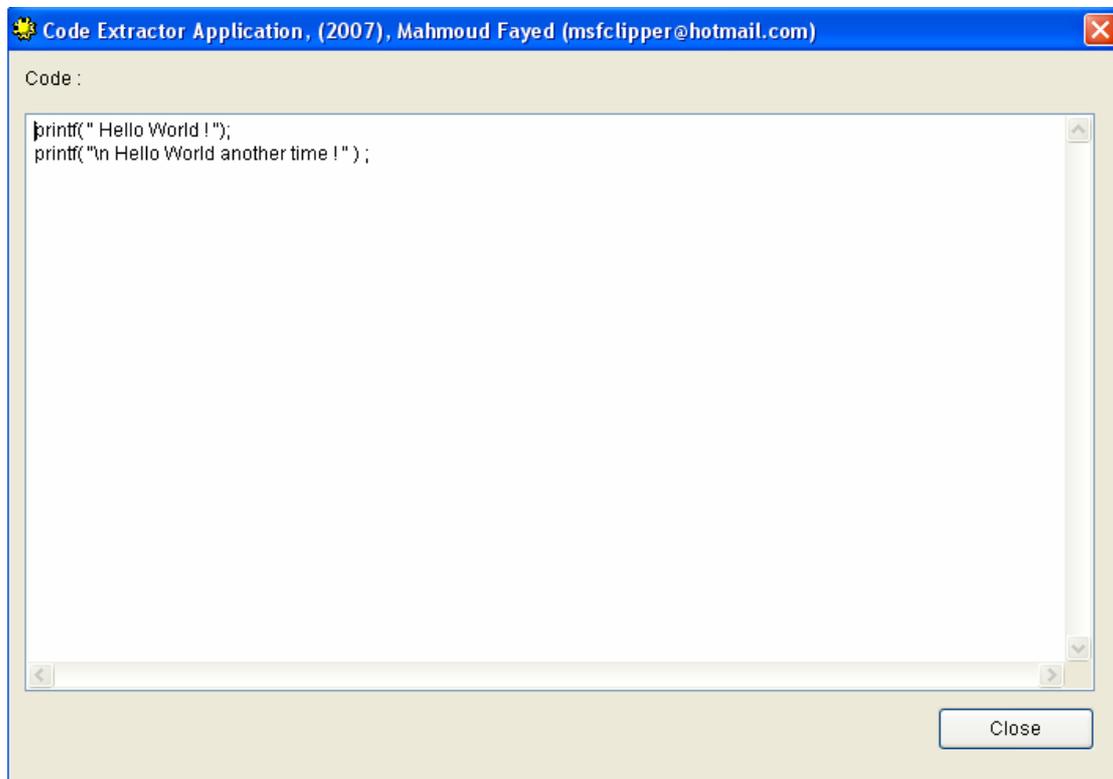
You can determine what goals you want to extract its code

يمكنك تحديد الاهداف التي تريد استخراج الاكواد الملحقة بها



the code extractor applications show the extracted code for you.

برنامج مستخرج الكود يعرض لك الكود الذي تم استخراجه

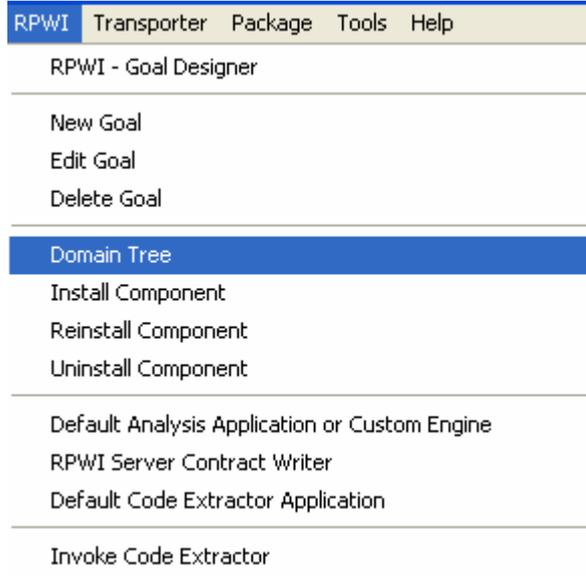


You should have Complete GUI, in you Custom Human Language for your Programming Language that you know or use.

من المفترض ان يكون لديك واجهة رسومية كاملة باللغة الانسانية التى تفهمها تعمل مع لغة البرمجة التى تعرفها او تستخدمها

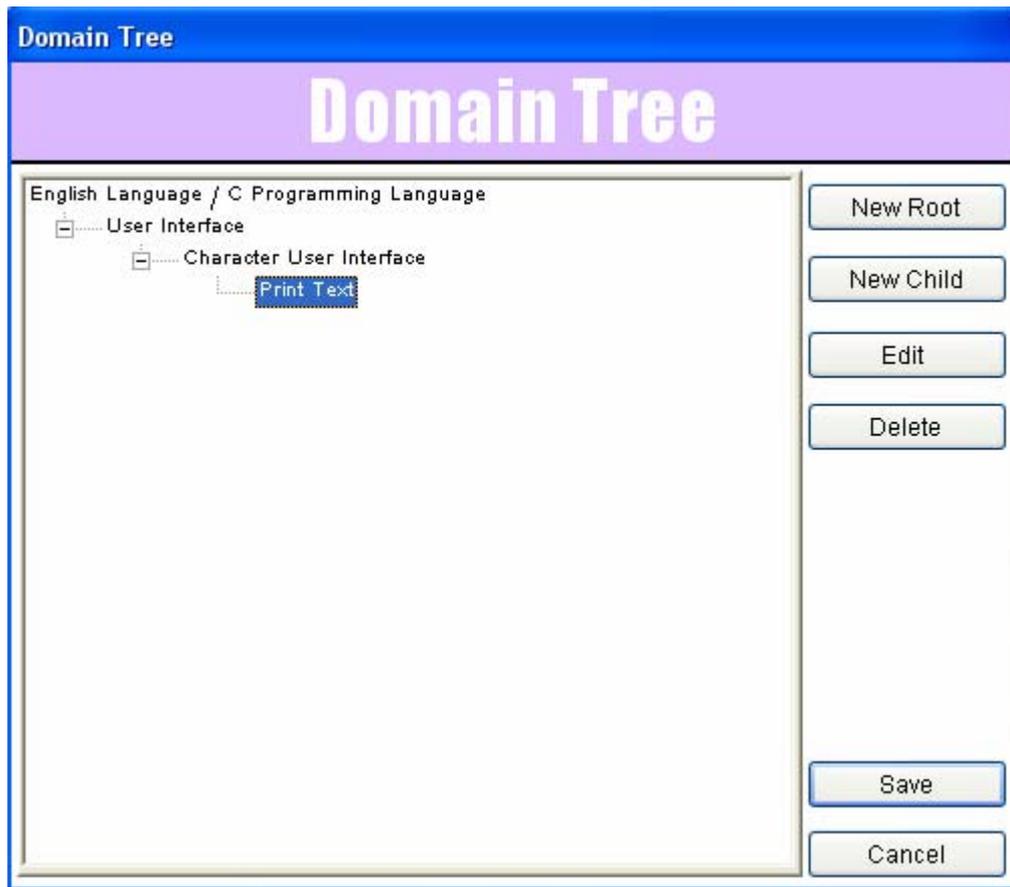
The Framework provides the designers that enable you to do all of that.

محيط التطوير يقدم لك المصمات التى تمكنك من عمل كل ذلك



from the domain tree you can manipulate the custom tree in your human language.

من شجرة المجال يمكنك ضبطها بالصورة الملائمة لك تبعا للغة الانسانية



You can customize your domain tree as you want.

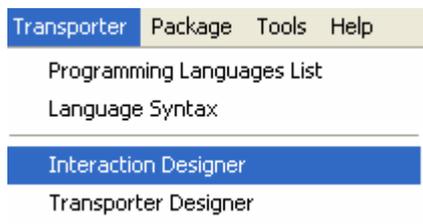
يمكنك ضبط شجرة المجال كما تريد

Now it's the time to develop our first transport

الان حان الوقت لتطوير اول ناقل

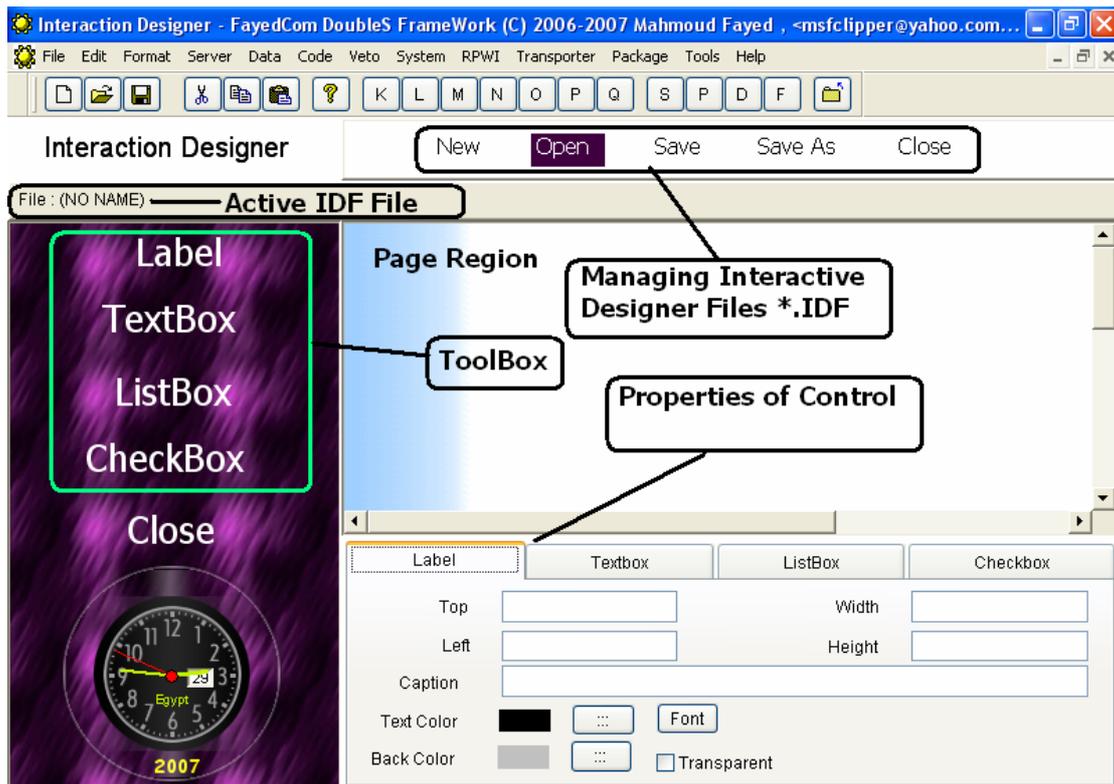
We start by designing the pages , so open interaction designer

نحن نبدأ بتصميم الصفحات – لذلك قم بفتح مصمم التفاعل



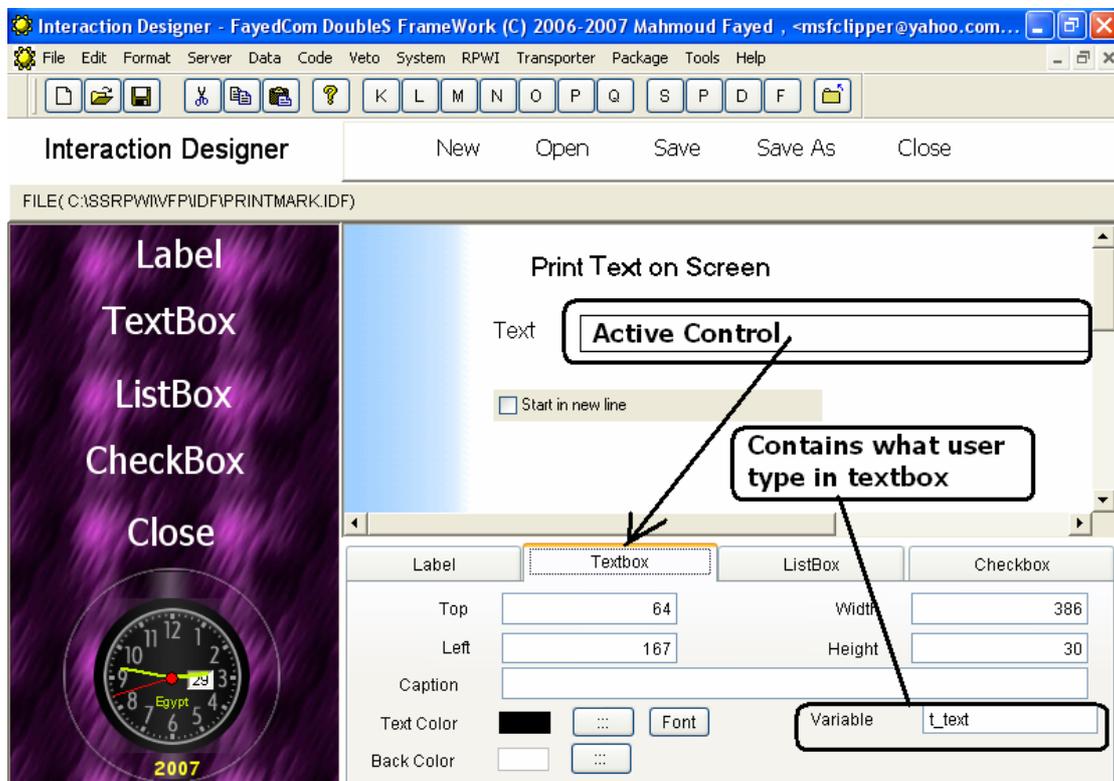
interaction designer is tiny version of form designer where you have design region, toolbox & properties tabs.

مصمم التفاعل عبارة عن اصدار صغير من مصمم النماذج حيث يكون لديك منطقة للتصميم وصندوق ادوات وخصائص



Open interaction designer file IDF, PRINTMARK.IDF

قم بفتح ملف تفاعل تم تصميمه مسبقا

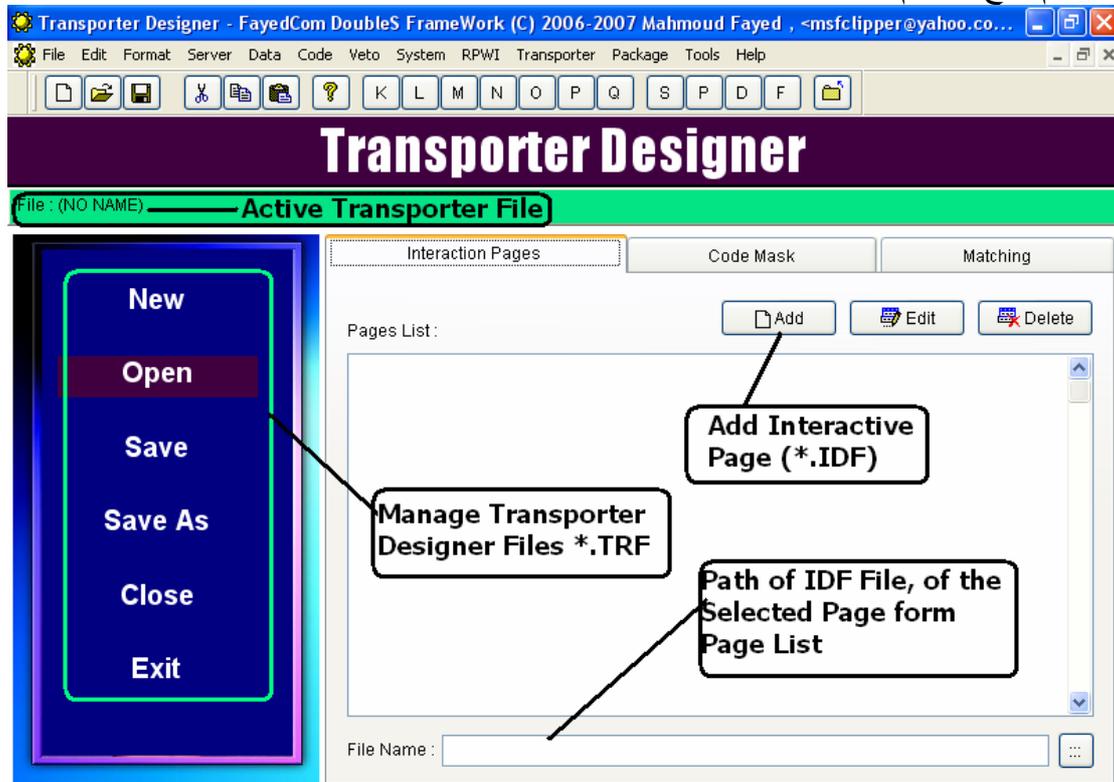


It's clear that textbox, checkbox and listbox have variable to store results of user interaction.

واضح تماما ان مربع الادخال ومربع الفحص وصندوق القائمة لكل منهما متغيرات لكي تخزن نتائج التفاعل مع المستخدم

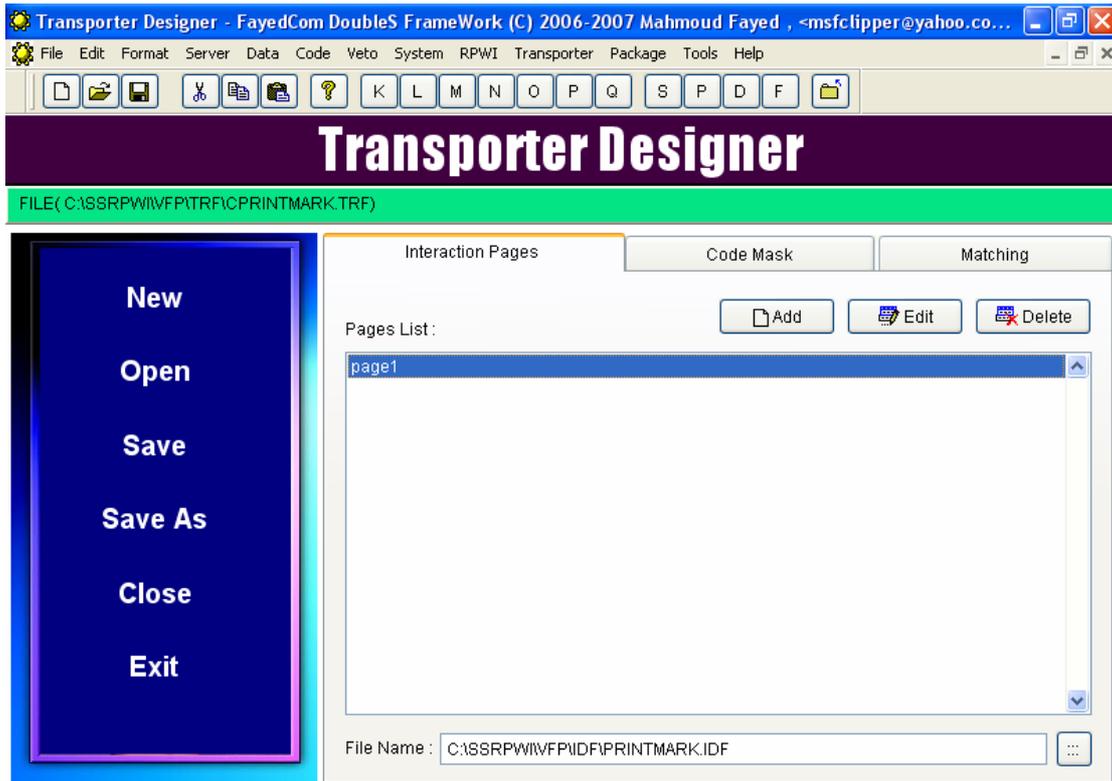
Now open transporter designer

الان قم بفتح مصمم الناقل



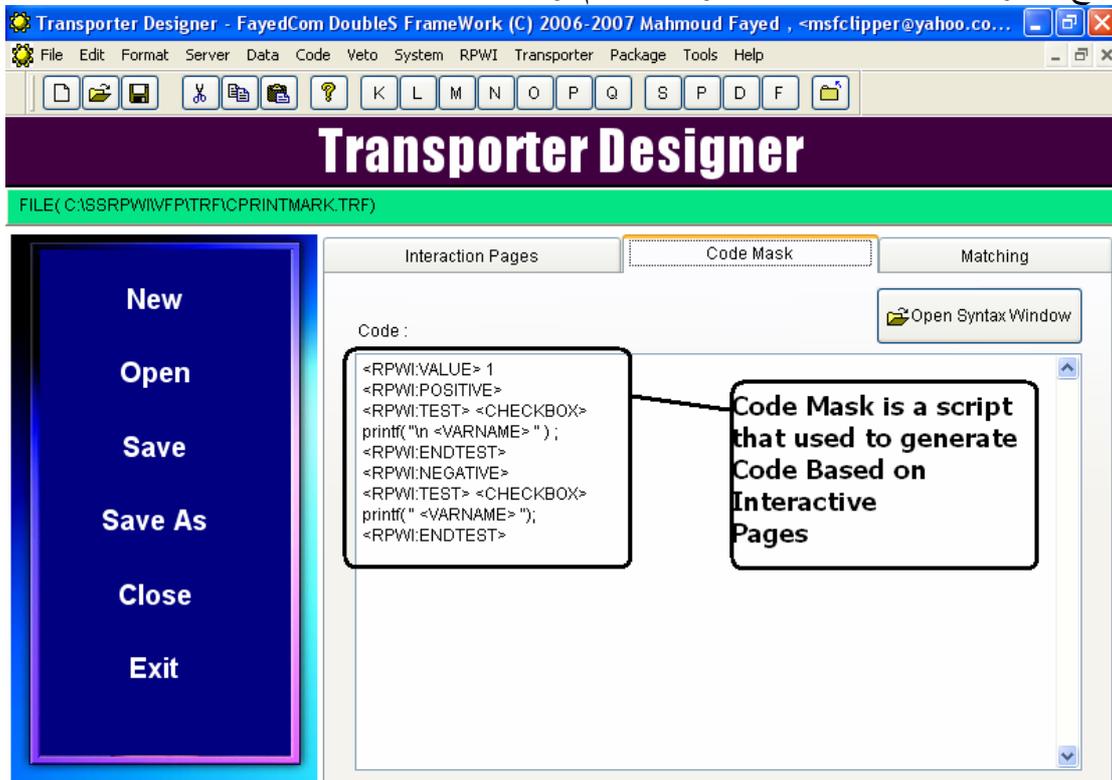
transporter designer determine interaction pages, add interaction pages as you want.

مصمم الناقل يحدد صفحات التفاعل- يمكنك اضافة صفحات التفاعل حسبما تريد



the code mask determine the generated code

قناع الشفرة code mask يحدد الكود الذي يتم توليده



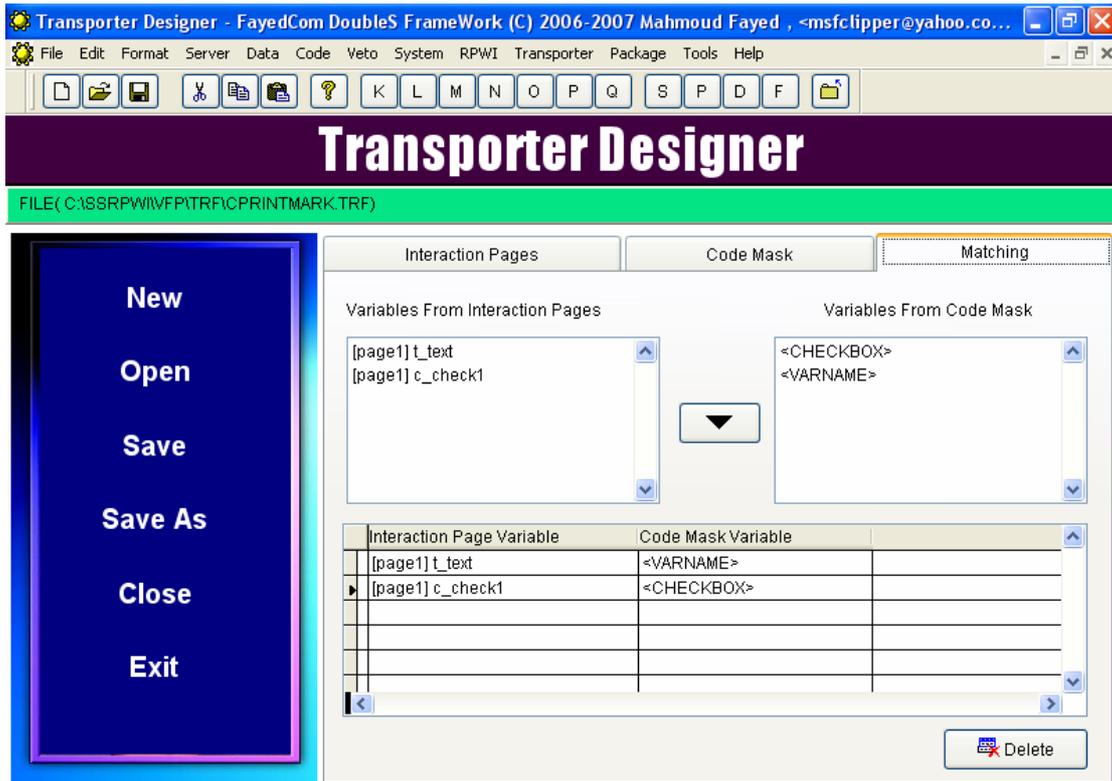
Rules to write Code Mask Script

القوانين التي تحدد كتابة قناع الشفرة code mask

- Generated code is written directly
 - الكود الذى نرغب فى توليده يكتب مباشرة
- Put variables between '<' and '>', i.e. <variable name>
 - توضع المتغيرات بين علامتى اصغرمن واكبر من
- <RPWI:POSITIVE>
 - Generate code between <RPWI:TEST> and <RPWI:ENDTEST> if the result of the test is true
 - يولد الكود داخل الاختبار TEST اذا كانت نتيجة الاختبار صحيحة
- <RPWI:NEGATIVE>
 - Generate code between <RPWI:TEST> AND <RPWI:ENDTEST> if the result of the test is false
 - يولد الكود اذا كانت نتيجة الاختبار غير سليمة
- <RPWI:VALUE> determine that value that will compared with the value of the variable after <RPWI:TEST>, if the two values are typical, the test is true, and converse.
 - يحدد القيمة التى سوف تقارن بمتغير من خلال الاختبار – اذا حدث تطابق يكون الاختبار سليم واذا لم يحدث تكون نتيجة الاختبار غير سليمة
- <RPWI:INFORMATION> used to write text will be written in information tab of the step, in goal designer.
 - يستخدم لكتابة نص يوضع فى الجزء الخاص بمعلومات الخطوة داخل مصمم الاهداف
- Nested <RPWI:TEST> and <RPWI:ENDTEST> is allowed.
 - يمكن السماح بعمل اختبار داخل اختبار
- Textbox Variable, returns what is written inside the textbox without any delimiters
 - قيم المتغيرات القادمة من صفحات التفاعل توضع مباشرة بدون ان يضيف الناقل لها اى علامات
- Checkbox Variable, returns 1 (Checked) or 0 (Not Checked)
 - مربع الفحص يعطى (1) لو كانت العلامة صح ويعطى صفر ولو لم يكن هناك علامة
- Listbox Variable, may return list item (the select item), or list item index (the selected item index) and this determined in the Interaction Designer, Default is List item index.
 - متغير قائمة العناصر ممكن ان يعطى العنصر المحدد او رقمه والحالة الافتراضية انه يرجع الرقم – ومن خلال مصمم التفاعل يمكن تغيير ذلك

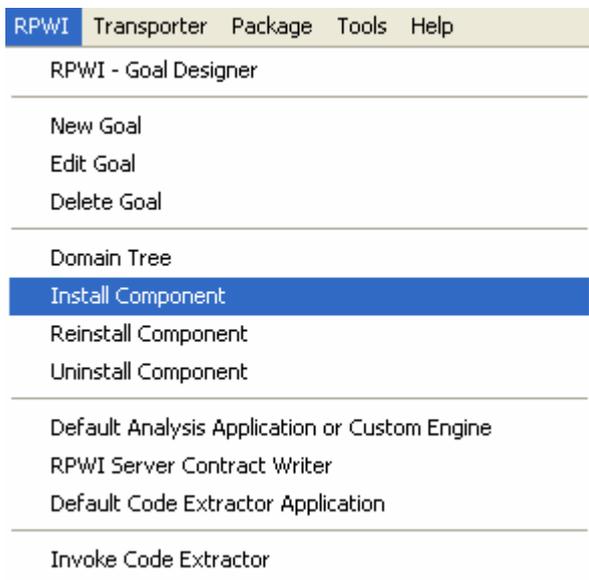
From the matching tab you can determine the pairs of variables between code mask variables & interaction pages variables.

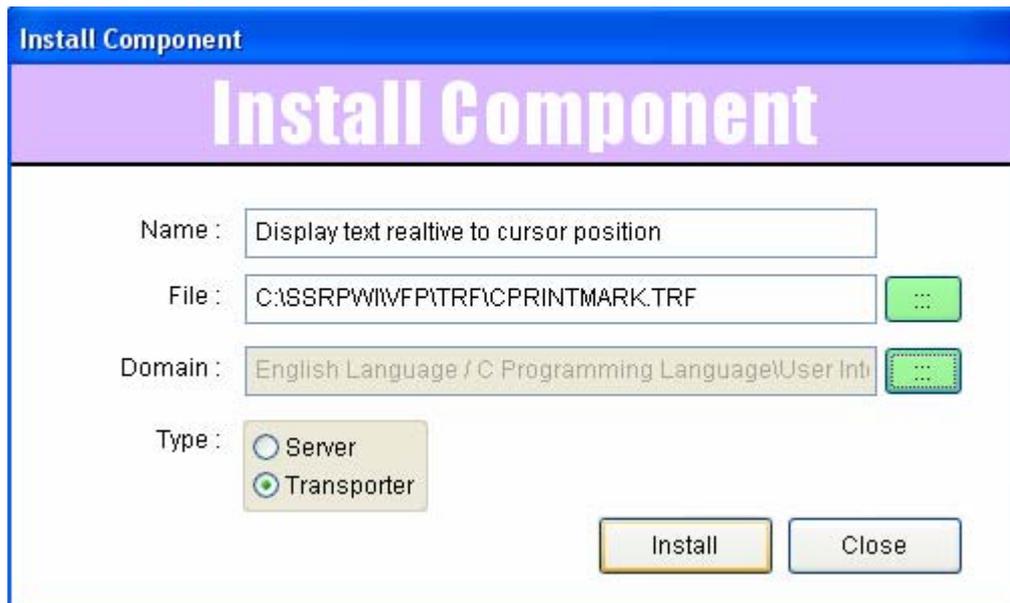
من الجزء الخاص بالتوفيق يتم تحديد ازواج من المتغيرات بين متغيرات صفحات التفاعل ومتغيرات قناع الكود



Transporter is component that can be plugged in Domain Tree
So you can install it simply.

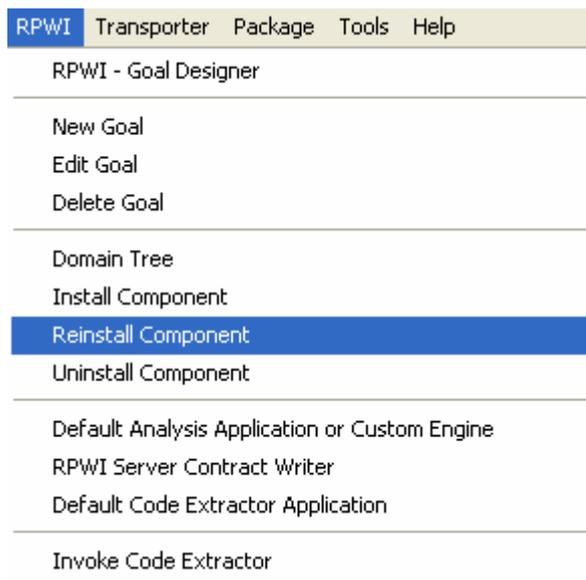
الناقل هو عبارة عن مكون يمكن ادخاله في شجرة المجال ويمكنك تحميله بسهولة

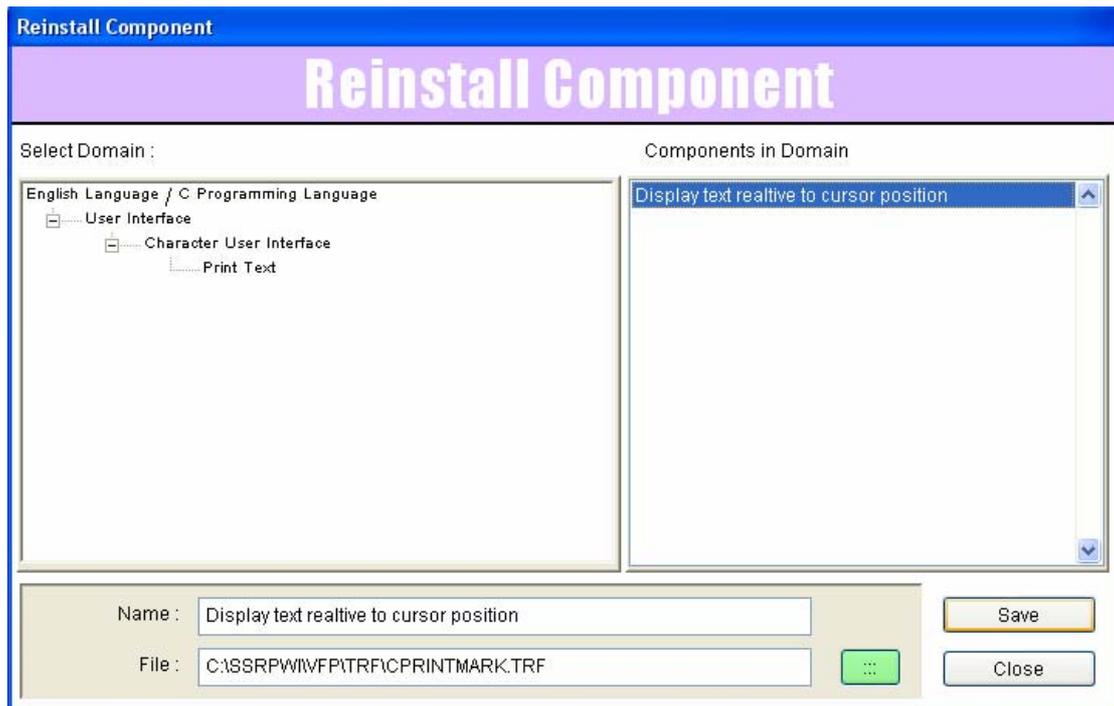




You can reinstall any component at any time

يمكنك إعادة تحميل أي مكون في أي وقت



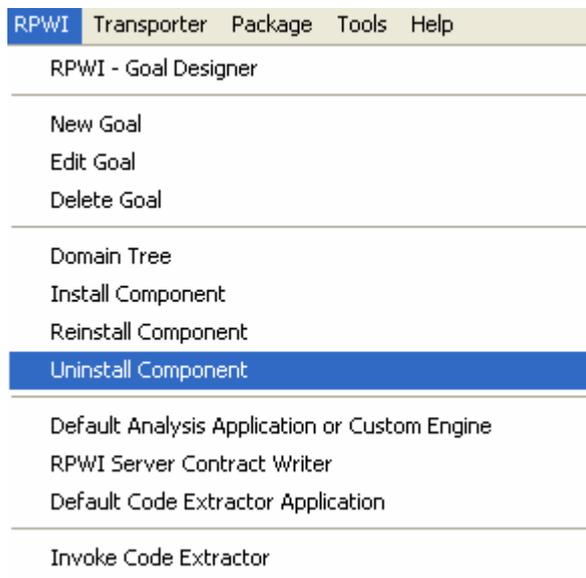


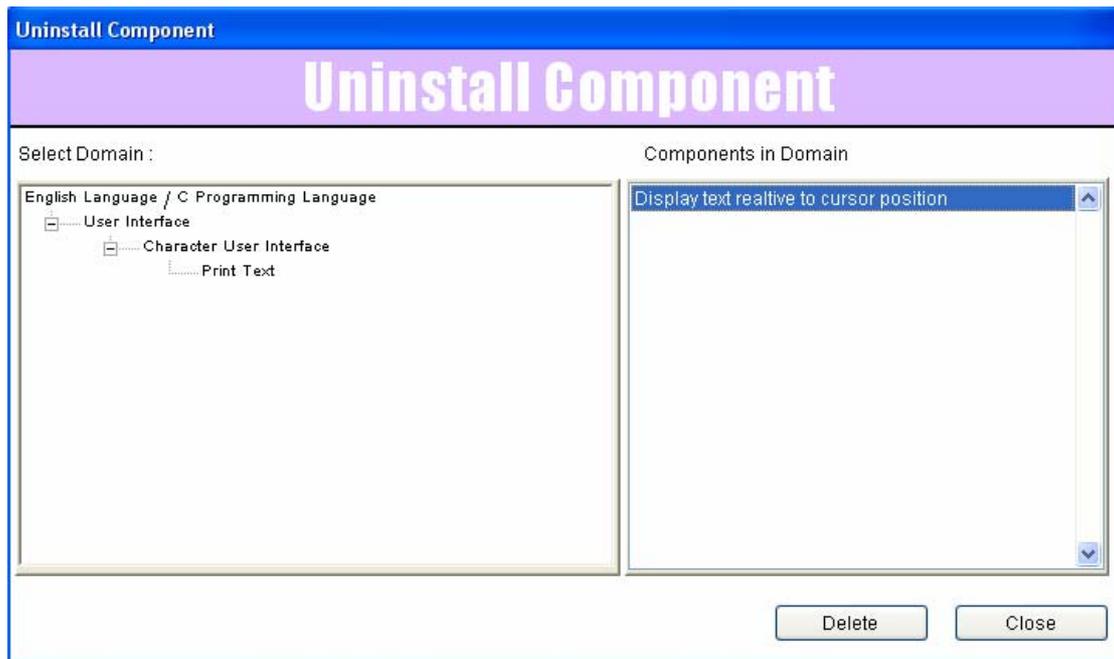
reinstallation enable you to change the name and the file of the transporter.

اعادة التحميل تمكنك من تغيير اسم الملف واسم الناقل

Also you can uninstall the transporter at any time.

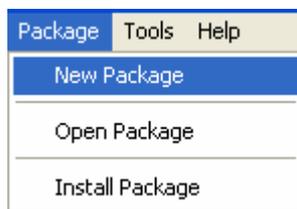
ايضا يمكننا ازالة اى ناقل فى اى وقت

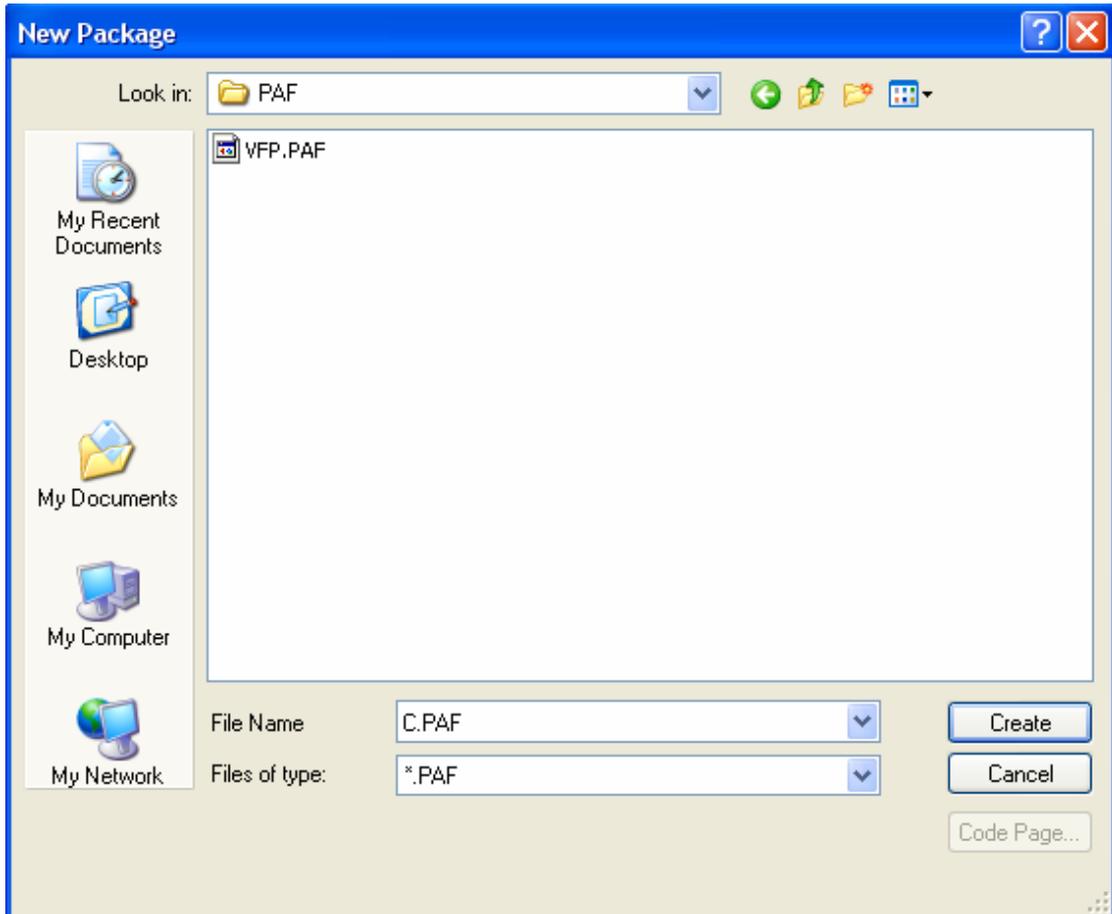




Domain tree can be saved as package so it can be installed at any time or ported to another user , remember that the package still require the components (Transporters & Servers) files because it's not embedded in the package , the package only store Domain Tree and installed Components details.

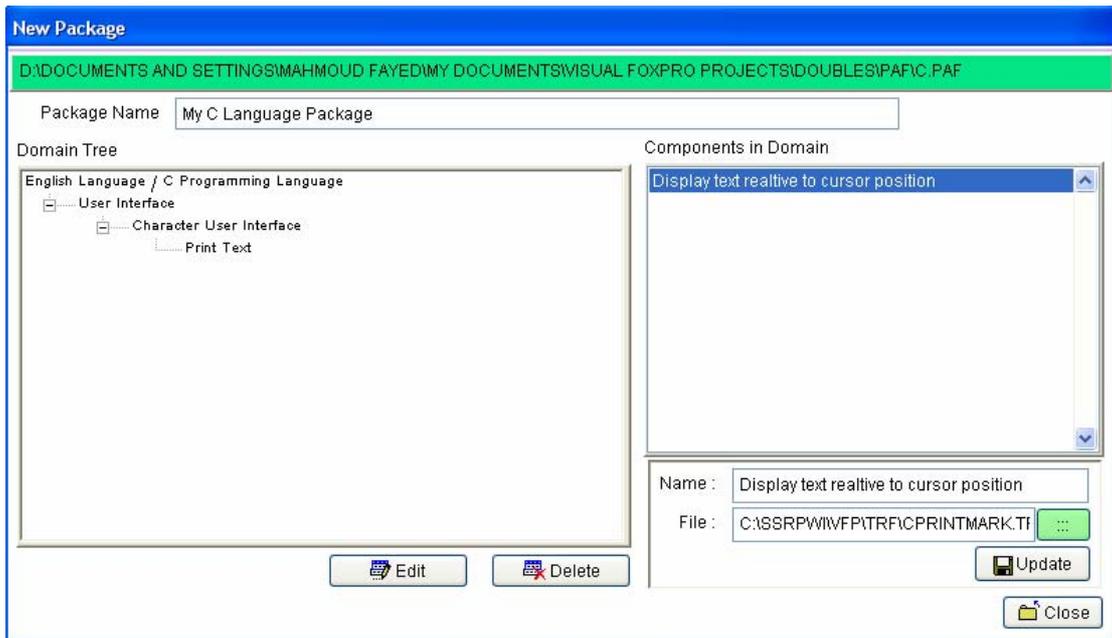
يمكنك حفظ شجرة المجال فى الحقيقية – ولكن تذكر ان الحقيقية تشترط وجود الخوادم والنواقل –
فهى لا تتضمنهما بداخلها

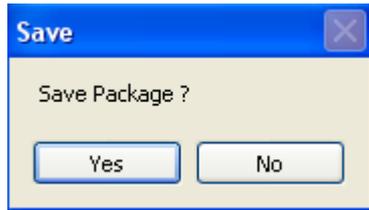




package files have the extension *.PAF

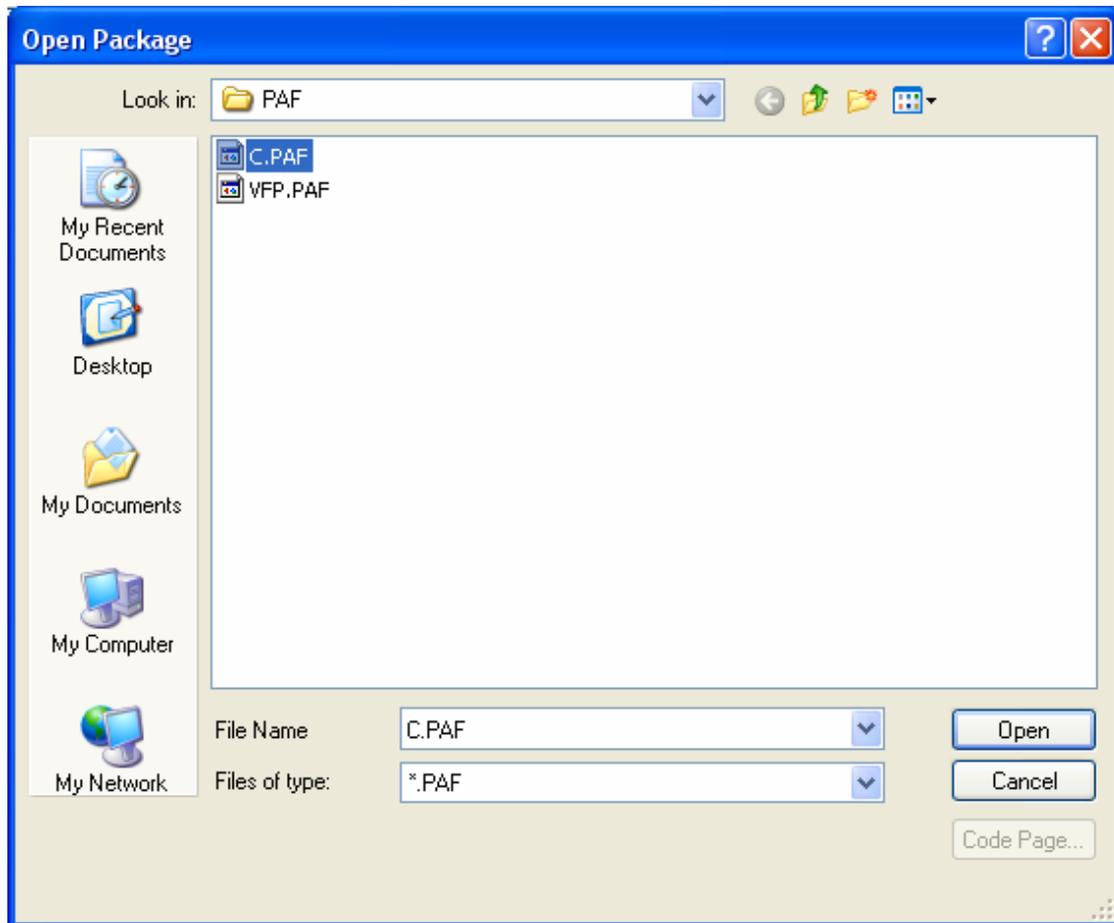
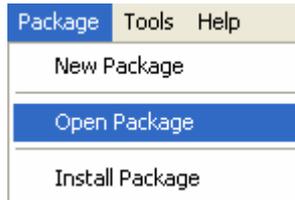
ملفات الحقيبة تحمل الامتداد PAF

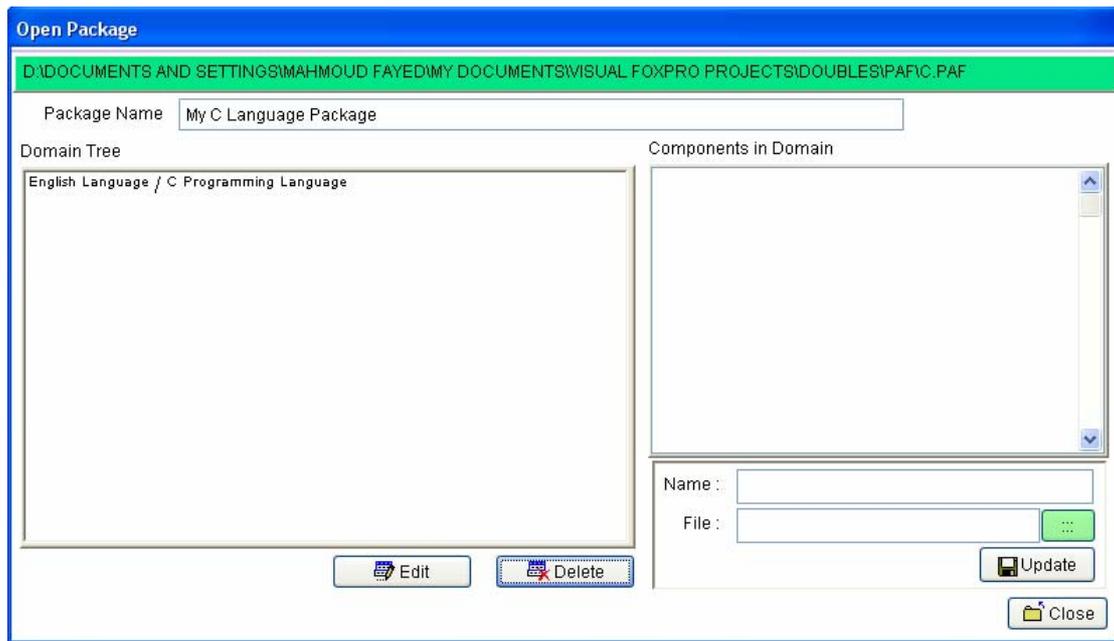




It's clear that when you create new package , its content is extracted from the domain tree.

من الواضح انه لعمل حقيبة جديدة فانها تستخرج من شجرة المجال



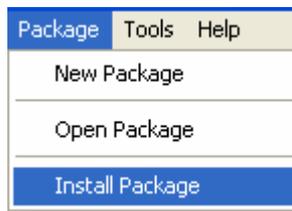


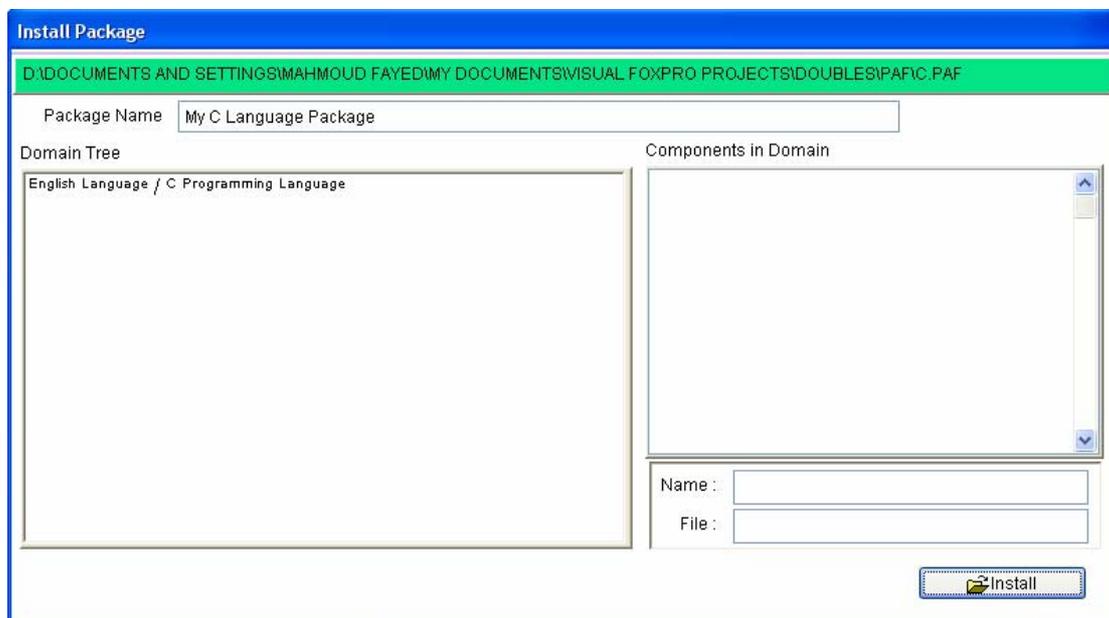
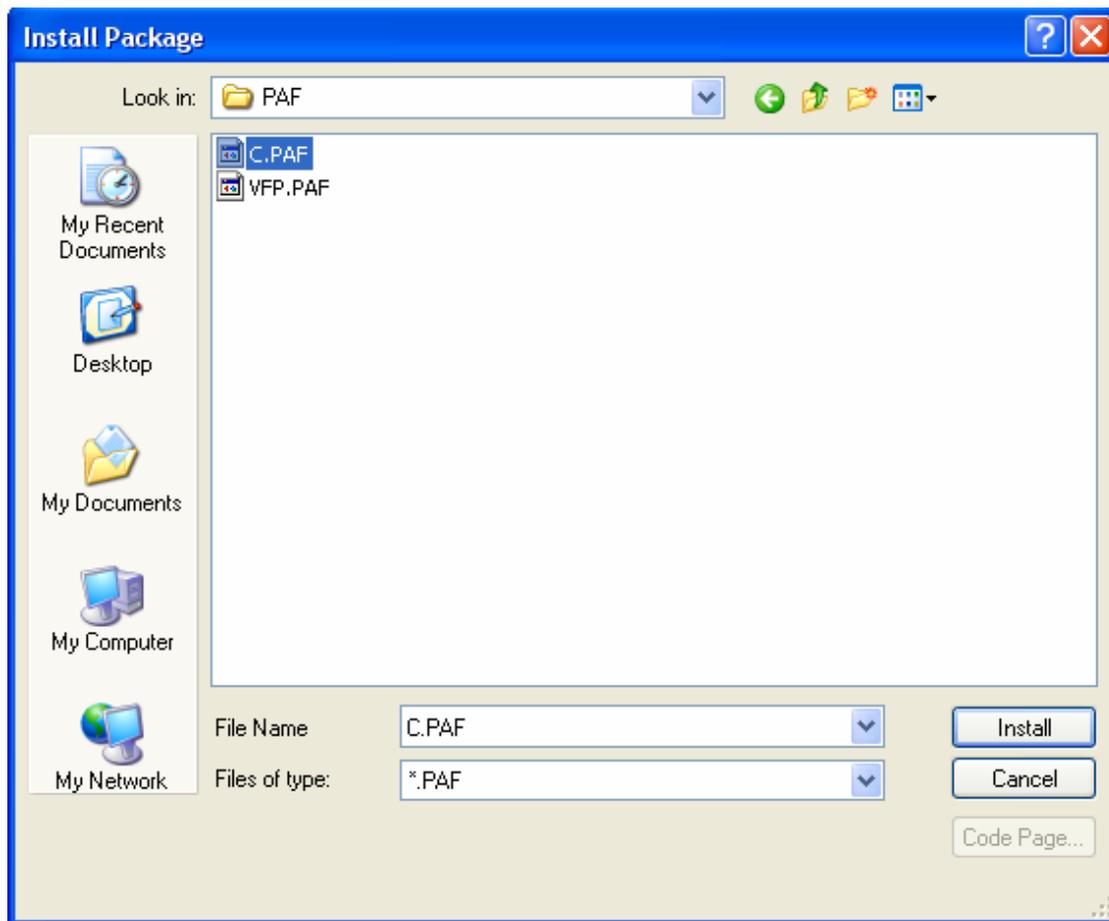
You can open package at any time, change domain tree, reinstall components

يمكنك فتح الحقيبة في اى وقت وتغيير شجرة المجال – واعدادة تحميل المكونات

You can install any package at any time.

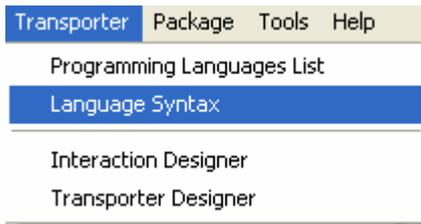
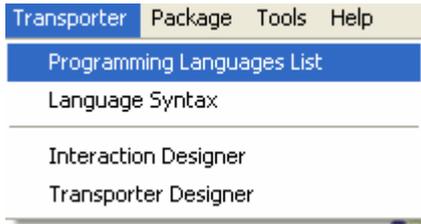
يمكنك تحميل الحقيبة في اى وقت

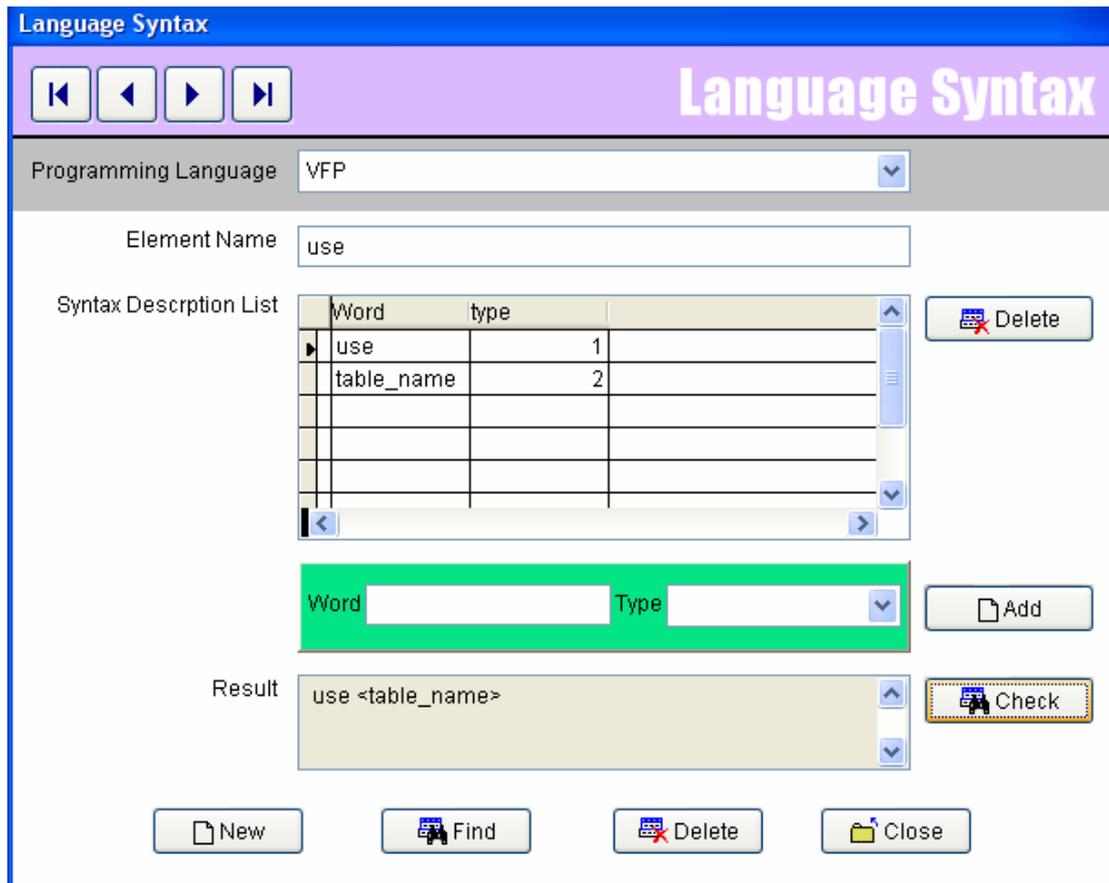




In transporter you need to write the generated code, so it's helpful to have Syntax files that help you write Generated code with the same format

في الناقل انت بحاجة لكتابة الشفرة التي يتم توليدها لذلك لديك ملفات القواعد النجوية التي تساعدك على كتابة الشفرة بنفس الشكل





The Question now, what about the Server?

السؤال الآن – ماذا عن الخادم؟

The Server gets its input from the analysis application as text files
 For default analysis application the input is a_1.txt, a_2.txt, a_3.txt &
 a_4.txt

ان الخادم يحصل على الدخل في صورة ملفات تنتج من التطبيق المسئول عن تحليل البيانات
 وهذه الملفات هي اربعة ملفات

The output from the server to goal designer is 3 text files

O_code.txt, o_ana.txt & o_inf.txt

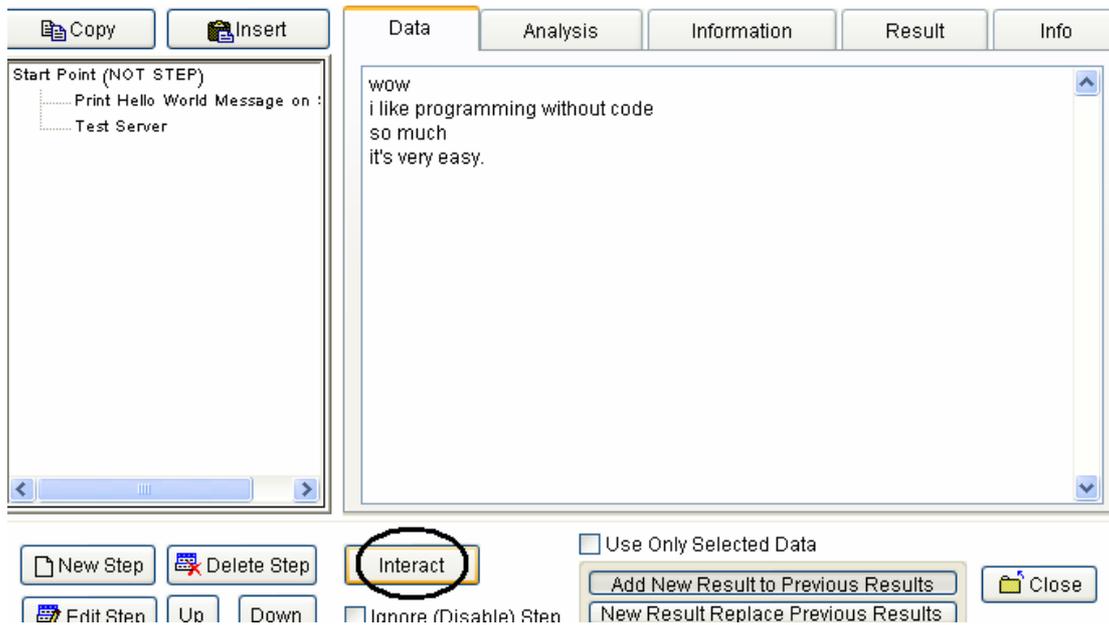
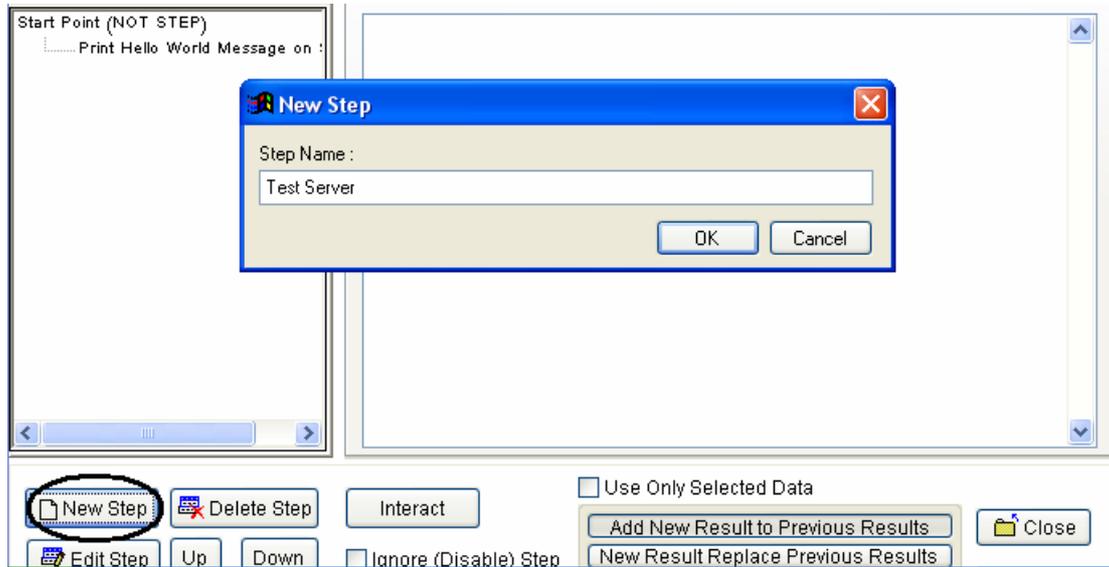
الخرج من الخادم عبارة عن ثلاث ملفات نصية يرجعها الى مصمم الاهداف

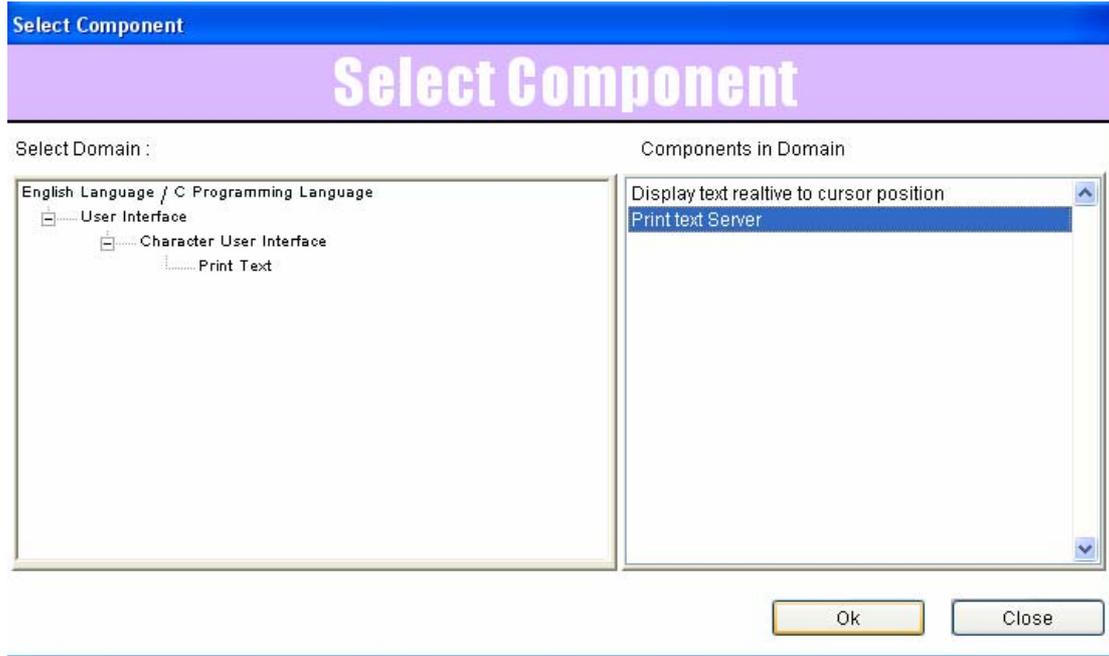
RPWI SERVER , Input/Output Files

File	Type	Usage
A_1.TXT	Input	Data Statics
A_2.TXT	Input	Each Line Alone (Line per Line)
A_3.TXT	Input	Each Word Alone (Line per word)
A_4.TXT	Input	Each Letter Alone (Line per Letter)
O_ANA.TXT	Output	What analysis method used
O_INF.TXT	Output	What Information is available

You can write RPWI Server using any programming language you know or like, also its good idea to use DoubleS Technology for this simple job.

يمكنك كتابة خادم البرمجة بدون كود باى لغة برمجة تعرفها او تحبها
ولكن يفضل استخدام نمط البرمجة الخادم الممتاز لهذه المهمة البسيطة





Server is component can be installed in Domain Tree.

الخادم هو مكون يمكن تحميله في شجرة المجال

Server Passport is used to know the details of the server under usage like Server Name, Purpose, Author, Supported human Languages, Supported programming languages and License

جواز سفر الخادم يستخدم لمعرفة تفاصيل الخادم تحت الاستخدام مثل اسم الخادم والغرض منه والمؤلف – واللغات الانسانية التي يدعمها – ولغات البرمجة – ورخصة الاستخدام

Server Passport

Server Passport

Feild	Value
Server Name	print text on screen
Author	Mahmoud Fayed
Date	12/07/2007
Company	FayedCom
Address	Egypt
Contact	msfclipper@hotmail.com
Programing Languages	CA-Clipper/Harbour/xHarbour/xBase++/CA-Visual Objects/Visual FoxPro
Human Languages	English

License

Freeware & Open Source

click "ok" button, to continue

اضغط زر OK لكي تستمر

```

D:\WINDOWS\system32\cmd.exe
RPWI Server : Print Text
  xHarbour Output
  C Output
  
```

Goal : Hello World Program

Goal Designer

Custom User Interface (Human Language)

Copy Insert Data Analysis Information Result Info

Start Point (NOT STEP)
Print Hello World Message on :
Test Server

```
printf("wow \n");  
printf("i like programming without code \n");  
printf("so much \n");  
printf("it's very easy. \n");
```

RPWI Transporter Package Tools Help

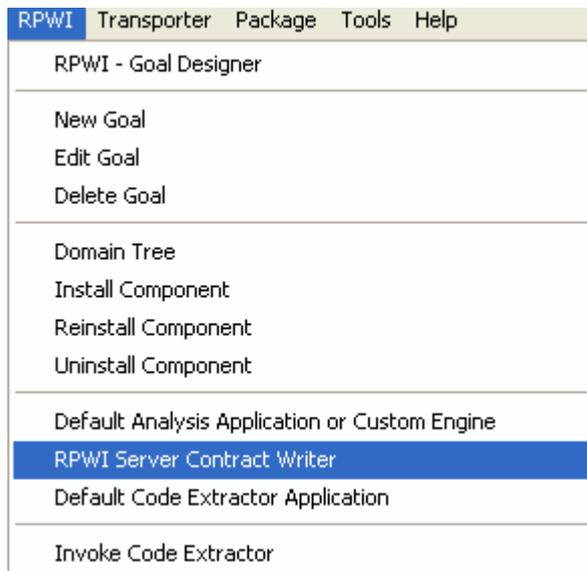
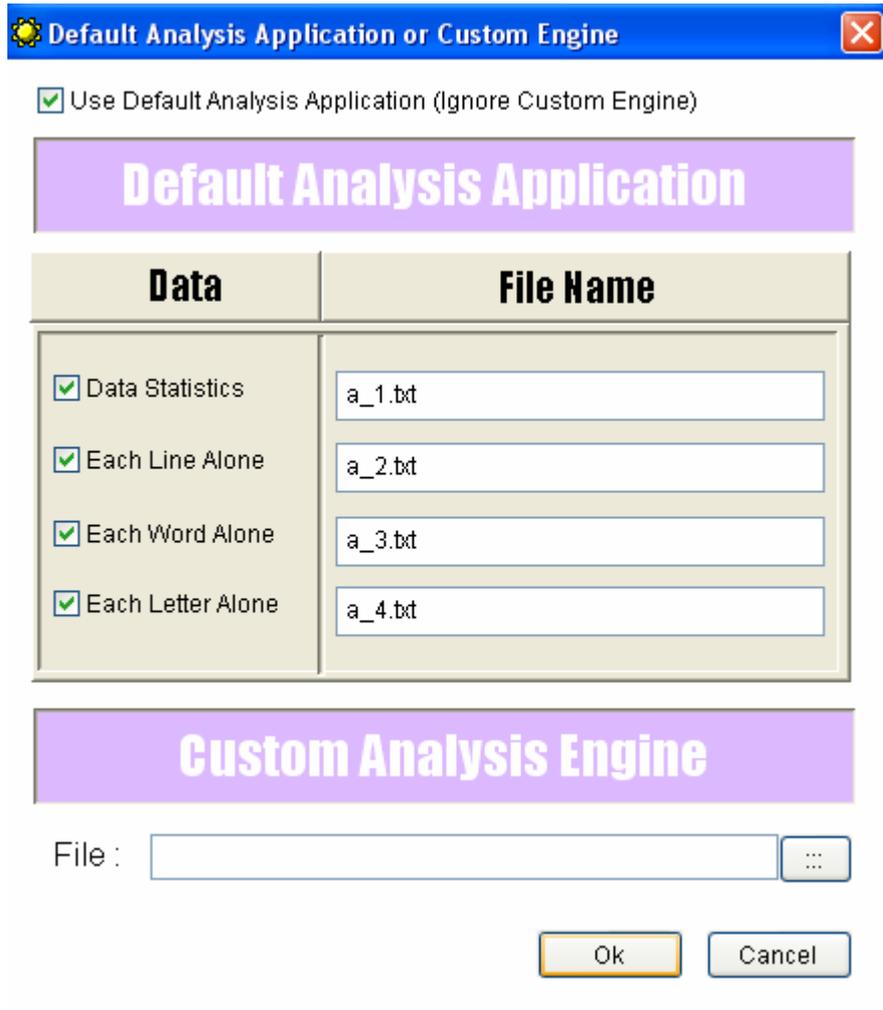
RPWI - Goal Designer

New Goal
Edit Goal
Delete Goal

Domain Tree
Install Component
Reinstall Component
Uninstall Component

Default Analysis Application or Custom Engine
RPWI Server Contract Writer
Default Code Extractor Application

Invoke Code Extractor



RPWI Server Contract Writer

Server Passport

Field	Value
Server Name	<input type="text"/>
Author	<input type="text"/>
Date	<input type="text"/>
Company	<input type="text"/>
Address	<input type="text"/>
Contact	<input type="text"/>
Programming Languages	<input type="text"/>
Human Languages	<input type="text"/>

License

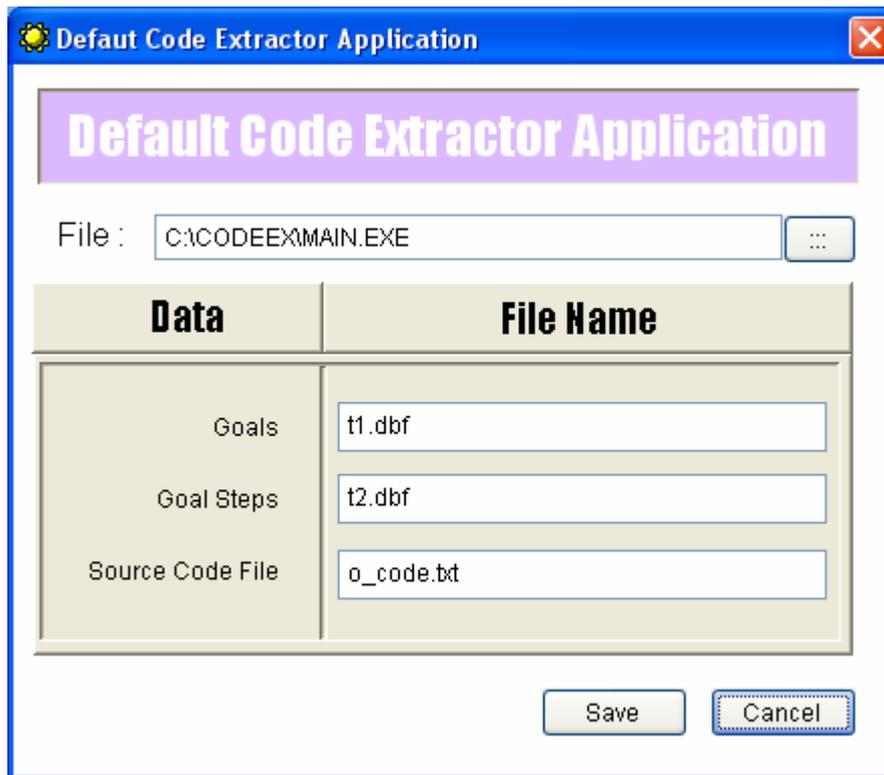
You can Describe your server as you want, then click "Write" Button to save the description in Server Passport File *.SPF , but be sure that the name of the "Server Passport File" is the same as the name of the RPWI Server Application *.EXE

يمكنك وصف الخادم كما تريد في جواز السفر الذى يتم حفظه في ملف يحمل اسم ملف التطبيق الخاص بالخادم ولكن بامتداد SPF بدلا من EXE

For example:

If the RPWI Server Application is: "MyServer.EXE"

Then the passport file must be: "MyServer.SPF"



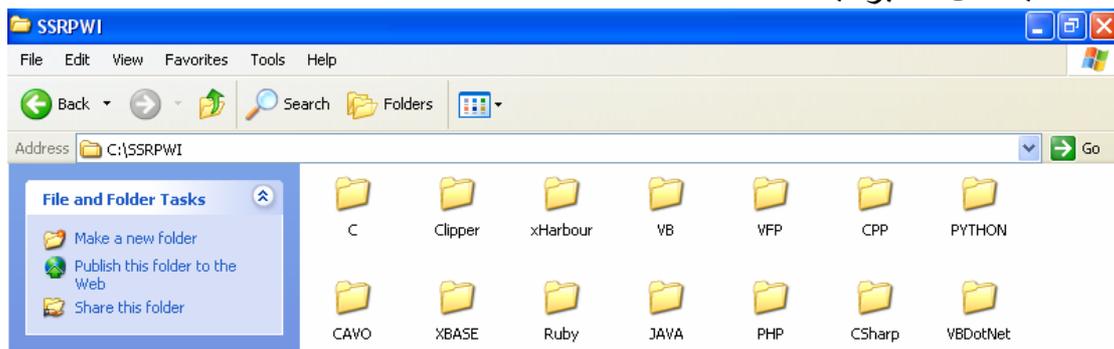
Organizing Component Files

تنظيم ملفات المكونات

In path C:\SSRPWI فى المسار

There is folder for each programming language

هناك مجلد لكل لغة برمجة

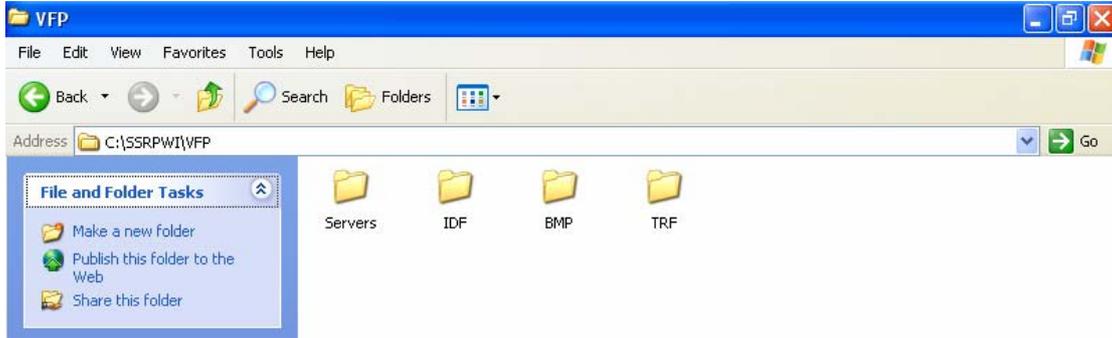


Each folder contains four sub folders

كل مجلد يشمل اربعة مجلدات فرعية

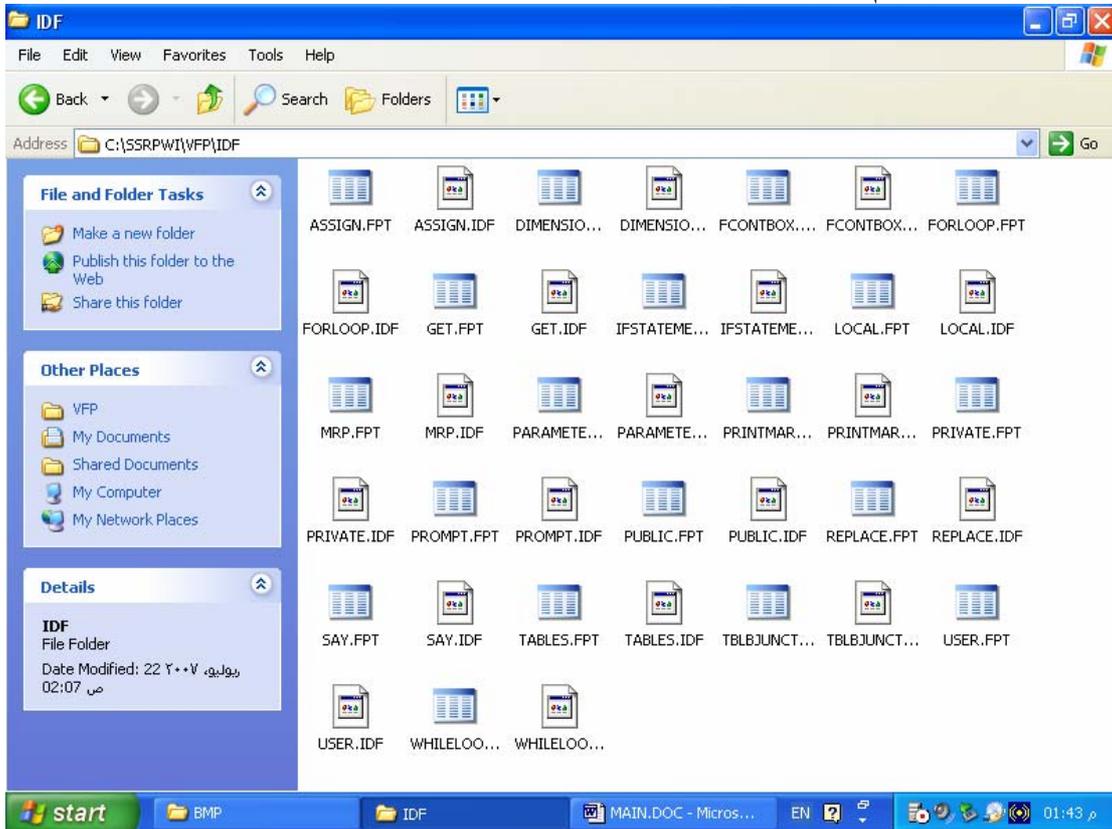
- + Servers تطبيقات خوادم البرمجة بدون كود
 - o Contains RPWI Servers Applications
- + IDF ملفات الصفحات التفاعلية
 - o Contains Interactive Pages
- + BMP ملفات الصور
 - o Contains bitmap files used by Interactive Pages

- ✚ ملفات الناقل TRF
 - Contains Transporter Files



Each Interactive Page represented by two files FILENAME.IDF File & FILENAME.FPT File.

كل صفحة تفاعلية يتم تمثيلها بملفين – ملف IDF و ملف FPT



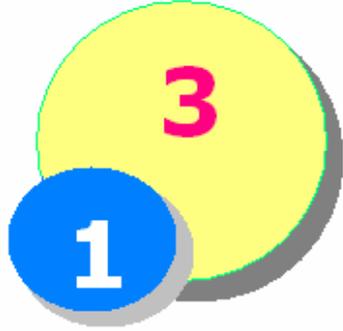
Each Transporter represented by two files FILENAME.TRF File & FILENAME.FPT

كل ناقل يمثل بملفين – ملف TRF و ملف FPT

سمات التطبيقات المتطورة



السمات المميزة للتطبيقات



سبق لنا وان اوضحنا الفرق جيدا بين التطبيقات والنظم - وهذا الباب يهدف الى استعراض السمات المميزة للتطبيقات لبيان مدى سهولة برمجة وتطوير هذه النوعية من البرمجيات - ثم بعد ذلك يعود الكتاب للحديث من جديد عن النظم ويستعرض ادوات برمجة النظم - ومن ثم يتطرق الى فن برمجة النظم.

بخصوص تطبيقات قواعد البيانات

كما نعلم ان تطبيقات قواعد البيانات ليست كلها بسيطة - بل يندج بعضها تحت مصطلح النظم حيث تكون الحاجة الى الابتكار - اما بخصوص التطبيقات فهي مصنفة الى ثلاثة انواع

+ تطبيقات تعمل على جهاز واحد Desktop Applications

+ تطبيقات للشبكة المحلية Server-Based Application

o حيث يتم عمل مشاركة Share لملفات البيانات

+ تطبيقات الزبون الخادم Client-Server

+ تطبيقات الانترنت Web applications

والسمات الاساسية لهذه التطبيقات

+ استخدام مدير قاعدة بيانات DBMS يتولى ادارة قاعدة

البيانات بحيث يعزل المبرمج عن كيفية تخزين البيانات

الفعلى على اقراص التخزين - ويريحك من التفكير فى

عمليات تداول البيانات من الاضافة والتعديل والبحث

والحذف - كما يقدم ملامح اخرى كثيرة مثل السرية

والامان.

+ استخدام واجهة للمستخدم قياسية تتوفر بصورة تلقائية

من قبل ادوات التطوير - وقد توجد مصمات Designers

ومعالجات Wizards تجعل عملية تصميم واجهة

المستخدم عملية فى غاية البساطة والمتعة

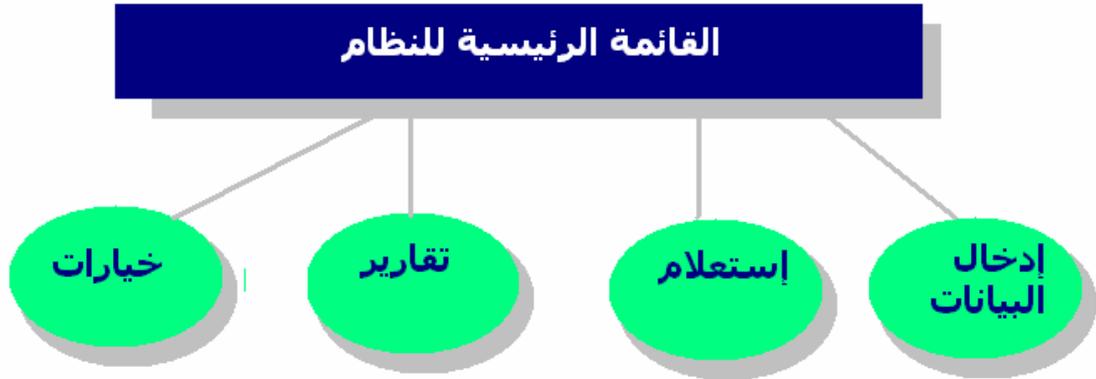
كل ما على المطور او المبرمج كتابة اكواد للربط بين واجهة المستخدم وقاعدة البيانات - ويكمن هنا دور نمط البرمجة مثل (برمجة الكائنات - البرمجة الهيكلية - برمجة الخادم الممتاز)

احيانا قد توجد ادوات توفر على المطور او المبرمج عملية كتابة الكود حيث يتم الربط بين واجهة المستخدم وقاعدة البيانات بصورة مباشرة

وفيما يلي صور ل احد تطبيقات قواعد البيانات والذي يقوم بادارة المؤسسات التجارية (المبيعات + المخازن + الموظفين) وهو من برمجة مؤلف الكتاب - باستخدام Visual FoxPro 9 SP2

ونلاحظ الاهتمام بمظهر التطبيق - لان عملية برمجة التطبيقات اصبحت لا تستغرق وقتا - مما يسمع بالاهتمام بالمظهر.





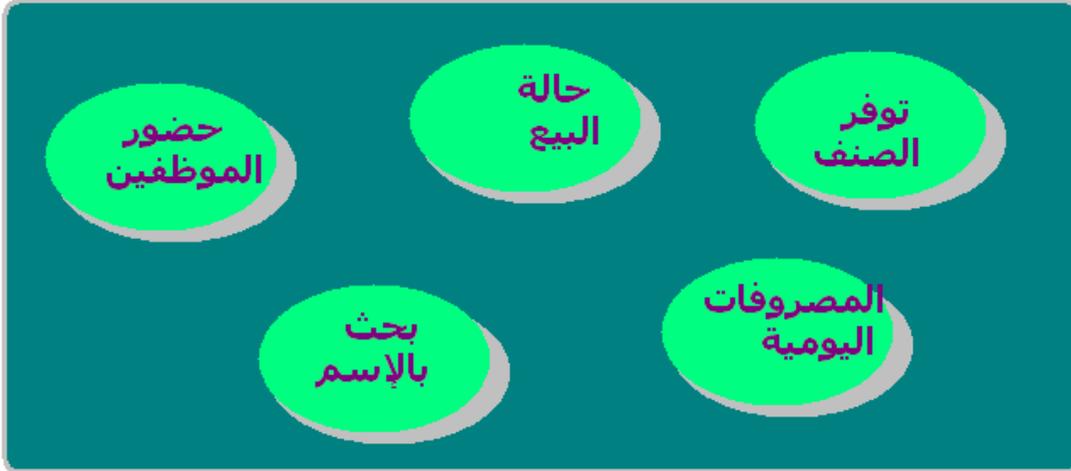
يتكون النظام من ٤ عناصر رئيسية

- ١ - إدخال البيانات
- ٢ - الاستعلام عن أى شىء داخل النظام
- ٣ - استخراج تقارير مميزة بسرعة فائقة
- ٤ - خيارات لتسهيل بعض المهام



فى اى برنامج فى العالم - قبل الحصول على النتائج المطلوبة لابد من تغذية البرنامج بالبيانات ويقدم البرنامج اسهل الادوات لكى يتم تسجيل البيانات بمنتهى المرونة.

الاستعلامات المجهزة داخل النظام



بعد إدخال البيانات الى الحاسب من خلال البرنامج تظهر الفائدة العظيمة حيث يمكن الاستعلام (البحث) عن اى معلومة بمنتهى السهولة والسرعة

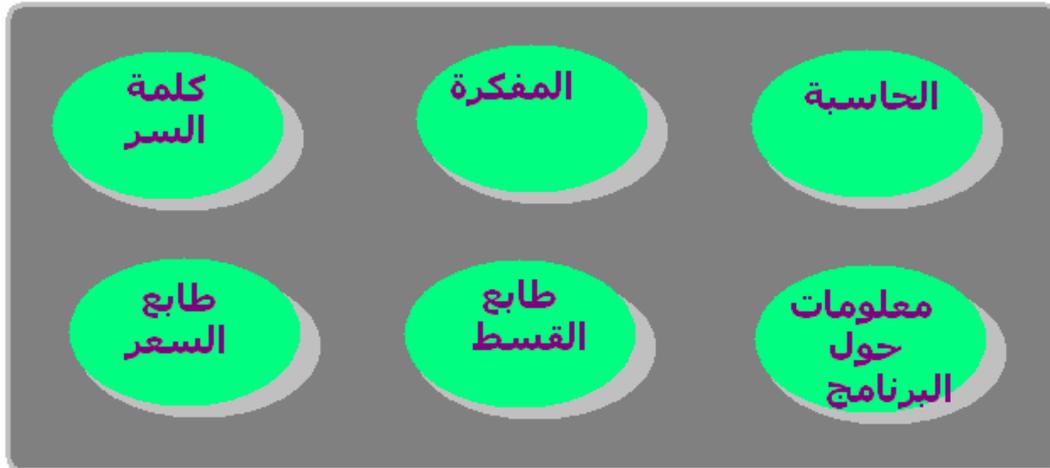
التقارير المجهزة داخل البرنامج



ان اى برنامج فى العالم تظهر قوته فى حجم وشمولية التقارير التى يصدرها - يريحك البرنامج من العناء الشديد والوقت الضائع

فى استخراؒ التقارير ويقوم هو بهذة المهمو بدلا منك بمنتهى
السرعة والكفاءة

خيارات اضافة



هى عملية لتسهيل اداء بعض المهام بدلا من اغلاف البرنامج
لتشغيل الحاسبة او المفكرة - فانك تصل اليهم بسهولة وبسرعة
ويتيح البرنامج عمل كلمة سر - واصدار طوابع تحمل كلمة
(القسط) او (السعر) فى حالة الرغبة فى الحصول على طوابع
عامة غير محددة لصنف معين



الشاشة الرئيسية للنظام
كما نرى توجد ساعة لمعرفة الوقت فى اى لحظة - حيث اننا
فى عصر الزمن - والوقت من ذهب

البيانات هى

- ١ - معلومات الاصناف
- ٢ - الاصناف
- ٣ - الاسعار
- ٣ - الواردات
- ٤ - الموردين
- ٥ - الادارة الكاملة للاصناف
- ٦ - فاتورة قبض نقدي
- ٧ - بيانات العملاء
- ٨ - فاتورة قبض اجل
- ٩ - نظم التقسيط
- ١٠ - بيانات الموظفين
- ١١ - حضور وانصراف الموظفين
- ١٢ - المصروفات اليومية
- ١٣ - المصروفات الشهرية



شاشة الاستعلام

تتيح شاشة الاستعلام

- الاستعلام عن توفر الصنف
- حالة بيع صنف معين
- حضور الموظفين
- المصروفات اليومية
- بحث بالاسم



شاشة التقارير

- + تتبح الحصول على
- + قائمة الاصناف والاسعار
- + كارت اسعار صنف محدد
- + كارت سعر صنف محدد
- + مجموعة من كروت السعر
- + طلبية زبون
- + صافي الايراد
- + الكميات المباعة
- + بيانات العملاء
- + بيانات المودين
- + جرد المخازن
- + الواردات من جميع الاصناف
- + الواردات تبعا لصنف معين
- + الواردات تبعا لمورد معين
- + الواردات تبعا لمجموعة



شاشة الخيارات

- تتيح هذه الشاشة
- الالة الحاسبة
 - المفكرة
 - كلمة السر
 - معلومات حول البرنامج
 - طابع القسط
 - طابع السعر

شاشات مختارة من ضمن شاشات البرنامج

شاشة الادارة الكاملة للاصناف

الاصناف الاساسية الاصناف الفرعية العدد :- ٨

<p>تليفير يونات</p> <p>تلاجة</p> <p>غسالة</p> <p>بوتجاز</p> <p>ديب فريزر</p> <p>الاتات</p> <p>الادوات المنزلية</p>	<p>بوتجاز - يونى فيرسال ٥٥ * ٨٠ كلاسيك - ٥ شعلة استالسثيل</p> <p>بوتجاز - يونى فيرسال - ٤ شعلة استالسثيل</p> <p>بوتجاز - ايد ستار ٤ عيون</p> <p>بوتجاز - يونيفرسال ٩٠*٦٠ استلس - ٥ شعلة</p> <p>بوتجاز - فريش ٩٠/٦٠ بالامان - ٥ شعلة</p> <p>بوتجاز - يونى فيرسال ٩٠/٦٠ بالامان - ٥ شعلة</p> <p>بوتجاز - اوسكار ايطالى ٩٠/٦٠ بالامان اللوان - ٥ شعلة</p>
---	---

موافق كود الصنف

الوصول من خلال القائمة
 الوصول من خلال الكود

 بيانات الاصناف

الاسعار	550.00	السعر التقسيطى (جنيه)	550.00	سعر البيع (جنيه)
	<input type="text"/>	عدد القطع فى الجملة	<input type="text"/>	سعر الجملة للقطعة (جنيه)

شاشة الادارة الكاملة للاصناف

يمكن من خلالها اضافة الاصناف وتعديل اسعارها وتحديد الواردات (الكميات التى اتت الى المخازن + المورد)

تتميز الاصناف داخل البرنامج بانها مقسمة الى مجموعات كل مجموعة بها عدد من الاصناف وهذه احد المميزات البارزة للبرنامج

شاشة الفواتير النقدية (الدفع الفوري)

1305.00	الاجمالي (جنبيه)	18/03/2007	تاريخ الفاتورة	683	رقم الفاتورة
			اسم المشتري		
			شفيق احمد منصور		

إلغاء	%	المبلغ الكلي	المشتراة	الكمية	الجملة	العدد	للجملة	سعر	الجملة	السعر	اسم الصنف
	0	360.00	1	0	0.00	360.00	0	360.00	0	360.00	غسالة - الملكة عادية -
	0	385.00	1	0	0.00	385.00	0	385.00	0	385.00	فرن - الحشاش/٤
	0	210.00	1	0	0.00	210.00	0	210.00	0	210.00	بطانية - ٦ كيلو
	0	350.00	2	0	0.00	175.00	0	175.00	0	175.00	انبوبة - - -

إضافة المشتريات الى الفاتورة

شاشة الوصول السريع للاصناف

الفاتورة الحالية 1

عدد الفواتير 88

سابق ▶ ◀ تالي

▶▶ نهاية ▶ بداية

الصنف

الكمية سعر البيع خصم (%)

الكمية المتوفرة السعر

سعر الجملة عدد القطع في الجملة

إغلاق عادي مميز حذف بحث إضافة

شاشة الفواتير
والتي تتيح استخراج الفواتير الجديدة
ويتميز البرنامج بوجود اكثر من شكل لطباعة الفاتورة
١- شكل مميز ينفرد به البرنامج
٢- شكل عادي كاغلب المحلات التجارية

Report Preview - t_invnw.frx

شركة معرض المستقبل

أمام بنك مصر - ميدان النافورة - برج الحسام - منوف
ت / 0122526127 - 0103995918

فاتورة نقدية - مستخرجة من شركة معرض المستقبل

رقم الفاتورة	683	تاريخ الفاتورة	18/03/2007	الإجمالي (جنيه)	1305.0
اسم المشتري	شفيق احمد منصور				
اسم الصنف	- غسالة - الملكة عادية				
السعر	360.00	السعر للجملة	0.00	عدد القطع في الجملة	0
الكمية المشتراة	1	المبلغ الكلي	360.00	نسبة الخصم (%)	0
اسم الصنف	فرن - الحشاش/؛				
				الاصفي	385.00

شكل مميز

Report Preview - t_invnw.frx

شركة معرض المستقبل

أمام بنك مصر - ميدان النافورة - برج الحسام - منوف
ت / 0122526127 - 0103995918

فاتورة نقدية - مستخرجة من شركة معرض المستقبل

رقم الفاتورة	683	تاريخ الفاتورة	18/03/2007	الإجمالي (جنيه)	1305.0
اسم المشتري	شفيق احمد منصور				

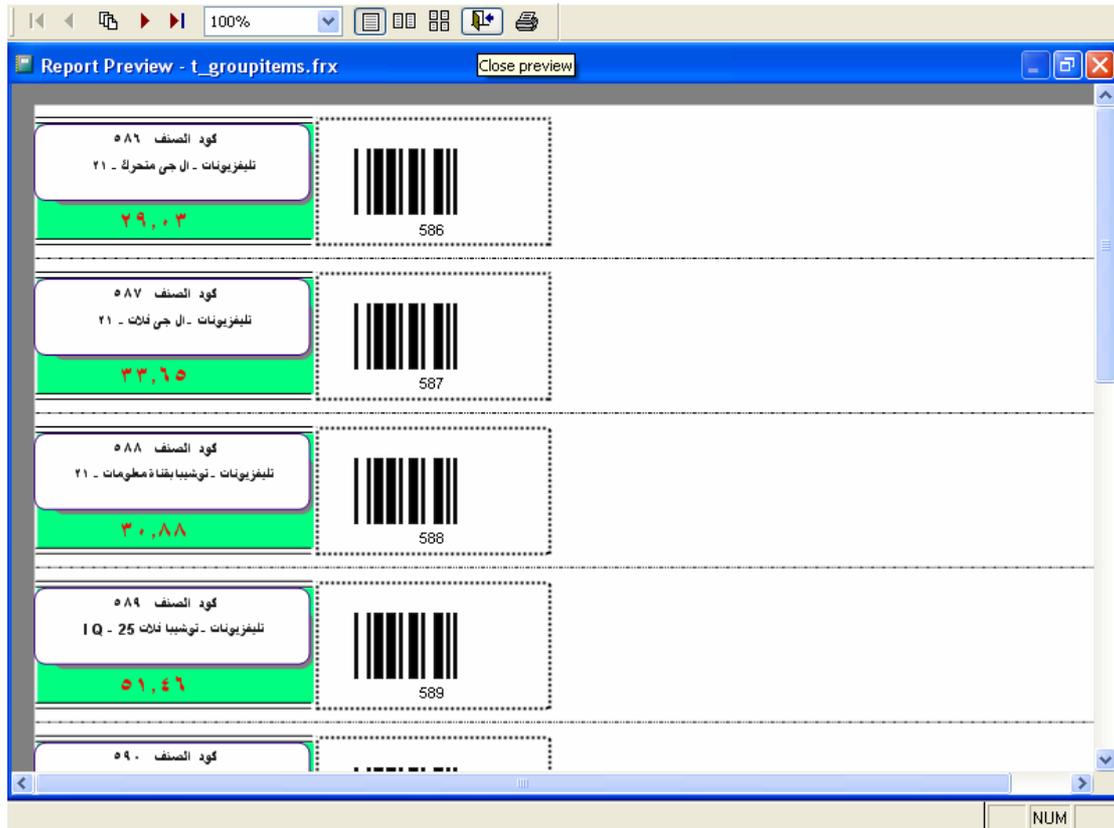
كود الصنف	اسم الصنف	السعر	الكمية	الخصم (%)	الاصفي
545	- غسالة - الملكة عادية	360.00	1	0	360.00
661	فرن - الحشاش/؛	385.00	1	0	385.00
481	بطانية - ٦ كيلو	210.00	1	0	210.00
579	-- التوبية	175.00	2	0	350.00

شكل عادي



شاشة البحث بالاسم

نلاحظ انه يمكن البحث بجزء من الاسم فى اى من بيانات العملاء او الموردين او الموظفين او الفواتير وعن ظهور قائمة الاسماء فانه يمكن اختيار هذا الاسم للوصول الى بقية البيانات



تقرير عن مجموعة من كروت السعر

ونلاحظ ان البرنامج يدعم bar code وهذه اجد مميزاته

Report Preview - t_items2.frx

شركة معرض المستقبل

أمام بنك مصر - ميدان النافورة - برج الحسام - منوف
ت / 0103995918 - 0122526127

قائمة الاصناف ونظم التقسيط

التقسيم - القسط الشهري						الصف		
٨٤	٧٢	٦٠	٤٨	٣٦	٢٤	إسم السلعة المعروضة للبيع		
٨٤	٧٢	٦٠	٤٨	٣٦	٢٤	كاش	الصف	الكود
٢٦,١٩	٢٩,٠٣	٣٢,٠٨	٣٦,٦٧	٤٤,٣٦	٥٩,٥٨	١١٠٠	تليفونات - ال جي متحرك - ٢١	٥٨٦
٣٠,٣٦	٣٣,٦٥	٣٧,١٩	٤٢,٥٠	٥١,٣٥	٦٩,٠٦	١٢٧٥	تليفونات - ال جي فلات - ٢١	٥٨٧
٢٧,٨٦	٣٠,٨٨	٣٤,١٣	٣٩,٠٠	٤٧,١٣	٦٣,٣٨	١١٧٠	تليفونات - نوسيبا بقناة معلومات - ٢١	٥٨٨
٤٦,٤٣	٥١,٤٦	٥٦,٨٨	٦٥,٠٠	٧٨,٥٤	١٠٥,٦٣	١٩٥٠	تليفونات - نوسيبا فلات - 25 - I Q	٥٨٩

تغيير الاصناف - واسعارها - مع نظم التقسيط

Report Preview - t_gain.frx

شركة معرض المستقبل

صافى الايراد

من تاريخ 01/03/2007 الى تاريخ 01/06/2007

رقم الفاتورة	اسم المشتري	الاجمالى (جنيه)
683	شفيق احمد منصور	1305.0
688	د-السيد سمير	255.00
705	عبدالناصر جمعة	175.00
710	نجلاء عبد العال	250.00

تقرير صافى الايراد فى فترة معينة من تاريخ الى تاريخ

Report Preview - t_store.frx

شركة معرض المستقبل

قائمة الاصناف و الكميات - جرد المخازن

الكمية	الاصناف	الكود
المفترض توفرها بالمخازن	اسم السلعة المعروضة للبيع	كود الصنف
1	تليفزيونات - ال جي منحرك - ٢١	586
1	تليفزيونات - ال جي فلات - ٢١	587
1	تليفزيونات - نوسيبا بقناة معلومات - ٢١	588
1	تليفزيونات - نوسيبا فلات - I Q - 25	589
1	تليفزيونات - سامسونج فلات - ٢١	590
1	تليفزيونات - سامسونج فلات - ١٥	591
0	تليفزيونات - ترومان فلات - ٣٤	592
0	تلفونات - سه - فلات - ٢١	605

NUM

تقرير - جرد المخازن

Report Preview - t_comesup.frx

شركة معرض المستقبل

أمام بنك مصر - ميدان النافورة - برج الحسام - منوف
ت / 0103995918 - 0122526127

الواردات تبعا للمورد

المورد شركة الجمال - ش ك

من تاريخ 01/03/2007 الى تاريخ 11/06/2007

كود الصنف	الاصناف	الكمية	تكلفة القطعة	التكلفة الكلية
49	ديب تريزر - الكترولستار - ٤ درج	1	1140.00	1140.00
50	عسالة - اديال زانوسى اقواتيك - تول انومانك ١٨ بروجرام	2	2175.00	4350.00
51	عسالة - فوش - تول انومانك	1	970.00	970.00
53	بوتجاز - يوني فيرسال ٥٥ ٨٠٠ ٥٥ - شطة استالستيل	1	1280.00	1280.00

NUM

الواردات تبعا للمورد

بخصوص برمجة الالعاب

انى فى الواقع احترم مبرمجين الالعاب القدامى لدرجة كبيرة - حيث انه فى العالم القديم - عالم DOS كان معظم مبرمجين الالعاب هم مبرمجين محترفين بلغات المستوى المنخفض مثل اسمبلى ولديهم دراية جيدة بلغة سى C مما يمكنهم من التعامل مع العتاد Hardware بكفاءة وكان ذلك ضرورى للرسم على الشاشة وتشغيل ملفات الصوت والموسيقى و دعم مؤشر الفارة فى البيئة الرسومية.

اما الان فيوجد العديد من ادوات برمجة الالعاب مماسمح للعديد من المبرمجين بغزو هذا المجال - كما اننى اسمع الان عن محاولات عربية جيدة فى سوق الالعاب وهذا اسعدنى كثيرا لان المحاولات الفردية نادرا ماتحقق نتيجة فى برمجة الالعاب التى تستلزم فكر وفن ومهارة وفريق عمل متكامل.

وللالعاب تصنيفات عديدة فمثلا هناك

✚ 2D Games

✚ 3D Games

وهناك

✚ Computer Games

✚ Micro Games

وهناك

- Action Games
تتميز بالسرعة - والصور تكون حقيقية بقدر الامكان - والصوت يكون متنوع - ودعم اكثر من لاعب هدف حيوى جدا
- Combat Games
تشمل هذه النوعية عادة شخصيتين يتصارعان معا - كل منهما يحاول هزيمة الاخر
- Adventure Games—
تحاول هذه الالعاب عرض عالم حتى يستكشفه اللاعب وغالبا ما تعتمد على قصة مشهورة
- Puzzle Games
تتطلب من اللاعب استخدام المنطق والذكاء للوصول للحل
- Strategy Games

تتطلب هذه الالعب التخطيط والنظر للمستقبل

- Role Playing Games (RPG)
العب المهام والادوار - حيث تقوم بمهمة حتى تفوز
- Simulation Games
العب المحاكاة
- Trivia Games
العب اختبار المعرفة - عن طريق الاسئلة - مثل من سيربح المليون
- Word Games
العب الكلمات - مثل الكلمات المتقاطعة
- Card Games
العب الكروت (شوكر - ولد - بنت - شايب - الارقام ١ الى ١٠)
- Games of Chance
العب الحظ
- Sports Games
العب الرياضات المختلفة
- Toys
هذه الالعب نادرة الوجود لكنها هامة وملفتة للانتباه - انها الالعب التي لاتشمل فائز او خاسر وانما توفر للمستخدم فكرة التحكم بادوات مثل (عزف الموسيقى) - (بناء المباني) - (تحريك الانسان الالى) وهكذا.

وبخصوص مراحل تصميم الالعب

✚ Preproduction

وهي مرحلة ماقبله الانتاج لتحديد ما سيتم فعله وتشمل الاجابة على الاسئلة التالية
ما سمات ونوع اللعبة ؟
ما هي قيود اللعبة ؟
ما هي المهمة المركزية للعبة ؟
ما هي المدخلات وما هي المخرجات ؟
من سوف يلعب اللعبة ؟

✚ Prototyping

✚ Programming

✚ Play testing

ولعنا نذكر شيئاً هاماً – ان هناك ألعاب يتم برمجتها أولاً – بدون جرافيك أو صوت – فقط من أجل عمل اختبار لميكانيكية برمجة اللعبة

انظر الشكل التالي وهو عبارة عن نموذج مبسط للعبة من برمجة مؤلف الكتاب وقد تم برمجتها باستخدام نمط البرمجة DoubleS وكان الهدف من البرمجة هو اختبار نظام إدارة الأحداث Event-Driven System حيث انه أحد الدعائم الرئيسية التي تركز عليها برمجة الألعاب – وحيث ان اللعبة ليست بغرض الانتاج فهي لا تحتوي على جرافيك أو تأثيرات صوتية.



وفيما يلي الشيفرة المصدرية التي كتبها المؤلف.

```
*-----*
*
* http://www.sourceforge.net/projects/doublesvscoop
*
* DoubleS Project (DoubleS Paradigm, DoubleS Framework & DoubleS Library)
*
* Copyright(C) 2006-2007, Mahmoud Fayed (msfclipper@hotmail.com)
*
*-----*

#include "SSXHB.CH"          && DoubleS Library Header File

SET PROCEDURE TO SSLIB.PRG  && DoubleS Library Source Code

FUNCTION MAIN()

START DOUBLES              && One Time At the start of Application
New Server EarthWolf Type Slave Server Eigen Value 000
```

Details:

```
DataUnit:          && Data Unit Not Used Here
  ATOM KILLER
    SHELL MOVE
      SUBSHELL PLACE
        VAR  V_STEP
        VAR  V_TIME
        VAR  V_DIR
        VAR  V_BOOMS
        VAR  V_BTIME
        VAR  V_NBTIME
```

```
  ATOM HUMAN
    SHELL MOVE
      SUBSHELL PLACE
        VAR  SYS_KEYBOARD
        VAR  V_X
        VAR  V_Y
        VAR  V_HDIR
        VAR  V_HTIME
        VAR  V_GOP1
        VAR  V_GOP2
        VAR  V_X2
        VAR  V_Y2
        VAR  V_HDIR2
        VAR  V_HTIME2
```

```
CodeUnit:
  MAIN RESISTANCE CIRCUITS\C1\B3\MYMAIN
  Circuit C1
  MainSwitch      On
    Branch B1
      Switch      On
        Resistance KeyBuffer
        Resistance NewBoom
        Resistance MoveBoom
        Resistance MoveKiller
        Resistance CheckExit
    Branch Human1
      Switch      On
        Resistance MoveHuman1
        Resistance KeyHuman1
    Branch Human2
      Switch      On
        Resistance MoveHuman2
        Resistance KeyHuman2
    Branch B2
      Switch      On
        Resistance GameOver
    Branch B3
      Switch Off
        Resistance mymain
        Resistance DrawKiller
        Resistance DrawHuman
        Resistance DrawHuman2
        Resistance DrawBOOMS
```

```
VetoUnit:          && Veto Unit Not Used Here
```

End Of Server

```
Resistance Start()      Address Code Unit : Circuits\c1\b1\start
Resistance KeyBuffer()  Address Code Unit : Circuits\c1\b1\KeyBuffer
Resistance NewBoom()    Address Code Unit : Circuits\c1\b1\NewBoom
Resistance MoveBoom()   Address Code Unit : Circuits\c1\b1\MoveBoom
Resistance MoveHuman1() Address Code Unit : Circuits\c1\human1\MoveHuman1
Resistance MoveHuman2() Address Code Unit : Circuits\c1\human2\MoveHuman2
Resistance MoveKiller() Address Code Unit : Circuits\c1\b1\MoveKiller
Resistance KeyHuman1()  Address Code Unit : Circuits\c1\human1\KeyHuman1
Resistance KeyHuman2()  Address Code Unit : Circuits\c1\human2\KeyHuman2
Resistance CheckExit()  Address Code Unit : Circuits\c1\b1\CheckExit

Resistance GameOver()   Address Code Unit : Circuits\c1\b2\GameOver
Resistance mymain()     Address Code Unit : Circuits\c1\b3\mymain
Resistance drawkiller() Address Code Unit : Circuits\c1\b3\drawkiller
Resistance drawhuman()  Address Code Unit : Circuits\c1\b3\drawhuman
Resistance drawh2()     Address Code Unit : Circuits\c1\b3\drawhuman2
Resistance drawBOOMS()  Address Code Unit : Circuits\c1\b3\drawBOOMS
```

```

Server FireOn

RETURN

Resistance NewBoom Code
IF SECONDS() - V_NBTIME >= 1
V_NBTIME = SECONDS()
Y = int(V_STEP * 2.5)
X = int(V_STEP * 8)
Y2 = int((9-V_STEP) * 2.5)
X2 = int((9-V_STEP) * 8)
AADD(V_BOOMS,{Y2,6,4})
AADD(V_BOOMS,{3,X2,3})
AADD(V_BOOMS,{21,X,2})
AADD(V_BOOMS,{Y,73,1})
ENDIF
End of Resistance

Resistance MoveBoom Code
IF SECONDS() - V_BTIME >= 0.1
V_BTIME = SECONDS()
call resistance circuits\c1\b3\drawBOOMS(.t.)
if len(v_booms) > 0
for x = 1 to len(V_booms)
IF V_BOOMS[X][3] = 1 && LEFT
V_BOOMS[X][2] = V_BOOMS[X][2] - 1
ENDIF
IF V_BOOMS[X][3] = 2 && UP
V_BOOMS[X][1] = V_BOOMS[X][1] - 1
ENDIF
IF V_BOOMS[X][3] = 3 && DOWN
V_BOOMS[X][1] = V_BOOMS[X][1] + 1
ENDIF
IF V_BOOMS[X][3] = 4 && RIGHT
V_BOOMS[X][2] = V_BOOMS[X][2] + 1
ENDIF
next
for x = 1 to len(V_booms)
IF V_BOOMS[X][2] < 6
ADEL(V_BOOMS,X)
ASIZE(V_BOOMS,LEN(V_BOOMS)-1)
LOOP
ENDIF
IF V_BOOMS[X][2] > 73
ADEL(V_BOOMS,X)
ASIZE(V_BOOMS,LEN(V_BOOMS)-1)
LOOP
ENDIF
IF V_BOOMS[X][1] < 3
ADEL(V_BOOMS,X)
ASIZE(V_BOOMS,LEN(V_BOOMS)-1)
LOOP
ENDIF
IF V_BOOMS[X][1] > 21
ADEL(V_BOOMS,X)
ASIZE(V_BOOMS,LEN(V_BOOMS)-1)
LOOP
ENDIF
NEXT
call resistance circuits\c1\b3\drawBOOMS()
endif
call resistance Circuits\c1\b2\GameOver
ENDIF
End of Resistance

Resistance MoveHuman1 Code
IF SECONDS() - V_HTIME >= 0.04
V_HTIME = SECONDS()
if v_HDIR >= 1 .AND. V_HDIR <= 4
call resistance circuits\c1\b3\drawhuman(.t.)
ENDIF
IF V_HDIR = 1 && left
v_x = v_x - 1
IF V_X < 6

```

```

        V_X = 6
    ENDIF
ENDIF
IF V_HDIR = 2 && up
    v_y = v_y - 1
    IF V_Y < 3
        V_Y = 3
    ENDIF
ENDIF
IF V_HDIR = 3 && down
    v_y = v_y + 1
    IF V_Y > 21
        V_Y = 21
    ENDIF
ENDIF
IF V_HDIR = 4 && right
    v_x = v_x + 1
    IF V_X > 73
        V_X = 73
    ENDIF
ENDIF
if v_HDIR >= 1 .AND. V_HDIR <= 4
    call resistance circuits\c1\b2\drawhuman
ENDIF
call resistance Circuits\c1\b2\GameOver
ENDIF
End of Resistance

Resistance MoveHuman2 Code
IF SECONDS() - V_HTIME2 >= 0.04
    V_HTIME2 = SECONDS()
    if v_HDIR2 >= 1 .AND. V_HDIR2 <= 4
        call resistance circuits\c1\b3\drawhuman2(.t.)
    ENDIF
IF V_HDIR2 = 1 && left
    v_x2 = v_x2 - 1
    IF V_X2 < 6
        V_X2 = 6
    ENDIF
ENDIF
IF V_HDIR2 = 2 && up
    v_y2 = v_y2 - 1
    IF V_Y2 < 3
        V_Y2 = 3
    ENDIF
ENDIF
IF V_HDIR2 = 3 && down
    v_y2 = v_y2 + 1
    IF V_Y2 > 21
        V_Y2 = 21
    ENDIF
ENDIF
IF V_HDIR2 = 4 && right
    v_x2 = v_x2 + 1
    IF V_X2 > 73
        V_X2 = 73
    ENDIF
ENDIF
if v_HDIR2 >= 1 .AND. V_HDIR2 <= 4
    call resistance circuits\c1\b3\drawhuman2
ENDIF
call resistance Circuits\c1\b2\GameOver
ENDIF
End of Resistance

Resistance MoveKiller Code
IF SECONDS() - V_TIME >= 0.2
    V_TIME = SECONDS()
    IF V_DIR = 1
        V_STEP = V_STEP + 1
    ELSE
        V_STEP = V_STEP - 1
    ENDIF
IF V_STEP = 0
    V_STEP = 1

```

```

        V_DIR = 1
    ENDIF
    IF V_STEP = 9
        V_STEP = 8
        V_DIR = 0
    ENDIF
    set color to bg+/rb
    @0,0 CLEAR TO 24,5
    @0,0 CLEAR TO 2,79
    @22,0 CLEAR TO 24,79
    @0,74 CLEAR TO 24,79
    call resistance Circuits\c1\b3\drawkiller(V_STEP,1)
    call resistance Circuits\c1\b3\drawkiller(V_STEP,2)
    call resistance Circuits\c1\b3\drawkiller(V_STEP,3)
    call resistance Circuits\c1\b3\drawkiller(V_STEP,4)
ENDIF
End of Resistance

Resistance KeyHuman1 Code
x = SYS_KEYBOARD
    if x = 4
        V_HDIR = 4
    endif
    if x = 19
        V_HDIR = 1
    endif
    if x = 5
        V_HDIR = 2
    endif
    if x = 24
        V_HDIR = 3
    endif
End of Resistance

Resistance KeyHuman2 Code
x = SYS_KEYBOARD
    if x = ASC('D') .OR. x = ASC('d')
        V_HDIR2 = 4
    endif
    if x = ASC('A') .or. x = ASC('a')
        V_HDIR2 = 1
    endif
    if x = ASC('W') .or. x = ASC('w')
        V_HDIR2 = 2
    endif
    if x = ASC('X') .or. x = ASC('x')
        V_HDIR2 = 3
    endif
End of Resistance

Resistance KeyBuffer Code
SYS_KEYBOARD = inkey()
end of resistance

Resistance CheckExit Code
x = SYS_KEYBOARD
if x = 27
    set color to w/n
    clear
    set cursor on
    quit
endif
End Of Resistance

Resistance GameOver Code
IF .NOT. LEN(V_BOOMS) = 0
    FOR X = 1 TO LEN(V_BOOMS)
        IF V_BOOMS[X][1] = V_Y .AND. V_BOOMS[X][2] = V_X
            SAVE SCREEN TO OLDSCR
            SET COLOR TO W/R
            CLEAR
            @12,30 SAY "Game Over Player 1 "
            sleeptime 1
            RESTORE SCREEN FROM OLDSCR
            WITH CIRCUIT C1
        ENDIF
    ENDFOR
ENDIF

```

```

        WITH BRANCH HUMAN1
        SET SWITCH OFF
    END WITH BRANCH
END WITH CIRCUIT
V_X = 0
V_Y = 0
V_GOP1 = .T.
P1 = 0
P2 = 0
FOR Y = 1 TO LEN(MYEARTH)
    FOR T = 1 TO LEN(MYEARTH[Y])
        IF MYEARTH[Y][T] = 1
            P1 = P1 + 1
        ENDIF
        IF MYEARTH[Y][T] = 2
            P2 = P2 + 1
        ENDIF
    NEXT
NEXT
ENDIF
IF V_BOOMS[X][1] = V_Y2 .AND. V_BOOMS[X][2] = V_X2
    SAVE SCREEN TO OLDSCR
    SET COLOR TO W/R
    CLEAR
    @12,30 SAY "Game Over Player 2 "
    sleeptime 1
    WITH CIRCUIT C1
        WITH BRANCH HUMAN2
        SET SWITCH OFF
    END WITH BRANCH
END WITH CIRCUIT
RESTORE SCREEN FROM OLDSCR
V_X2 = 0
V_Y2 = 0
V_GOP2 = .T.
P1 = 0
P2 = 0
FOR Y = 1 TO LEN(MYEARTH)
    FOR T = 1 TO LEN(MYEARTH[Y])
        IF MYEARTH[Y][T] = 1
            P1 = P1 + 1
        ENDIF
        IF MYEARTH[Y][T] = 2
            P2 = P2 + 1
        ENDIF
    NEXT
NEXT
ENDIF
NEXT
ENDIF
if V_GOP1 = .T. .and. V_GOP2 = .T.
    P1 = 0
    P2 = 0
    FOR X = 1 TO LEN(MYEARTH)
        FOR T = 1 TO LEN(MYEARTH[X])
            IF MYEARTH[X][T] = 1
                P1 = P1 + 1
            ENDIF
            IF MYEARTH[X][T] = 2
                P2 = P2 + 1
            ENDIF
        NEXT
    NEXT
    set color to w/w
    clear
    set color to bg+/gr,gr+/gr
    @ 2,0 clear to 6,79
    @3,45 say "          (SCORE)"
    @4,30 say " PLAYER [1] "
    @4,45 say STR(P1)
    @5,30 say " PLAYER [2] "
    @5,45 say STR(P2)
    set color to bg+/B,gr+/b
    @ 8,0 clear to 14,79
    @10,40 say " Try Again "

```

```

@12,40 say " Exit "
set color to b/B,gr+/b
@10,30 prompt "|||||"
@12,30 prompt "|||||"
menu to cont
if cont = 1
    V_GOP1 = .f.
    V_GOP2 = .f.
    WITH CIRCUIT C1
        WITH BRANCH HUMAN2
            SET SWITCH ON
        END WITH BRANCH
    END WITH CIRCUIT
    WITH CIRCUIT C1
        WITH BRANCH HUMAN1
            SET SWITCH ON
        END WITH BRANCH
    END WITH CIRCUIT
    release all
    call resistance circuits\c1\b3\mymain
else
    SYS_KEYBOARD = 27
    call resistance Circuits\c1\b1\CheckExit
endif
ENDIF
End Of Resistance

Resistance mymain Code
set color to w/w
clear
set color to bg+/gr,gr+/gr
@ 2,0 clear to 6,79
    @3,17 SAY "   Doubles (Super Server) Programming Paradigm "
    @4,17 SAY "   Copyright(C) 2006-2007, Mahmoud Fayed "
    @5,17 SAY "   Simple Game : EarthWolf v1.0 "
set color to bg+/B,gr+/b
@8,0 clear to 16,79
@10,30 say " One Player "
@12,30 say " Two Players W=Up D=Right A= Left X=Down"
@14,30 say " Exit "
set color to b/B,gr+/b
@10,20 prompt "|||||"
@12,20 prompt "|||||"
@14,20 prompt "|||||"
menu to mainmenu
if mainmenu = 3
    set color to w/n
    clear
    set cursor on
    quit
endif
SET COLOR TO W/B
CLEAR
SET COLOR TO BG+/RB
call resistance Circuits\c1\b3\drawkiller(5,1)
call resistance Circuits\c1\b3\drawkiller(5,2)
call resistance Circuits\c1\b3\drawkiller(5,3)
call resistance Circuits\c1\b3\drawkiller(5,4)
SET CURSOR OFF
SELECT ADDRESS KILLER:MOVE:PLACE
LOAD ACTIVE SUBSHELL FROM MEMORY
V_TIME = SECONDS()
V_STEP = 5
V_DIR = 1
V_BOOMS = {}
V_BTIME = SECONDS()
V_NBTIME = SECONDS()
SELECT ADDRESS HUMAN:MOVE:PLACE
LOAD ACTIVE SUBSHELL FROM MEMORY
V_X = 40
V_Y = 10
V_HDIR = 0
V_HTIME = SECONDS()
V_GOP1 = .F.
if .not. mainmenu = 1

```

```

V_GOP2 = .F.
V_X2 = 39
V_Y2 = 10
V_HDIR2 = 0
V_HTIME2 = SECONDS()
Call resistance circuits\c1\b3\drawhuman2
else
    WITH CIRCUIT C1
        WITH BRANCH HUMAN2
            SET SWITCH OFF
        END WITH BRANCH
    END WITH CIRCUIT
    V_GOP2 = .t.
    V_X2 = 0
    V_Y2 = 0
endif
PUBLIC MYEARTH
MYEARTH = {}
SYS_KEYBOARD = 0
Call resistance circuits\c1\b3\drawhuman
* 3 6 21 73
FOR X = 3 TO 21
AADD(MYEARTH, {0,0,0,0,0,0,0,0,0,0,;
               0,0,0,0,0,0,0,0,0,0,;
               0,0,0,0,0,0,0,0,0,0,;
               0,0,0,0,0,0,0,0,0,0,;
               0,0,0,0,0,0,0,0,0,0,;
               0,0,0,0,0,0,0,0,0,0,;
               0,0,0,0,0,0,0,0,0,0,;
               0,0,0})
NEXT
End of Resistance

Resistance drawkiller Code
parameters step,place
local myform,myform2,myform3,myform4
set color to bg+/rb

if place = 1 .or. place = 4
step2 = 2.5 * (9-step)
step = 2.5 * step
else
step2 = 8 * (9-step)
step = 8 * step
endif

myform = {"±±   ",;
          "±±±±³-",;
          "±±   "}

myform2= {"±±±±±",;
          " ±³± ",;
          " | "}

myform3= {" | ",;
          " ±³± ",;
          "±±±±±"}

myform4 ={"   ±±",;
          "-³±±±±",;
          "   ±±  "}

if place = 1 && left
    @step2,0 say myform[1]
    @step2+1,0 say myform[2]
    @step2+2,0 say myform[3]
endif

if place = 2 && up
    @0,step2 say myform2[1]
    @1,step2 say myform2[2]
    @2,step2 say myform2[3]
endif

```

```

if place = 3 && down
  @22,step say myform3[1]
  @23,step say myform3[2]
  @24,step say myform3[3]
endif

if place = 4 && right
  @step,74 say myform4[1]
  @step+1,74 say myform4[2]
  @step+2,74 say myform4[3]
endif

End of Resistance

Resistance drawhuman Code
parameters phide
local myform,myform2,myform3,myform4
set color to bg+/b
if phide = .t.
  @v_y,v_x say "+"
  if .not. v_x = 0 .and. .not. v_y = 0
    MYEARTH[V_Y-2][V_X-5] = 1
  endif
else
  myform = "[]"
  myform2 = "--"
  myform3 = ""
  myform4 = "[]"
  if v_hdir = 1 && left
    @v_y,v_x say myform4
  endif
  if v_hdir = 2 && up
    @v_y,v_x say myform2
  endif
  if v_hdir = 3 && down
    @v_y,v_x say myform3
  endif
  if v_hdir = 4 .or. v_hdir = 0 && right
    @v_y,v_x say myform
  endif
endif
End of Resistance

Resistance drawh2 Code
parameters phide
local myform,myform2,myform3,myform4
set color to gr+/b
if phide = .t.
  if .not. v_x2 = 0 .and. .not. v_y2 = 0
    @v_y2,v_x2 say "+"
    MYEARTH[V_Y2-2][V_X2-5] = 2
  endif
else
  myform = "[]"
  myform2 = "--"
  myform3 = ""
  myform4 = "[]"
  if v_hdir2 = 1 .or. v_hdir2 = 0 && left
    @v_y2,v_x2 say myform4
  endif
  if v_hdir2 = 2 && up
    @v_y2,v_x2 say myform2
  endif
  if v_hdir2 = 3 && down
    @v_y2,v_x2 say myform3
  endif
  if v_hdir2 = 4 && right
    @v_y2,v_x2 say myform
  endif
endif
End of Resistance

Resistance drawbooms Code

```

```

parameters phide
local x
set color to n/b
if phide = .t.
  if len(v_booms) > 0
    for x = 1 to len(V_booms)
      myy = v_booms[x][1] - 2
      myx = v_booms[x][2] - 5
      if myy >= 1 .and. myx >= 1
        if MYEARTH[myy][myx] = 0
          @v_booms[x][1],v_booms[x][2] say " "
        endif
        if MYEARTH[myy][myx] = 1
          set color to bg+/b
          @v_booms[x][1],v_booms[x][2] say "+ "
        endif
        if MYEARTH[myy][myx] = 2
          set color to gr+/b
          @v_booms[x][1],v_booms[x][2] say "+ "
        endif
      endif
    next
  endif
else
  if len(v_booms) > 0
    for x = 1 to len(V_booms)
      @v_booms[x][1],v_booms[x][2] say "ع"
    next
  endif
endif
End of Resistance

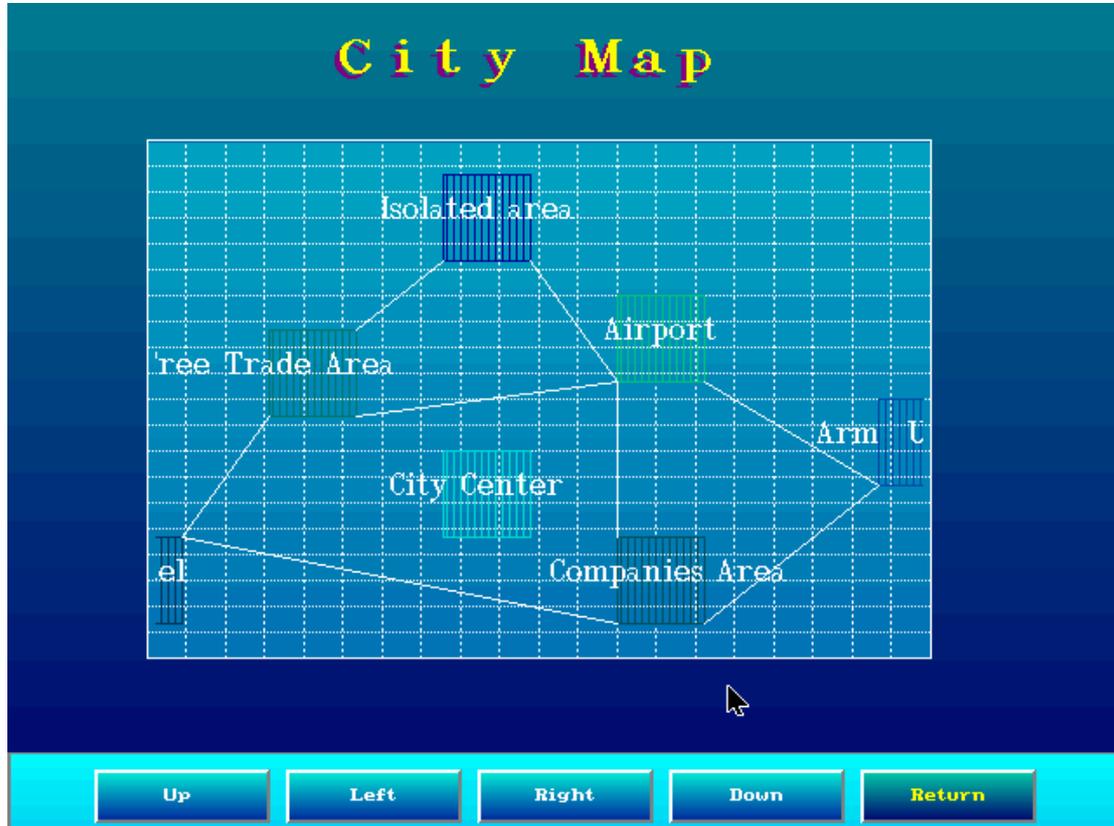
```

وحيث ان برمجة الالعاب تعد فى غاية المتعة لانك تعوض التعب الذى تبذله فى البرمجة من خلال اختبارك للعبة وذلك لا يتم الا اذا لعبت بها - وتلك متعة لا توصف

وقد كنت اهوى تطوير الالعاب الاستراتيجية المبنية على التفكير بدلا من الاكشن - ومن المحاولات الجيدة التى قمت بعملها - لعبة صغيرة الحجم - تحت DOS بعنوان Love World 1.0 وتقوم فكرة اللعبة انك اثناء سفرك مع صديقتك (زوجتك) واثناء العودة وبالتحديد قبل نصف الساعة من الرحيل اكتشفت انها غير موجودة وقد كنتم مقيمين فى فندق - ومعك ١٠٠٠ دولار ويجب عليك ان تحضرها فى اقل من نصف ساعة ولديك موبايل يمكنك استخدامه فى الاتصال باحد الارقام المسجلة بمفكرة الهاتف والتى تشمل رقم اصدقائك وصاحب الفندق وضابط الشرطة وسائق التاكسى - ومعك ايضا خريطة المدينة حيث يمكنك التنقل فيها وانت على علم بها.

وقد قمت ببرمجة اللعبة باستخدام لغة كليبر بالاضافة الى المكتبة DGE 5.0 لدعم الجرافك - اما الواجهة الرسومية GUI الخاصة باللعبة فقد برمجتها بنفسى قبل البدء فى برمجة اللعبة

- وهى لعبة بسيطة - من الناحية البرمجية - لعل الجزء الفنى الوحيد فيها هى الخريطة التى يمكنك التحرك فيها بالفأرة ولوحة المفاتيح.



Telephone Note

Name	Telephone
[1] - Hotel manager	1234567
[2] - Taxi driver	8901234
[3] - Police officer	4242111
[4] - Eman	2713350
[5] - Fady	0843352

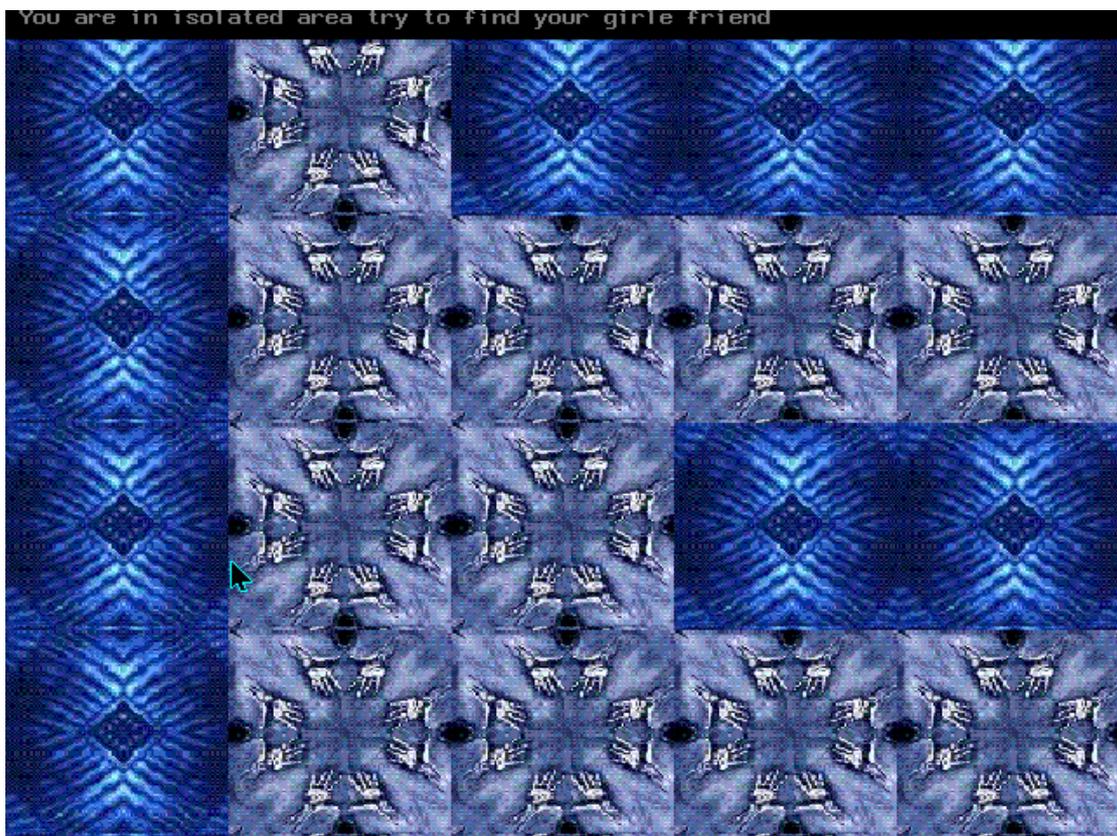
Return

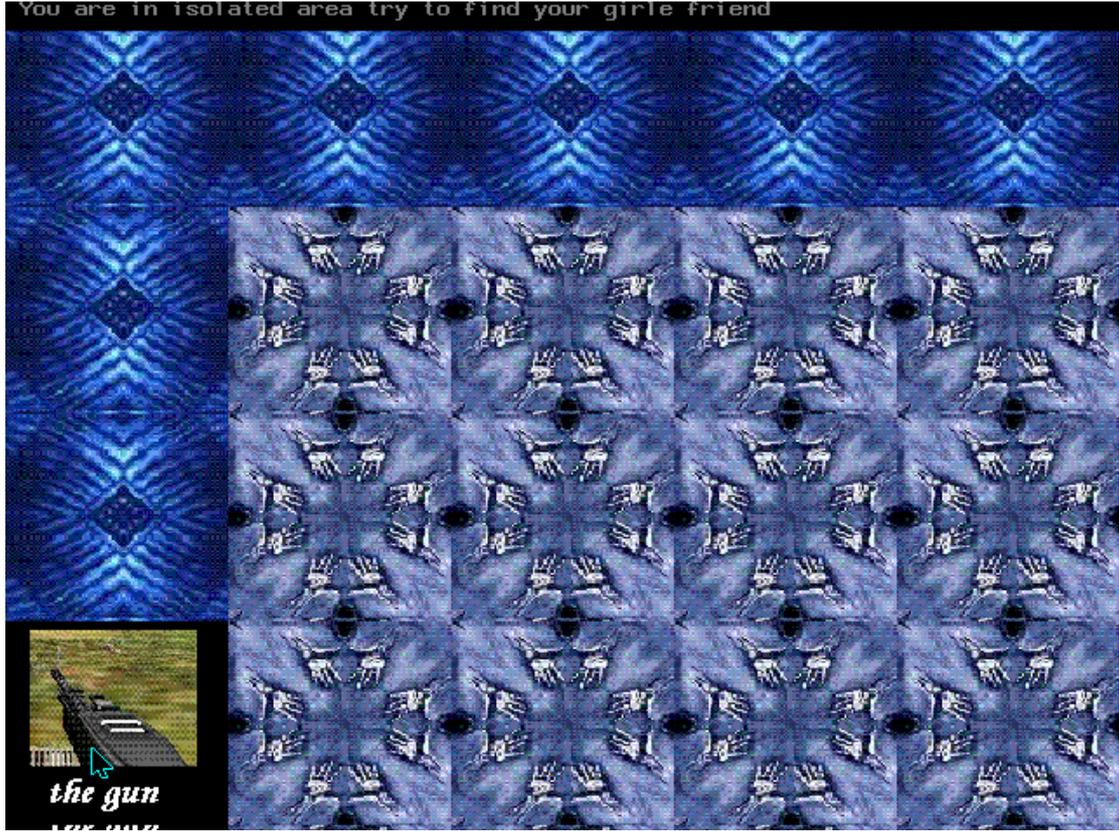
Mobile Screen

Number :8901234

1	2	3	
4	5	6	
7	8	9	0

On Off Call Clear Return





ولعلنا نستنتج من تلك الجولة السريعة متطلبات تطبيقات الالعاب

✚ Event Driven System

نظام ادارة الاحداث - وقد يكون غير موجود
ويتم برمجة الاحداث بدون نظام لادارتها وذلك بصراحة
هو الشائع.

✚ Multi Media Engine

محرك يدعم تعدد الوسائط - بمعنى امكانية تداخل اكثر من
موثر فى التطبيق معا فى نفس الوقت - وهذه المؤثرات هى
(النص - الصور - الصوت - الكلام - الحركة)

✚ Simple Database

ونادر ماتحتاج لعبة قاعدة بيانات بمعنى الكلمة - حيث لا يحدث
ذلك الى فى الالعاب الاستراتيجية المعقدة - وغالبا مايكفى
ملف بيانات بسيط لتسجيل أسماء اللاعبين المميزن.

وميكانيكية برمجة الالعاب تكون كالتالى

✚ Paint the screen.

رسم الشاشة

✚ Get any user input.

استقبال الدخل من المستخدم

✚ Make any game state changes.

تغير حالات اللعبة

✚ Redraw the graphics or sounds accordingly.

اعادة رسم الصور ومتابعة الصوت تبعا للتغيرات التي حدثت

ملحوظة

تطورت برمجة الالعب كبقية فروع البرمجيات تطورا كبيرا فى الفترة الاخيرة وتدخلت فى فروع اخرى فى عالم البرمجيات وانتشرت تطبيقات الالعب التي تدعم اكثر من لاعب عبر الشبكة المحلية Local Area Network او عبر الانترنت - وايضا بدأت الالعب المصممة لاجهزة الدقيقة مثل الموبايل فى التالى.



الادوات التي ارشحها لبرمجة الالعب

✚ Python/PyGame

✚ Euphoria Programming Language

✚ C/C++ & Allegro (Game Programming Library)

✚ Special version of J2ME (Java 2 Micro Edition) known as personal Java

عجائب وطرائف

رغم تطور تكنولوجيا برمجة الالعاب وبراعة المبرمجين فى تقديم كل ماهو مثير وجديد - الا اننا نفاجىء بان بعض الالعاب مثل لعبة الكروت (ولد-بنت-شايب-الارقام من ١ الى ١٠) والتي توجد بصورة ورق بلاستيك لم ينتهى عصرها رغم وجودها على الحاسب - بل نجدها تقريبا فى كل منزل - وكذلك لعبة الشطرنج والطاولة - وان كان الشطرنج الالكترونى قد لاقى جاذبية لاتقل عن الشطرنج الخشبى - الا ان لعبة الكروت لم تحقق ذلك.

اتذكر اننى وانا فى العاشرة من العمر قد قمت بتاليف لعبتين باستخدام الكوتشينة (الكروت) البلاستيك - واثناء رحلاتى بين مصر والسعودية عن طريق الباخرة او العبارة او السفينة (جدة - السويس و السويس - جدة) كنت اعلم من اتعرف عليهم هاتين اللعبتين - وكنت رغم فرحى عند السفر (بالطائرة) لان الرحلة ستكون مريحة - الا اننى كنت انزعج لانها ستمر سريعا بدون متعة كما كانت السفينة التى تبخر فى البحر ٤٨ ساعة.

وكان اسم اللعبة الاولى (الخمسة Five) واسم اللعبة الثانية (التسعة Nine)

وقد اعجبت اللعبتين كل من سمع بهما لدحة اننى واثناء تجولى فى السفينة وجدت اربعة لا اعرفهم يلعبون لعبة الخمسة - فقلت لهم (انتم بتلعبم الخمسة !) قالم لى (انت تعرفها) - (قلت لهم طبعا انا الى الفتها) - ولعبت معاهم ولا اذكر اننى قد فزت ابدا !.

العبة الخمسة (Five)

كل لاعب ياخذ خمسة ورقات - لايعرف ماهم - حيث يتم وضعهم على الارض بحيث يكونم غير مكشوفين

يبدأ كل لاعب بالدور (بالترتيب) بان يكشف ورقة من الاوراق الخمسة بحيث ان صادف وكان يوجد ورقة مثلها مكشوفة او مجموع ورقتين او اكثر يطابق الورقة التى كشفها يكسب اللاعب هذه الاوراق مع ورقته التى كشفها
ان كشف اللاعب احد اوراقه وكانت عبارة عن خمسة ياخذ كل الاوراق المكشوفة
اذا انتهى كشف كل الاوراق وبقى اوراق فى الارض وكان هناك لاعب لم يكسب اى ورق - ياخذ هو الورق المكشوف ولكن اذا كان كل اللاعبين قد كسبم ورق و تبقى ورق مكشوف يرجع الى بقية الورق من جديد بحيث يوزع على كل لاعب خمسة اوراق مرة اخرى
الفوز فى اللعبة من يجمع اكبر قدر من الاوراق

العبة التسعة (Nine)

كل لاعب ياخذ تسعة ورقات يمسك بهم جميعا فى يده ويتفحصهم جيدا
يتم عمل سحب من بقية الورق بعد توزيع تسعة اوراق على كل لاعب
يتم اخذ ورقة من السحب ووضعها على الارض لتكون بداية اللعبة
الهدف من اللعبة ان يتخلص اللاعب من الورق الذى معه اولا
يمر الدور على اللاعبين بالترتيب
اللاعب الذى عليه الدور - عليه ان يلعب ورقة مثل الورقة التى على الارض او ورقة اقل منها برقم او اكبر منها برقم - مثل لو على الارض (٢) يلعب (١) او (٢) او (٣)
فى حالة عدم وجود ورقة مناسبة مع اللاعب يمكنه ان يسحب ورقة من السحب فان كانت مناسبة يلعبها وان لم يكن - يسحب مرة ثانية وثالثة فان كانت مناسبة يلعبها وان لم يكن يمر الدور من عليه بان يقول Pass - مع امكانية استمراره فى السحب ان رغب فى ذلك

✚ إذا كان اللاعب معه ورقم تحمل الرقم (٩) يمكنها ان يلعبها
 فى اى وقت مهما كانت الصورة التى على الارض ثم فى
 نفس الدور يلعب ورقة اخرى وهكذا يغير مسار اللعبة
 ✚ إذا كان اللاعب معه ثلاثة اوراق متتالية مثل (١) و(٢) و(٣)
 او ثلاثة اوراق متشابهة مثل (١) و (١) و(١) يمكنها ان
 يضعها على الارض حيث انها تحل محل الورقة التى تحمل
 الرقم (٩) وبهذا يغير مسار اللعب.

بخصوص برمجة التطبيقات الرياضية والهندسية

هذا المجال له لغاته وادواته الخاصة مثل Fortran و MATLAB ومع ذلك يمكنك استخدام لغات غير متخصصة فى هذا المجال - ولكن عندها كن حذرا لانك سوف تحتاج الى ان تكون مبرمج فنان مبدع او مبرمج عالم مطلع على الكثير من الالجورزمات الخاصة.

اتذكر ان احد اساتذة الجامعة بقسم الرياضيات طلب من اخى الكبير برنامج لحل مشكلة بسيطة - وقد سالنى اخى ان كنت قد قمت بعمل برنامج شبيه لحل هذه المشكلة (وكان موضوع البرنامج هو Topology) فاجبت بلا - وجاء الفضول - وسالت ماهو هذا الموضوع

قالى لى يكون لديك مجموعة من العناصر وليكن X

$$X = \{ a , b , c \}$$

نريد تكوين مجموعات يطلق عليها Topology بحيث ان

$$T_0 = \{ X , \phi \}$$

ويكون هناك Topologies كثيرة بحيث يضاف الى T_0 مجموعات اخرى مشتقة من المجموعة X بشرط انها تحقق الشروط التالية تقاطع كل عنصرين داخل Topology ينتج مجموعة تنمى لها - واتحاد اى عنصرين داخل Topology ينتمى لها. ومشكلة هذا التطبيق الوحيدة هى استنتاج جميع الاحتمالات من المجموعة X وهذه الاحتمالات هى

$$\{a\} , \{b\} , \{c\} , \{a,b\}, \{a,c\}, \{b,c\}$$

وهى صغيرة فى هذه المجموعة بينما اذا كانت

$$X = \{ a, b, c, d \}$$

فان الاحتمالات تكون

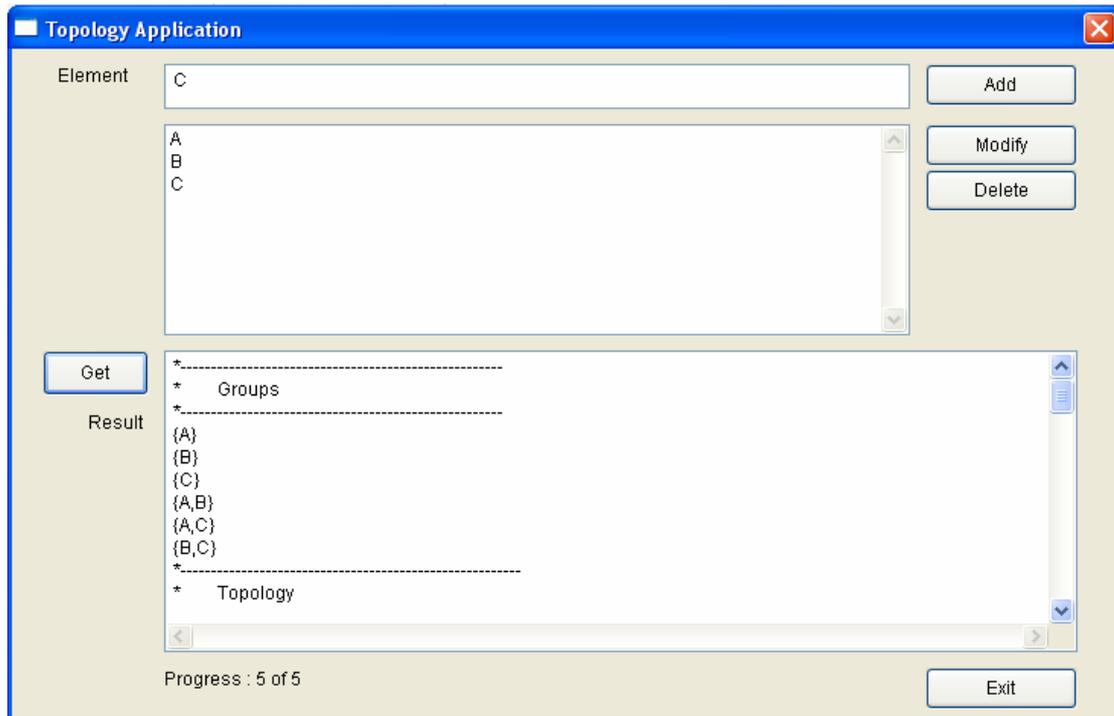
$\{a\}, \{b\}, \{c\}, \{d\}, \{a,b\}, \{a,c\}, \{a,d\}, \{b,c\}, \{b,c\}, \{c,d\}, \{a,b,c\}, \{a,b,d\}, \{b,c,d\}$

ونحن نريد الجورزم يولد هذه المجموعات بالشكل الذى تراه
وحيث ان مسالة الفن متوفرة والحمد لله - فقد تم عمل البرنامج
خلال ساعتين بعد تصميم الالجورزم على الورق

وقد قمت بابتكار هذا الالجورزم - وقد يكون قد سبقنى احد اليه -
انا لا اهتم بذلك - لان هذا لا يعد بحث علمى بالنسبة لى.

قامت فكرة العمل على انى قمت بتعريف Σ مصفوفات
المصفوفة الاولى هى العناصر المكونة للمجموعة X وقد سميت
هذه المصفوفة BASE
والمصفوفة الثانية هى ايضا فى البداية عبارة عن نفس عناصر
المجموعة X وقد سميت هذه المصفوفة REFERENCE
فى البداية يتم وضع جميع عناصر المصفوفة X كنتائج - بحيث
كل عنصر فى مجموعة - وهذا اسهل شىء.
يتم عمل جملة FOR بعدد عناصر المصفوفة X بحيث داخل هذه
الحلقة يتم عمل جملة FOR على جميع عناصر المصفوفة
REFERENCE ومن داخل جملة FOR يتم عمل جملة FOR اخرى
على جميع عناصر المصفوفة BASE لدمج المصفوفتين معا بشرط
الا يحدث تكرار بين عنصر من REFERENCE مع عنصر من BASE
العناصر التى يتم دمجها يتم وضعها فى مصفوفة ثالثة تسمى
RESULT بعد الانتهاء من عملية الجمع - يتم اخذ اخر عنصر فى
المصفوفة RESULT ليكون هو المصفوفة REFERENCE
وبتلك الطريقة نولد جميع الاحتمالات

وفى يلى شكل البرنامج وقد قمت بعمله بلغة HarbourMiniGUI



وقد تم الاختبار على مثال بسيط اختصارا للوقت
ونتيجة التنفيذ

*-----

* Groups

*-----

{A}
{B}
{C}
{A,B}
{A,C}
{B,C}

*-----

* Topology

*-----

T1 = Group + Fay + {A}
T2 = Group + Fay + {B}
T3 = Group + Fay + {C}
T4 = Group + Fay + {A,B}
T5 = Group + Fay + {A,C}
T6 = Group + Fay + {B,C}
T8 = Group + Fay + {A}, {B}
T9 = Group + Fay + {A}, {C}
T10 = Group + Fay + {A}, {A,B}

T11 = Group + Fay + {A}, {A,C}
 T12 = Group + Fay + {A}, {B,C}
 T13 = Group + Fay + {B}, {C}
 T14 = Group + Fay + {B}, {A,B}
 T15 = Group + Fay + {B}, {A,C}
 T16 = Group + Fay + {B}, {B,C}
 T17 = Group + Fay + {C}, {A,B}
 T18 = Group + Fay + {C}, {A,C}
 T19 = Group + Fay + {C}, {B,C}
 T20 = Group + Fay + {A,B}, {A,C}
 T21 = Group + Fay + {A,C}, {B,C}
 T22 = Group + Fay + {A}, {B}, {C}
 T23 = Group + Fay + {A}, {B}, {A,B}
 T24 = Group + Fay + {A}, {B}, {A,C}
 T25 = Group + Fay + {A}, {B}, {B,C}
 T26 = Group + Fay + {A}, {C}, {A,B}
 T27 = Group + Fay + {A}, {C}, {A,C}
 T28 = Group + Fay + {A}, {C}, {B,C}
 T29 = Group + Fay + {A}, {A,B}, {A,C}
 T30 = Group + Fay + {A}, {A,C}, {B,C}
 T31 = Group + Fay + {B}, {C}, {A,B}
 T32 = Group + Fay + {B}, {C}, {A,C}
 T33 = Group + Fay + {B}, {C}, {B,C}
 T34 = Group + Fay + {B}, {A,B}, {A,C}
 T35 = Group + Fay + {B}, {A,C}, {B,C}
 T36 = Group + Fay + {C}, {A,B}, {A,C}
 T37 = Group + Fay + {C}, {A,C}, {B,C}
 T38 = Group + Fay + {A,B}, {A,C}, {B,C}
 T39 = Group + Fay + {A}, {B}, {C}, {A,B}
 T40 = Group + Fay + {A}, {B}, {C}, {A,C}
 T41 = Group + Fay + {A}, {B}, {C}, {B,C}
 T42 = Group + Fay + {A}, {B}, {A,B}, {A,C}
 T43 = Group + Fay + {A}, {B}, {A,C}, {B,C}
 T44 = Group + Fay + {A}, {C}, {A,B}, {A,C}
 T45 = Group + Fay + {A}, {C}, {A,C}, {B,C}
 T46 = Group + Fay + {A}, {A,B}, {A,C}, {B,C}
 T47 = Group + Fay + {B}, {C}, {A,B}, {A,C}

T48 = Group + Fay + {B},{C},{A,C},{B,C}
 T49 = Group + Fay + {B},{A,B},{A,C},{B,C}
 T50 = Group + Fay + {C},{A,B},{A,C},{B,C}
 T51 = Group + Fay + {A},{B},{C},{A,B},{A,C}
 T52 = Group + Fay + {A},{B},{C},{A,C},{B,C}
 T53 = Group + Fay + {A},{B},{A,B},{A,C},{B,C}
 T54 = Group + Fay + {A},{C},{A,B},{A,C},{B,C}
 T55 = Group + Fay + {B},{C},{A,B},{A,C},{B,C}
 T56 = Group + Fay + {A},{B},{C},{A,B},{A,C},{B,C}

والجدير بالذكر ان هذه ال Topologies ليست كلها صحيحة - بل بعضها يخالف الشروط - حيث اننى لم اهتم بكون البرنامج يفحص الشروط - وانما كان التركيز على توليد جميع الاحتمالات فقط - اما بخصوص فحص الشروط فهى عملية بسيطة يمكن اضافتها بسهولة ان توفر الوقت

وفيما يلى الشفيرة المصدرية

```

*-----*
* Topology Application
* 2007 , Mahmoud Fayed
*-----*
#include <minigui.ch>
function main()
load window mainform
activate window mainform
return

function sys_Add()
local myele
myele := alltrim(mainform.text_1.value)
if empty(myele)
msginfo("Please enter element name","sorry")
return
endif
mainform.list_1.additem(myele)
return

function sys_Modify()
LOCAL MYVALUE,myop,myres
MYVALUE = mainform.list_1.value
if .not. myvalue = 0
myop = mainform.list_1.item ( myvalue )
myres = inputbox("Element","Modify",myop)
if .not. empty(alltrim(myres))
mainform.list_1.item ( myvalue ) := myres
endif
endif
endif
  
```

```

return

function sys_Delete()
LOCAL MYVALUE
MYVALUE = mainform.list_1.value
if .not. myvalue = 0
    mainform.list_1.deleteitem ( myvalue )
endif
return

function sys_GO()
local mycount
local x,x2,x3,x4
mycount := mainform.list_1.itemcount
if mycount == 0
msginfo("No Elements","Sorry")
return
endif
myele = {}
mygroups = {}
for x = 1 to mycount
aadd(myele,mainform.list_1.item ( x ))
next
myresult = "*-----" +
chr(13)+chr(10)+;
        "*           Groups           " + chr(13)+chr(10)+;
        "*-----" +
chr(13)+chr(10)
for x = 1 to mycount
aadd(mygroups,"{" + myele[x]+"}")
myresult = myresult +"{" + myele[x]+"}" + chr(13) + chr(10)
next
mybase = {}
myref = {}
myres = {}
for x = 1 to mycount
aadd(mybase,myele[x])
aadd(myref,myele[x])
next
for x1 = 1 to (mycount - 2)
do events
    aadd(myres, {})
    for x2 = 1 to len(myref)
do events
        for x3 = 1 to len(mybase)
do events
            if .not. myref[x2] == mybase[x3]
            if at(alltrim(myref[x2]),alltrim(mybase[x3])) = 0
            if at(alltrim(mybase[x3]),alltrim(myref[x2])) = 0
                mynewe := myref[x2] + "," + mybase[x3]
                mynewe2 := mybase[x3] + "," + myref[x2]
                isold = .f.
                for x4 = 1 to len(myres[x1])
                    if
sys_tt(sys_aarr(alltrim(myres[x1][x4]),sys_aarr( alltrim(mynewe))) =
.t.
                                isold = .t.
                                exit
                    endif
                endfor
            endif
        endif
    endif
enddo
enddo
enddo
endif

```

```

        if
sys_tt(sys_aarr(alltrim(myres[x1][x4])),sys_aarr(alltrim(mynewe2))) =
.t.
                isold = .t.
                exit
        endif
    next
    if isold = .f.
        aadd(myres[x1],mynewe)
    endif
endif
endif
endif
    next
next
myref = {}
for x2 = 1 to len(myres[x1])
    aadd(myref,myres[x1][x2])
next
for x2 = 1 to len(myres[x1])
    aadd(mygroups,"{" + myres[x1][x2] + "}")
    myresult = myresult + "{" + myres[x1][x2] + "}" + chr(13)
+ chr(10)
next
mainform.edit_1.value := myresult
next
myresult = myresult + ;
    "*-----"
+ chr(13)+chr(10)+;
    "*      Topology      " + chr(13)+chr(10)+;
    "*-----"
+ chr(13)+chr(10)
tcount = 1
myele = mygroups
mycount = len(myele)
for x = 1 to mycount
myresult = myresult +"T" + alltrim(Str(tcount)) + " = Group + Fay + "
+ myele[x]+ chr(13) + chr(10)
tcount = tcount + 1
next
mybase = {}
myref = {}
myres = {}
for x = 1 to mycount
aadd(mybase,myele[x])
aadd(myref,myele[x])
next
for x1 = 1 to (mycount - 1)
mainform.label_3.value := "Progress : " + alltrim(str(x1)) + " of " +
alltrim(str(mycount - 1 ))
do events
    aadd(myres, {})
    for x2 = 1 to len(myref)
        do events
            for x3 = 1 to len(mybase)
                do events
                    if .not. myref[x2] == mybase[x3]
                    if at(alltrim(myref[x2]),alltrim(mybase[x3])) = 0
                    if at(alltrim(mybase[x3]),alltrim(myref[x2])) = 0
                    mynewe := myref[x2] + "," + mybase[x3]

```

```

        mynewe2 := mybase[x3] + "," + myref[x2]
        isold = .f.
        for x4 = 1 to len(myres[x1])
            if
sys_tt(sys_aarr(alltrim(myres[x1][x4])),sys_aarr( alltrim(mynewe))) =
.t.
                isold = .t.
                exit
            endif
            if
sys_tt(sys_aarr(alltrim(myres[x1][x4])),sys_aarr(alltrim(mynewe2))) =
.t.
                isold = .t.
                exit
            endif
        next
        if isold = .f.
            aadd(myres[x1],mynewe)
        endif
    endif
    endif
    endif
    next
next
myref = {}
for x2 = 1 to len(myres[x1])
    aadd(myref,myres[x1][x2])
next
for x2 = 1 to len(myres[x1])
    myresult = myresult +"T" + alltrim(str(tcount+1)) + " =
Group + Fay + " + myres[x1][x2] + chr(13) + chr(10)
    tcount = tcount + 1
next
mainform.edit_1.value := myresult
next
return

function sys_aarr(p1) && analysis array
local myarr
myarr = {}
laststart = 1
for t = 1 to len(p1)
if substr(p1,t,1) = ","
aadd(myarr,substr(p1,laststart,t-laststart))
laststart = t+1
endif
next
aadd(myarr,substr(p1,laststart,t-laststart+1))
return myarr

function sys_tt(p1,p2) && test typical between two arrays
v1 = len(p1)
v2 = len(p2)
if v1 != v2
return .f.
endif
for x = 1 to v1
do events
    mye = p1[x]
    isfound = .f.

```

```

for x2 = 1 to v2
do events
    if alltrim(p2[x2]) == alltrim(mye)
        isfound = .t.
        exit
    endif
next
if isfound = .f.
    return .f.
endif
next
return .t.

```

بخصوص برمجة نظم الذكاء الاصطناعي

الجدير بالذكر ان هذه النظم لها طبيعة خاصة - بحاجة الى دراسة بتفرغ - وتحتاج لتطبيقها الى فريق عمل. وهناك طرق تستخدم في حل مشاكل انظمة الذكاء الاصطناعي ومع ذلك يمكن استخدامها في تطبيقات اخرى مثل طرق البحث وتمثيل البيانات. التطبيق البسيط التالي عبارة عن برنامج قمت بكتابته بلغة سي يقوم بتمثيل شجرة داخل ذاكرة الحاسب ومن ثم يمكنه ترتيب هذه الشجرة الترتيب الصحيح حتى يقوم برسمها على الشاشة ويمكنه البحث عن اي عنصر بداخلها.



```
enter root node A
```

```
select the parent  
Item Value := A
```

```
[1] Yes  
[2] No
```

```
2
```

```
Item Value := B
```

```
[1] Yes  
[2] No
```

```
1_
```

```
A      B      C  
      D      E
```

وفيما يلي الشفيرة المصدرية للبرنامج

```
#include "iostream.h"  
#include "stdlib.h"  
#include "stdio.h"
```

```

#include "conio.h"
#include "graph.h"

void mah_say(int myrow,int mycol,char mystr[]);
void mah_clear(int myrow,int mycol,int myrow2,int mycol2);
void enterroot(void);
void enterchild(void);
void cleartree(void);
void drawtree(void);
void depth(void);
void breadth(void);
void newid(void);
void selectparent(void);
void arrangetree(void);

struct
{
    int itemcode;
    char itemvalue[50];
    int parentcode;
    int itemlevel;
} imytree[100],amytree[100];

int arrarr[100]; /* arrange array */

int lastid ;

int parentindex ;

void main(void)
{

    imytree[0].itemcode = 0 ;
    imytree[0].parentcode = 0 ;
    imytree[0].itemlevel = 0 ;

    char mych;
    while (1)
    {
        _setbkcolor(1);
        _clearscreen(_GCLEARSCREEN) ;
        mah_say(2,20,"          -*{ AI Report }*-");

mah_say(3,15,"=====");
        mah_say(4,20,"Mahmoud Samir Ibrahim Fayed : Section 5");

mah_say(5,15,"=====");
        _setbkcolor(3);
        mah_clear(7,22,24,59);
        mah_say(8,27, "[1] Enter Root           ");
        mah_say(10,27,"[2] Enter Child           ");
        mah_say(12,27,"[3] Clear Tree           ");
        mah_say(14,27,"[4] Draw Tree           ");
        mah_say(16,27,"[5] Depth-First         ");
        mah_say(18,27,"[6] Breadth-First        ");
        mah_say(20,27,"[7] Exit                 ");
        mah_say(22,35,"-*{ }*-");
        mych = getch();
        if (mych == '1')
            enterroot();
    }
}

```

```

        if (mych == '2')
            enterchild();
        if (mych == '3')
            cleartree();
        if (mych == '4')
            drawtree();
        if (mych == '5')
            depth();
        if (mych == '6')
            breadth();
        if (mych == '7')
        {
            _setbkcolor(0);
            _clearscreen(_GCLLEARSCREEN) ;
            exit(0);
        }
    }
}

void mah_say(int myrow,int mycol,char mystr[])
{
    _settextposition(myrow,mycol);
    printf(" %s \n",mystr);
}

void mah_clear(int myrow,int mycol,int myrow2,int mycol2)
{
    int x,y;
    for(x = myrow ; x<= myrow2 ; x++)
    {
        _settextposition(x,mycol);
        for( y = mycol ; y<= mycol2 ; y++)
            printf("ġ");
    }
}

void enterroot(void)
{
    _setbkcolor(1);
    _clearscreen(_GCLLEARSCREEN) ;
    cout<< "enter root node ";
    cin>> imytree[0].itemvalue ;
}

void enterchild(void)
{
    selectparent();
    newid();
    _setbkcolor(1);
    _clearscreen(_GCLLEARSCREEN) ;
    cout<< "enter item value ";
    cin>> imytree[lastid].itemvalue ;
    imytree[lastid].itemcode = lastid ;
    imytree[lastid].itemlevel = imytree[parentindex].itemlevel + 1 ;
    imytree[lastid].parentcode = parentindex ;
}

void cleartree(void)
{

```

```

        _setbkcolor(1);
        _clearscreen(_GCLEARSCREEN) ;
        lastid = 0;
        mah_say(12,25,"Now, Tree become Empty !");
        getch();
    }

void drawtree(void)
{
    _setbkcolor(1);
    _clearscreen(_GCLEARSCREEN) ;
    arrangetree();
    int x;
    for(x = 0 ; x<= lastid ; x++)
    {
        mah_say(x,(amytree[x].itemlevel *
10),amytree[x].itemvalue);
    }
    getch();
}

void depth(void)
{
    char mygoal[50];
    _setbkcolor(1);
    _clearscreen(_GCLEARSCREEN) ;
    cout<< " Enter Goal :- " ;
    cin>> mygoal ;
    _setbkcolor(1);
    _clearscreen(_GCLEARSCREEN) ;
    mah_say(0,0," Step          State          Queue");
    int myrow;
    int x,y,z,w,q,q2;
    int arrarr2[100];
    int arrarr3[100];
    int myqueue[100];
    int myq2[100];
    int myq2size ;
    int queuesize ;
    queuesize = 100;
    for (x = 0 ; x<100 ; x++)
    {
        myqueue[x] = -1 ;
        myq2[x] = -1 ;
    }
    arrarr[0] = 0;
    myrow = 2;
    for(x = 0 ; x<= lastid ; x++)
    {
        _settextposition(myrow,4);
        printf("%d",myrow-1);
        mah_say(myrow,23,imytree[arrarr[x]].itemvalue);
        if (imytree[arrarr[x]].itemvalue[0] == mygoal[0] )
            break;

        if (x == 0)
            mah_say(myrow,41,"(Empty)");
        else
        {

```

```

        q2 = 0 ;
        for(q = 99 ; q > 0 ; q-- )
        {
            if (myqueue[q] != -1)
            {
mah_say(myrow,41+(q2*3),imytrees[myqueue[q]].itemvalue);
                q2++;
            }
        }
    }
    // get successors
    z = -1 ;
    for(y = 1 ; y<= lastid ; y++)
    {
        if( imytrees[y].parentcode ==
imytrees[arrarr[x]].itemcode )
        {
            z++;
            arrarr2[z] = y;
        }
    }

    if( z != -1 )
    {
        myrow++;
        _setttextposition(myrow,4);
        printf("%d",myrow-1);
        mah_say(myrow,23,imytrees[arrarr[x]].itemvalue);

        // add elements from top to current element
        for(y = 0 ; y<= x ; y++)
        {
            arrarr3[y] = arrarr[y] ;
        }

        // get count of nodes in queue
        myq2size = 100 - queuesize ;

        // add successors to arranged array

        for(w = 0 ; w<= z ; w++)
        {
            queuesize--;
            myqueue[queuesize] = arrarr2[w];
            arrarr3[w+y] = arrarr2[w] ;
        }

        // war here
        if (myq2size != 0)
        {
            for(w = 0 ; w < myq2size ; w++)
            {
                queuesize--;
                myqueue[queuesize] = myqueue[99-w];
                myqueue[99-w] = -1;
            }
        }
    }
    //end of war

```

```

        q2 = 0 ;
        for(q = 99 ; q > 0 ; q-- )
        {
            if (myqueue[q] != -1)
            {
mah_say(myrow,41+(q2*3),imytree[myqueue[q]].itemvalue);
                q2++;
            }
        }
        // remove first item from queue
        for(q = 99 ; q >= 0 ; q-- )
        {
            if (myqueue[q] != -1)
            {
                myqueue[q] = -1;
                break;
            }
        }
        myrow++;
        z = w+y;
        // ADD ELEMENTS THAT UNDER THE CURRENT ELEMENT
        for(w = x+1 ; w<= lastid; w++)
        {
            arrarr3[z] = arrarr[w] ;
            z++ ;
        }
        // set the new arranged array
        for ( w = 0 ; w <= lastid ; w++ )
        {
            arrarr[w] = arrarr3[w] ;
        }
    }
    for(x = 0 ; x<= lastid ; x++)
    {
        amytree[x] = imytree[arrarr[x]];
    }
    getch();
}

void breadth(void)
{
    char mygoal[50];
    _setbkcolor(1);
    _clearscreen(_GCLEARSCREEN) ;
    cout<< " Enter Goal :- " ;
    cin>> mygoal ;
    _setbkcolor(1);
    _clearscreen(_GCLEARSCREEN) ;
    mah_say(0,0," Step           State           Queue");
    int myrow;
    int x,y,z,w,q,q2;
    int arrarr2[100];
    int arrarr3[100];
    int myqueue[100];
    int myq2[100];
    int myq2size ;
    int queuesize ;
}

```

```

int state ;
state = 0;
queuesize = 100;
for (x = 0 ; x<100 ; x++)
{
    myqueue[x] = -1 ;
    myq2[x] = -1 ;
}
arrarr[0] = 0;
myrow = 2;
for(x = 0 ; x<= lastid ; x++)
{
    _setttextposition(myrow,4);
    printf("%d",myrow-1);
    mah_say(myrow,23,imytrees[state].itemvalue);
    if (imytrees[state].itemvalue[0] == mygoal[0] )
        break;

    if (x == 0)
        mah_say(myrow,41,"(Empty)");
    else
    {
        q2 = 0 ;
        for(q = 99 ; q > 0 ; q-- )
        {
            if (myqueue[q] != -1)
            {
mah_say(myrow,41+(q2*3),imytrees[myqueue[q]].itemvalue);
                q2++;
            }
        }
        // get successors
        z = -1 ;
        for(y = 1 ; y<= lastid ; y++)
        {
            if( imytrees[y].parentcode ==
imytrees[state].itemcode )
            {
                z++;
                arrarr2[z] = y;
            }
        }

        if( z != -1 )
        {
            myrow++;
            _setttextposition(myrow,4);
            printf("%d",myrow-1);
            mah_say(myrow,23,imytrees[state].itemvalue);

            // add elements from top to current element
            for(y = 0 ; y<= x ; y++)
            {
                arrarr3[y] = arrarr[y] ;
            }

            // get count of nodes in queue
            myq2size = 100 - queuesize ;

```

```

        // add successors to arranged array

        for(w = 0 ; w<= z ; w++)
        {
            queuesize--;
            myqueue[queuesize] = arrarr2[w];
            arrarr3[w+y] = arrarr2[w] ;
        }

        q2 = 0 ;
        for(q = 99 ; q > 0 ; q-- )
        {
            if (myqueue[q] != -1)
            {
mah_say(myrow,41+(q2*3),imytrees[myqueue[q]].itemvalue);
                q2++;
            }
        }
        // remove first item from queue
        for(q = 99 ; q >= 0 ; q-- )
        {
            if (myqueue[q] != -1)
            {
                state = myqueue[q];
                myqueue[q] = -1;
                break;
            }
        }
        myrow++;
        z = w+y;
        // ADD ELEMENTS THAT UNDER THE CURRENT ELEMENT
        for(w = x+1 ; w<= lastid; w++)
        {
            arrarr3[z] = arrarr[w] ;
            z++ ;
        }
        // set the new arranged array
        for ( w = 0 ; w <= lastid ; w++ )
        {
            arrarr[w] = arrarr3[w] ;
        }
    }
    for(x = 0 ; x<= lastid ; x++)
    {
        amytrees[x] = imytrees[arrarr[x]];
    }
    getch();
}

void newid(void)
{
    lastid++ ;
}

```

```

void selectparent(void)
{
    int x;
    _setbkcolor(1);
    _clearscreen(_GCLLEARSCREEN) ;
    char mych ;
    cout<< "select the parent \n";
    for(x = 0 ; x<= lastid ; x++)
    {
        cout << "Item Value := "
            << imytree[x].itemvalue;
        cout << "\n\n\n\n";
        cout << " [1] Yes \n " ;
        cout << " [2] No \n " ;
        cin>> mych;
        if (mych == '1' )
            break ;
    }
    parentindex = x;
}

void arrangetree(void)
{
    int x,y,z,w;
    int arrarr2[100];
    int arrarr3[100];
    arrarr[0] = 0;
    for(x = 0 ; x<= lastid ; x++)
    {
        // get successors
        z = -1 ;
        for(y = 1 ; y<= lastid ; y++)
        {
            if( imytree[y].parentcode ==
imytree[arrarr[x]].itemcode )
            {
                z++;
                arrarr2[z] = y;
            }
        }
        // add elements from top to current element
        for(y = 0 ; y<= x ; y++)
        {
            arrarr3[y] = arrarr[y] ;
        }
        // add successors to arranged array
        for(w = 0 ; w<= z ; w++)
        {
            arrarr3[w+y] = arrarr2[w] ;
        }
        z = w+y;
        // ADD ELEMENTS THAT UNDER THE CURRENT ELEMENT
        for(w = x+1 ; w<= lastid; w++)
        {
            arrarr3[z] = arrarr[w] ;
            z++ ;
        }
        // set the new arranged array
    }
}

```

```

        for ( w = 0 ; w <= lastid ; w++ )
        {
            arrarr[w] = arrarr3[w] ;
        }
    }
    for(x = 0 ; x<= lastid ; x++)
    {
        amytree[x] = imytree[arrarr[x]];
    }
}

```

وفيما يلي Dos Version بلغة CA-Clipper

```

C:\ D:\WINDOWS\system32\cmd.exe - ng

AI Report
Author  : Mahmoud Samir Ibrhaim Fayed
Section : 5
-----

Enter Tree Data
Draw Tree
Search using Depth-First Method
Search using Breadth-First Method
Exit

```

– القائمة الرئيسية للبرنامج

```

C:\ D:\WINDOWS\system32\cmd.exe - ng

AI Report
Author  : Mahmoud Samir Ibrhaim Fayed
Section : 5
-----

Enter Tree
Enter Root
Enter Child
Clear Tree
Return

```

– كيفية ادخال عناصر الشجرة من قبل البرنامج

```
D:\WINDOWS\system32\cmd.exe - ng

AI Report
Author : Mahmoud Samir Ibrhaim Fayed
Section : 5
-----

Enter Root : a
```

– ادخال نقطة الاصل

```
D:\WINDOWS\system32\cmd.exe - ng

AI Report
Author : Mahmoud Samir Ibrhaim Fayed
Section : 5
-----

Draw Tree
A
  B
  C
    D
    E
    F
    G
```

– رسم الشجرة مرتبة من قبل البرنامج

```
D:\WINDOWS\system32\cmd.exe - ng

AI Report
Author : Mahmoud Samir Ibrhain Fayed
Section : 5
-----

Depth-First Search
Goal : D_
```

البحث باستخدام Depth-First

```
D:\WINDOWS\system32\cmd.exe - ng

Step      State      Queue
1          A          <empty>
2          A          B,C
3          B          C
4          B          D,E,C
5          D          E,C_
```

نتيجة البحث باستخدام Depth-First Method

```

c:\ D:\WINDOWS\system32\cmd.exe - ng

AI Report
Author : Mahmoud Samir Ibrhaim Fayed
Section : 5
-----

Breadth-First Search

Goal : E_

```

البحث باستخدام Breadth First Search

```

c:\ D:\WINDOWS\system32\cmd.exe - ng

Step      State  Queue
1          A      <empty>
2          A      B,C
3          B      C
4          B      C,D,E
5          C      D,E
6          C      D,E,F,G
7          D      E,F,G
8          D      E,F,G
9          E      F,G

```

نتيجة البحث باستخدام Breadth-First Search

الشيفرة المصدرية Source Code :-

```

*-----*
* AIREP.PRG      *
* 2007, Mahmoud Fayed *
*-----*

set scoreboard off
MYTREE = {}          && MYTREE[N][3]
                    && MYTREE[N][1] = PARENT ID
                    && MYTREE[N][2] = ITEM ID
                    && MYTREE[N][3] = VALUE
                    && MYTREE[N][4] = LEVEL

treeid = 0
AADD(MYTREE, {0,0,"      ",0})
Do While .T.
set color to bg+/b

```

```

clear
@3,3 say " AI Report "
@4,3 say " Author  : Mahmoud Samir Ibrhaim Fayed "
@5,3 say " Section : 5 "
@6,3 say "-----"
set color to bg+/rb
@8,0 clear to 24,79
@10,2 prompt " Enter Tree Data "
@12,2 prompt " Draw Tree "
@14,2 prompt " Search using Depth-First Method "
@16,2 prompt " Search using Breadth-First Method "
@18,2 prompt " Exit "
menu to main
do case
  case main = 1
    entertree()
  case main = 2
    drawtree()
  case main = 3
    dfsearch()
  case main = 4
    bfsearch()
  case main = 5
    set color to w/n
    clear
    quit
endcase
Enddo
*-----*
proc entertree()
*-----*
do while .t.
set color to bg+/b
clear
@3,3 say " AI Report "
@4,3 say " Author  : Mahmoud Samir Ibrhaim Fayed "
@5,3 say " Section : 5 "
@6,3 say "-----"
set color to bg+/r,n/w
@8,0 clear to 24,79
@10,3 say " Enter Tree "
@12,5 prompt " Enter Root "
@14,5 prompt " Enter Child "
@16,5 prompt " Clear Tree "
@18,5 prompt " Return "
menu to etmenu
do case
  case etmenu = 1
    set color to bg+/rb
    @8,0 clear to 24,79
    v_root = MYTREE[1][3]
    @10,3 say " Enter Root : " Get v_root
    read
    MYTREE[1][3] = v_root
  case etmenu = 2
    set color to bg+/rb
    @8,0 clear to 24,79
    v_child = space(10)
    @10,3 say " Enter Child : " Get v_child
    read

```

```

    @12,3 say "Select Parent : "
    myarr = {}
    for x = 1 to len(mytree)
        aadd(myarr,mytree[x][3])
    next
    mych = achoice(14,3,20,70,myarr)
    treeid = treeid + 1

AADD(MYTREE, {mytree[mych][2],treeid,v_child,mytree[mych][4]+1})
    case etmenu = 3
        MYTREE = {}
        AADD(MYTREE,{-1,0,"          ",0})
        SET COLOR TO W/N
        CLEAR
        @12,20 SAY "Now, Tree is empty"
        INKEY(3)
    case etmenu = 4 .OR. etmenu = 0
        return
    endcase
enddo
return
*-----*
*-----*
proc drawtree()
*-----*
SET COLOR TO W/N
CLEAR
MYNEWARR = {}
FOR X = 0 TO TREEID
    FOR T = 1 TO LEN(MYTREE)
        IF MYTREE[T][1] = X

AADD(MYNEWARR, {MYTREE[T][1],MYTREE[T][2],MYTREE[T][3],MYTREE[T][4]})
        ENDIF
    NEXT
NEXT
mydarr = {}
AADD(MYDARR, {MYTREE[1][1],MYTREE[1][2],MYTREE[1][3],MYTREE[1][4]})
X = 1
DO WHILE X <= LEN(MYDARR)
    MYDARR2 = {}
    * ADD ELEMENTS FROM TOP ELEMENT TO CURRENT ELEMENT
    FOR T = 1 TO X

AADD(MYDARR2, {MYDARR[T][1],MYDARR[T][2],MYDARR[T][3],MYDARR[T][4]})
    NEXT
    MYID = MYDARR[X][2]
    * ADD SUCCESSOR
    FOR T = 2 TO LEN(MYTREE)
        IF MYTREE[T][1] = MYID

AADD(MYDARR2, {MYTREE[T][1],MYTREE[T][2],MYTREE[T][3],MYTREE[T][4]})
        ENDIF
    NEXT
    * ADD ELEMENTS THAT UNDER THE CURRENT ELEMENT
    FOR T = X+1 TO LEN(MYDARR)

AADD(MYDARR2, {MYDARR[T][1],MYDARR[T][2],MYDARR[T][3],MYDARR[T][4]})
    NEXT
MYDARR = MYDARR2

```

```

X = X + 1
ENDDO
set color to bg+/b
clear
@3,3 say " AI Report "
@4,3 say " Author : Mahmoud Samir Ibrhaim Fayed "
@5,3 say " Section : 5 "
@6,3 say "-----"
set color to bg+/rb
@8,0 clear to 24,79
@9,0 SAY "Draw Tree"
FOR X = 1 TO LEN(MYDARR)
? SPACE(MYDARR[X][4]*10)
?? MYDARR[X][3]
NEXT
INKEY(0)
return
*-----*
*-----*
proc dfsearch()
*-----*
set color to bg+/b
clear
@3,3 say " AI Report "
@4,3 say " Author : Mahmoud Samir Ibrhaim Fayed "
@5,3 say " Section : 5 "
@6,3 say "-----"
set color to bg+/rb
@8,0 clear to 24,79
@9,0 SAY "Depth-First Search"
myvalue = space(10)
@11,5 say "Goal :" get myvalue
read
mytable = {}
aadd(mytable, {mytree[1][3], "(empty)"} )
myqueue = {}
mystate = mytree[1]
do while .t.
    if mystate[3] = myvalue
        exit
    endif
    MYID = MYstate[2]
    * ADD SUCCESSOR
    MYQUEUE2 = {}
    FOR T = 2 TO LEN(MYTREE)
        IF MYTREE[T][1] = MYID
AADD(MYqueue2, {MYTREE[T][1], MYTREE[T][2], MYTREE[T][3], MYTREE[T][4]})
        ENDIF
    NEXT
    FOR X = 1 TO LEN(MYQUEUE)
        AADD(MYQUEUE2, MYQUEUE[X])
    NEXT
    MYQUEUE = MYQUEUE2
    * record in table
    mystr = ""
    for t = 1 to len(myqueue)
        if .not. empty(alltrim(mystr))
            mystr = mystr + "," + alltrim(myqueue[t][3])
        else

```

```

        mystr = alltrim(myqueue[t][3])
    endif

    next
    aadd(mytable, {mystate[3], mystr})

mystate = myqueue[1]
adel(myqueue, 1)
asize(myqueue, len(myqueue)-1)
    * record in table
    mystr = ""
    for t = 1 to len(myqueue)
        if .not. empty(alltrim(mystr))
            mystr = mystr + "," + alltrim(myqueue[t][3])
        else
            mystr = alltrim(myqueue[t][3])
        endif
    next
    aadd(mytable, {mystate[3], mystr})
enddo

set color to w/n
clear
? "Step"
?? space(10)
?? "State"
?? space(10)
?? "Queue"
for x = 1 to len(mytable)
? x
?? space(10)
?? mytable[x][1]
?? space(10)
?? mytable[x][2]
next
inkey(0)
return
*-----*
*-----*
proc bfsearch()
*-----*
set color to bg+/b
clear
@3,3 say " AI Report "
@4,3 say " Author : Mahmoud Samir Ibrhaim Fayed "
@5,3 say " Section : 5 "
@6,3 say "-----"
set color to bg+/rb
@8,0 clear to 24,79
@9,0 SAY "Breadth-First Search"
myvalue = space(10)
@11,5 say "Goal :" get myvalue
read
mytable = {}
aadd(mytable, {mytree[1][3], "(empty)"})
myqueue = {}
mystate = mytree[1]
do while .t.
    if mystate[3] = myvalue
        exit
    endif
enddo

```

```

endif
MYID = MYstate[2]
* ADD SUCCESSOR
FOR T = 2 TO LEN(MYTREE)
    IF MYTREE[T][1] = MYID
AADD(MYqueue, {MYTREE[T][1],MYTREE[T][2],MYTREE[T][3],MYTREE[T][4]})
        ENDIF
    NEXT
    * record in table
    mystr = ""
    for t = 1 to len(myqueue)
        if .not. empty(alltrim(mystr))
            mystr = mystr + "," + alltrim(myqueue[t][3])
        else
            mystr = alltrim(myqueue[t][3])
        endif

    next
    aadd(mytable, {mystate[3],mystr})

mystate = myqueue[1]
adel(myqueue,1)
asize(myqueue,len(myqueue)-1)
    * record in table
    mystr = ""
    for t = 1 to len(myqueue)
        if .not. empty(alltrim(mystr))
            mystr = mystr + "," + alltrim(myqueue[t][3])
        else
            mystr = alltrim(myqueue[t][3])
        endif

    next
    aadd(mytable, {mystate[3],mystr})
enddo
set color to w/n
clear
? "Step"
?? space(10)
?? "State"
?? space(10)
?? "Queue"
for x = 1 to len(mytable)
? x
?? space(10)
?? mytable[x][1]
?? space(10)
?? mytable[x][2]
next
inkey(0)
return
*-----*

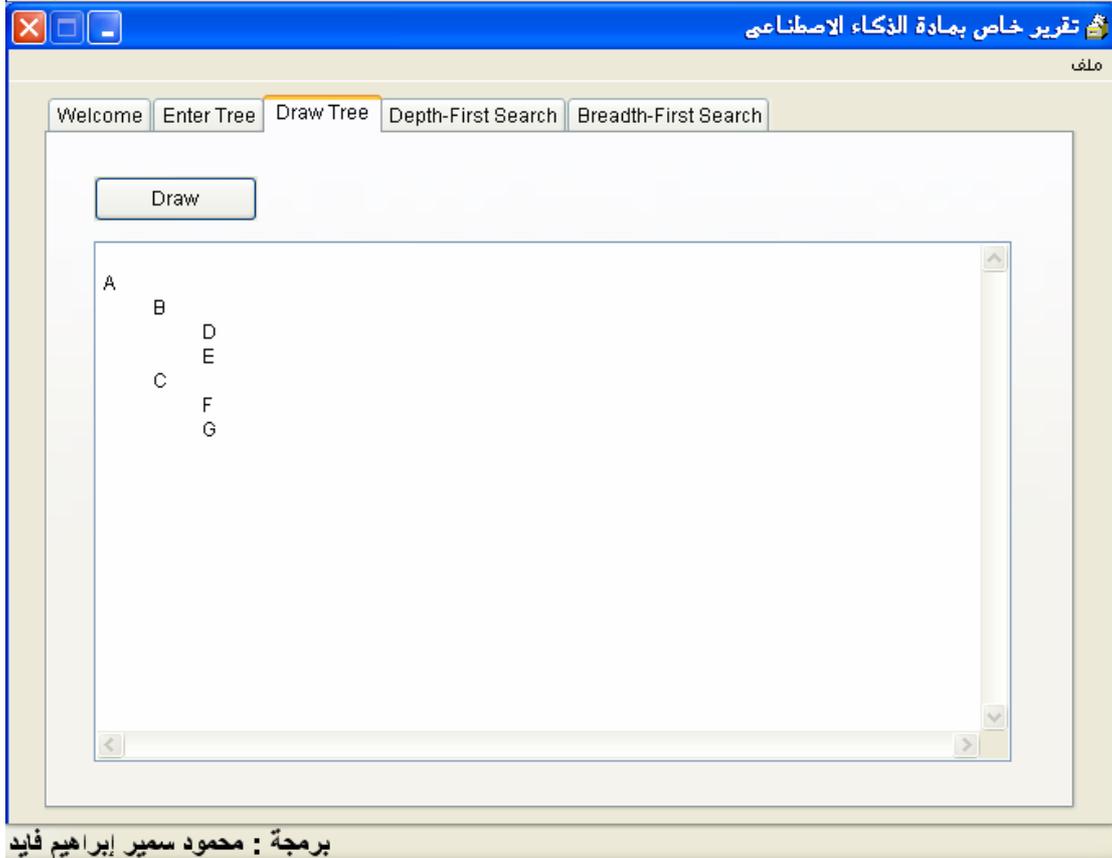
```

Windows Version – GUI واخيرا نسخة مطورة





برمجة : محمود سمير إبراهيم فايد



برمجة : محمود سمير إبراهيم فايد



برمجة : محمود سمير إبراهيم فايد



برمجة : محمود سمير إبراهيم فايد

الشفيرة المصدرية :-

```

* For Harbour/MiniGUI
#include "DoubleSHMG.ch"
#include "minigui.ch"
SET PROCEDURE TO SSLIB.PRG
Function Main
START DOUBLES
*-----*
* This file generated by DoubleS Framework 1.0
* True DoubleS Compiler -> Standard DoubleS Syntax & Statements
* Date : 05/01/07
* Time : 00:08:56
*-----*
New Server AIREP Type Slave Server Eigen Value 000
Details:
  DataUnit :
  CodeUnit :
    Main resistance CIRCUITS\C1\B1\STARTUP
      Circuit      C1
      MainSwitch   On
        Branch     B1
        ParallelTo 0
          Resistance RTL
          Resistance STARTUP
        Branch     B2
        ParallelTo 0
          Resistance SAVE_ROOT
          Resistance FILLLIST
          Resistance addchild
          Resistance removeall
          Resistance removeone
          Resistance DRAWTREE
          Resistance DSEARCH
          Resistance BSEARCH
          Resistance modify
        Branch     B3
        ParallelTo 0
        Switch     On
          Resistance ROOT_CHANGE

  VetoUnit :
End Of Server

  Resistance WIN32EVENTS() Address Code Unit : WIN32EVENTS

  ADD RESISTANCE WIN32EVENTS

Resistance RUZSLC1() address Code Unit : Circuits\C1\B1\RTL
Resistance RUZSLC2() address Code Unit : Circuits\C1\B1\STARTUP
Resistance RUZSLC3() address Code Unit : Circuits\C1\B2\SAVE_ROOT
Resistance RUZSLC4() address Code Unit : Circuits\C1\B2\FILLLIST
Resistance RUZSLC5() address Code Unit : Circuits\C1\B2\addchild
Resistance RUZSLC6() address Code Unit : Circuits\C1\B2\removeall

```

```

Resistance RUZSLC7() address Code Unit : Circuits\C1\B2\removeone
Resistance RUZSLC8() address Code Unit : Circuits\C1\B2\DRAWTREE
Resistance RUZSLC9() address Code Unit : Circuits\C1\B2\DSEARCH
Resistance RUZSLC10() address Code Unit : Circuits\C1\B2\BSEARCH
Resistance RUZSLC11() address Code Unit : Circuits\C1\B3\ROOT_CHANGE
Resistance RUZSLC12() address Code Unit : Circuits\C1\B2\modify

```

```

load window AIFORM
activate window AIFORM

```

```

FUNCTION STARTSERVER()
    Server FireON
RETURN

```

```

RESISTANCE WIN32EVENTS() CODE
    DO EVENTS
RETURN

```

```

* Resistance Code Unit : Circuits\C1\B1\RTL
Resistance RUZSLC1() code

```

```

#define GWL_EXSTYLE -20
#define WS_EX_LAYOUTRTL 0x400000
#define WS_EX_NOINHERITLAYOUT 0x100000
OldLong = callDll32("GetWindowLong","USER32.DLL",GetFormHandle
("AIFORM"),GWL_EXSTYLE)
*-- Set the window extended style RTL non-inheritable.
callDll32("SetWindowLong","USER32.DLL",GetFormHandle
('AIFORM'),GWL_EXSTYLE,;
WS_EX_LAYOUTRTL)
RETURN NIL

```

```

End Of Resistance

```

```

* Resistance Code Unit : Circuits\C1\B1\STARTUP

```

```

Resistance RUZSLC2() code
PUBLIC MYTREE,TREEID
MYTREE = {}          && MYTREE[N][3]
                    && MYTREE[N][1] = PARENT ID
                    && MYTREE[N][2] = ITEM ID
                    && MYTREE[N][3] = VALUE
                    && MYTREE[N][4] = LEVEL

```

```

treeid = 0
AADD(MYTREE,{0,0,"",0})
call resistance circuits\c1\b1\rtl
CENTER WINDOW AIFORM

```

```

End Of Resistance

```

```

* Resistance Code Unit : Circuits\C1\B2\SAVE_ROOT
Resistance RUZSLC3() code
LOCAL MYROOT
MYROOT = AIFORM.TEXT_2.VALUE
MYTREE[1][3] := MYROOT

```

End Of Resistance

```

* Resistance Code Unit : Circuits\C1\B2\FILLLIST
Resistance RUZSLC4() code

```

```

AIFORM.LIST_1.DeleteAllItems
for x = 1 to len(mytree)
    AIFORM.LIST_1.ADDITEM(mytree[x][3])
next

```

End Of Resistance

```

* Resistance Code Unit : Circuits\C1\B2\addchild
Resistance RUZSLC5() code
LOCAL X

```

```

MYCH = AIFORM.LIST_1.VALUE
IF .NOT. MYCH = 0
treeid = treeid + 1
V_CHILD = AIFORM.TEXT_3.VALUE
AADD(MYTREE, {mytree[mych][2], treeid, v_child, mytree[mych][4]+1})
CALL RESISTANCE CIRCUITS\C1\B2\FILLLIST
AIFORM.LIST_1.VALUE := MYCH
ELSE
MSGINFO("You Should Select Parent from Parent ListBox", "Sorry")
ENDIF

```

End Of Resistance

```

* Resistance Code Unit : Circuits\C1\B2\removeall
Resistance RUZSLC6() code
AIFORM.LIST_1.DeleteAllItems

```

```

MYTREE = {}
                && MYTREE[N][3]
                && MYTREE[N][1] = PARENT ID
                && MYTREE[N][2] = ITEM ID
                && MYTREE[N][3] = VALUE
                && MYTREE[N][4] = LEVEL

```

```

treeid = 0
AADD(MYTREE, {0,0,"",0})
MSGINFO("Process Done","Remove All items ")

```

End Of Resistance

```

* Resistance Code Unit : Circuits\C1\B2\removeone
Resistance RUZSLC7() code

  local x
  x := AIFORM.LIST_1.VALUE
  IF .NOT. X = 0
    AIFORM.LIST_1.DeleteItem ( x )
    adel(mytree,x)
    asize(mytree,len(mytree)-1)

  ELSE
  MSGINFO("You Should Select node ", "Sorry")
  ENDIF

End Of Resistance

* Resistance Code Unit : Circuits\C1\B2\DRAWTREE
Resistance RUZSLC8() code
MYNEWARR = {}
FOR X = 0 TO TREEID
  FOR T = 1 TO LEN(MYTREE)
    IF MYTREE[T][1] = X

AADD(MYNEWARR, {MYTREE[T][1],MYTREE[T][2],MYTREE[T][3],MYTREE[T][4]})
    ENDIF
  NEXT
NEXT
mydarr = {}
AADD(MYDARR, {MYTREE[1][1],MYTREE[1][2],MYTREE[1][3],MYTREE[1][4]})
X = 1
DO WHILE X <= LEN(MYDARR)
  MYDARR2 = {}
  * ADD ELEMENTS FROM TOP ELEMENT TO CURRENT ELEMENT
  FOR T = 1 TO X

AADD(MYDARR2, {MYDARR[T][1],MYDARR[T][2],MYDARR[T][3],MYDARR[T][4]})
  NEXT
  MYID = MYDARR[X][2]
  * ADD SUCCESSOR
  FOR T = 2 TO LEN(MYTREE)
    IF MYTREE[T][1] = MYID

AADD(MYDARR2, {MYTREE[T][1],MYTREE[T][2],MYTREE[T][3],MYTREE[T][4]})
    ENDIF
  NEXT
  * ADD ELEMENTS THAT UNDER THE CURRENT ELEMENT
  FOR T = X+1 TO LEN(MYDARR)

AADD(MYDARR2, {MYDARR[T][1],MYDARR[T][2],MYDARR[T][3],MYDARR[T][4]})
  NEXT
  MYDARR = MYDARR2
X = X + 1
ENDDO
MYDRAW = ""
FOR X = 1 TO LEN(MYDARR)
MYDRAW = MYDRAW + CHR(13) + CHR(10) + SPACE(MYDARR[X][4]*10)
MYDRAW = MYDRAW + MYDARR[X][3]

```

```

NEXT
AIFORM.EDIT_1.VALUE := MYDRAW

End Of Resistance

* Resistance Code Unit : Circuits\C1\B2\DSEARCH
Resistance RUZSLC9() code
myvalue = ALLTRIM(AIFORM.TEXT_4.VALUE)
mytable = {}
aadd(mytable, {mytree[1][3], "(empty)"})
myqueue = {}
mystate = mytree[1]
do while .t.
    if ALLTRIM(mystate[3]) = ALLTRIM(myvalue)
        exit
    endif
    MYID = MYstate[2]
    * ADD SUCCESSOR
    MYQUEUE2 = {}
    FOR T = 2 TO LEN(MYTREE)
        IF MYTREE[T][1] = MYID
AADD(MYqueue2, {MYTREE[T][1], MYTREE[T][2], MYTREE[T][3], MYTREE[T][4]})
        ENDIF
    NEXT
    FOR X = 1 TO LEN(MYQUEUE)
        AADD(MYQUEUE2, MYQUEUE[X])
    NEXT
    MYQUEUE = MYQUEUE2
    * record in table
    mystr = ""
    for t = 1 to len(myqueue)
        if .not. empty(alltrim(mystr))
            mystr = mystr + "," + alltrim(myqueue[t][3])
        else
            mystr = alltrim(myqueue[t][3])
        endif
    next
    aadd(mytable, {mystate[3], mystr})

mystate = myqueue[1]
adel(myqueue, 1)
asize(myqueue, len(myqueue)-1)
    * record in table
    mystr = ""
    for t = 1 to len(myqueue)
        if .not. empty(alltrim(mystr))
            mystr = mystr + "," + alltrim(myqueue[t][3])
        else
            mystr = alltrim(myqueue[t][3])
        endif
    next
    aadd(mytable, {mystate[3], mystr})

enddo
MYSTR = ""
MYSTR = MYSTR + CHR(13) + CHR(10)
MYSTR = MYSTR + CHR(13) + CHR(10)

```

```

MYSTR = MYSTR + "Step" + space(10) + "State" + space(10) + "Queue"
for x = 1 to len(mytable)
MYSTR = MYSTR + CHR(13) + CHR(10)
MYSTR = MYSTR + STR(x) + space(10) + mytable[x][1] + space(10) +
mytable[x][2]
next
AIFORM.EDIT_2.VALUE := MYSTR

End Of Resistance

* Resistance Code Unit : Circuits\C1\B2\BSEARCH
Resistance RUZSLC10() code
myvalue = ALLTRIM(AIFORM.TEXT_5.VALUE)
mytable = {}
aadd(mytable, {mytree[1][3], "(empty)"} )
myqueue = {}
mystate = mytree[1]
do while .t.
    if ALLTRIM(mystate[3]) = myvalue
        exit
    endif
    MYID = MYstate[2]
    * ADD SUCCESSOR
    FOR T = 2 TO LEN(MYTREE)
        IF MYTREE[T][1] = MYID
AADD(MYqueue, {MYTREE[T][1], MYTREE[T][2], MYTREE[T][3], MYTREE[T][4]})
            ENDIF
        NEXT
        * record in table
        mystr = ""
        for t = 1 to len(myqueue)
            if .not. empty(alltrim(mystr))
                mystr = mystr + "," + alltrim(myqueue[t][3])
            else
                mystr = alltrim(myqueue[t][3])
            endif
        next
        aadd(mytable, {mystate[3], mystr})

mystate = myqueue[1]
adel(myqueue, 1)
asize(myqueue, len(myqueue)-1)
    * record in table
    mystr = ""
    for t = 1 to len(myqueue)
        if .not. empty(alltrim(mystr))
            mystr = mystr + "," + alltrim(myqueue[t][3])
        else
            mystr = alltrim(myqueue[t][3])
        endif
    next
    aadd(mytable, {mystate[3], mystr})
enddo
MYSTR = ""
MYSTR = MYSTR + CHR(13) + CHR(10)

```

```

MYSTR = MYSTR + CHR(13) + CHR(10)
MYSTR = MYSTR + "Step" + space(10) + "State" + space(10) + "Queue"
for x = 1 to len(mytable)

MYSTR = MYSTR + CHR(13) + CHR(10)
MYSTR = MYSTR + STR(x) + space(10) + mytable[x][1] + space(10) +
mytable[x][2]
next
AIFORM.EDIT_3.VALUE := MYSTR

End Of Resistance

* Resistance Code Unit : Circuits\C1\B3\ROOT_CHANGE
Resistance RUZSLC11() code
MYROOT = AIFORM.TEXT_2.VALUE
IF .NOT. ALLTRIM(MYTREE[1][3]) == ALLTRIM(MYROOT)
CALL RESISTANCE CIRCUITS\C1\B2\SAVE_ROOT
CALL RESISTANCE CIRCUITS\C1\B2\FILLLIST
ENDIF

End Of Resistance

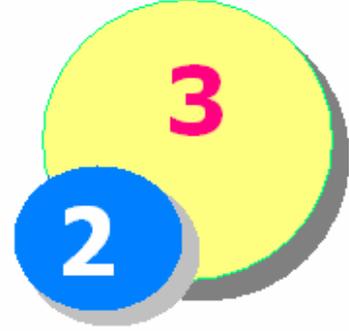
* Resistance Code Unit : Circuits\C1\B2\modify
Resistance RUZSLC12() code
LOCAL X
MYCH = AIFORM.LIST_1.VALUE
IF .NOT. MYCH = 0
V_CHILD := mytree[mych][3]

V_CHILD := INPUTBOX("Enter Name", "Modify")

mytree[mych][3] := V_CHILD
CALL RESISTANCE CIRCUITS\C1\B2\FILLLIST
ELSE
MSGINFO("You Should Select node ", "Sorry")
ENDIF

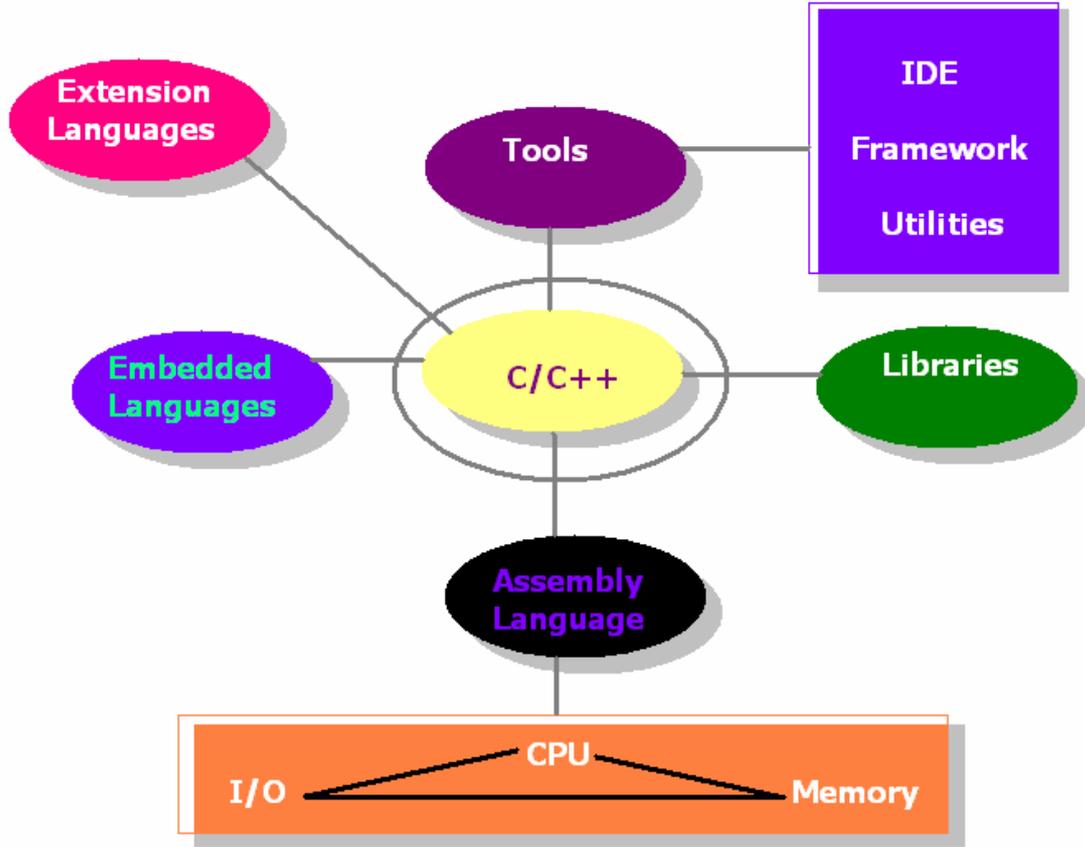
End Of Resistance
*-----*
```

ادوات برمجة النظم



ان ادوات برمجة النظم تختلف تماما عن برمجة التطبيقات - حيث ان التطبيقات غالبا ما تركز على لغة برمجة معينة ومحددة او مجموعة من النظم التى تتعاون معا فمثلا نستخدم نظام قاعدة البيانات اوراق لتخزين البيانات - ونقوم بكتابة الاكواد التى تتعامل معها باستخدام فيجوال بيسك او فيجوال فوكس برو - ونجعل واجهة المستخدم من خلال صفحات الويب.

لكن فى النظم الامر يختلف تماما
انظر الشكل التالى



فى الشكل السابق نجد ان اللغة التى تشكل نواة برمجة النظم هى بلا شك لغة C/C++ لما لها من قوة ومابها من مرونة.

تتيح لغة C/C++ التعاون مع لغة التجميع Assembly Language حيث يمكن استخدام inline assembly او استخدام ملفات تم ترجمتها بالاسمبلى Object Files
 تتيح لغة سى التعامل مع موارد الحاسب من العتاد Hardware بصورة مباشرة مثل المعالج CPU والمسجلات الخاصة به Registers وتتيح عمل مقاطعات Interrupts للمعالج وكذلك الوصول لذاكرة الحاسب Memory وايضا التعامل مع وحدات الادخال والايخارج I/O Devices
 يوجد المئات وربما الالاف من المكتبات التى يمكن استخدامها مع لغة سى - وتحتوى هذه المكتبات على المئات من الدوال الجاهزة للاستخدام فى مختلف الاغراض ومن اشهر مكتبات لغة سى المكتبة Allegro المخصصة لبرمجة الالعاب وتطبيقات تعدد الوسائط - والمكتبة WxWidgets المخصصة لواجهة المستخدم GUI

يوجد العديد من اللغات المتطورة التي يمكن ان تختبى داخل لغة سى Embedded Languages ومن امثلة هذه اللغات - لغة Python الغنية عن التعريف - ولغة Ruby الشهيرة - وهما لغتان مجانيين - ومفتوحين المصدر. يوجد لغات تم بنائها فوق لغة سى - ومن ثم تتيح دمج لغة سى داخل هذه اللغات - هنا اصبحت لغة سى هى اللغة المختبئة - ومن اشهر هذه اللغات - xHarbour وهو ايضا مجانى ومفتوح المصدر يوجد لها العديد من الادوات من بيئات تطوير متكاملة - اكثر من محيط تطوير متخصص فى انتاج فئة معينة من البرامج - والعديد من البرامج المساعدة (كتبت بلغة سى)

نلاحظ امكانية عمل النظام باكثر من لغة بمرونة فمثلا نستخدم
Assembly, C/C++, Python, Ruby & xHarbour

يرى علماء البرمجة انه لكى تكون ملما بمختلف فنون البرمجة عليك ان تتعلم خمسة لغات برمجة

- لغة التجميع Assembly Language
- لغة تدعم البرمجة الهيكلية Structure Programming
 - مثل لغة سى وباسكال وكوبول وكليبر وغيرها
- لغة تدعم برمجة الكائنات Object Oriented Programming
 - مثل لغة جافا Java و C++ وغيرها
- لغة تدعم انظمة الذكاء الاصناعى AI
 - مثل Prolog و Lisp
- لغة Script مثل VBScript و JavaScript و xHBScript

ولعلك تسال بماذا ابدا

- لغة بلغة تدعم البرمجة الهيكلية
- ثم لغة تدعم برمجة الكائنات
- ثم تعلم لغة Script
- ثم لغة التجميع

ثم لغة تدعم الذكاء الاصطناعي

ولعلك تسال عن لغات البرمجة الشهيرة - فهى

PHP

C#

AJAX (Asynchronous JavaScript and XML)

من الناحية الفنية - هى ليست لغة برمجة وانما
تستخدم XHTML or HTML, JavaScript and XML
لعمل تطبيقات الويب

JavaScript

Perl

C

Ruby and Ruby on Rails

Java

Python

VB.Net (Visual Basic .Net)

انى انصحك بتعلم برمجة الكائنات من خلال لغة Java - حتى وان
لم تكن تنوى استخدام اللغة فى مشاريعك - وذلك لانها افضل
لغة يمكن تعلم برمجة الكائنات من خلالها.

مترجمات لغة سى

يوجد العديد من مترجمات لغة سى- لمختلف انظمة التشغيل -
على سبيل المثال - لا الحصر

For dos

- BCC16 (Borland C++ 3.x, 4.x, 5.0x, DOS 16-bit)
- DJGPP (Delorie GNU C, DOS 32-bit)
- RXS32 (EMX/RXNNT/DOS GNU C, DOS 32-bit)
- Watcom (OpenWatcom, DOS 32-bit)

For Win32

- bcc32 (Borland C++ 4.x, 5.x, Windows 32-bit)
- gcc (Cygnus/Cygwin GNU C, Windows 32-bit)
- mingw32 (Cygnus/MinGW GNU C, Windows 32-bit)
- rxsnt (EMX/RXSNT/Win32 GNU C, Windows 32-bit)
- icc (IBM Visual Age C++, Windows 32-bit)
- msvc (Microsoft Visual C++, Windows 32-bit)
- watcom (OpenWatcom, Windows 32-bit)

✚ For Linux

- gcc (GNU C, 32-bit)

✚ For OS/2

- gcc (EMX GNU C, OS/2 32-bit)
- icc (IBM Visual Age C++ 3.0, OS/2 32-bit)

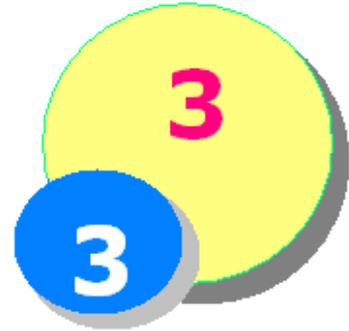
ولتعلم لغة سى يمكنك

- البداية بكتاب "الشامل فى لغة سى" تأليف خليل اونيس ويمكنك الحصول عليه من <http://www.kutub.info> وقد تفحصت الكتاب بنفسى وفى الواقع لاغبار عليه - وهو مناسب لمن لايعرف اى شىء عن اللغة - حيث ياخذ الكتاب بيد القارىء من البداية - حتى يصل به الى بر الامان - بحيث يكون القارىء ملما بجميع اساسيات اللغة.
- يمكنك ايضا قراءة كتاب "تعلم لغة سى" تأليف (حامد مصطفى قدرى) و (أحمد عبدالفتاح)
- كتاب The C Programming Language تأليف كل من **Brian W. Kernighan and Dennis M. Ritchie.** وبالتاكيد الكل يعرف دينتس ريتشى - مصمم لغة سى ونظام اليونكس
- كتاب The Complete Reference C++ تأليف Herb Schildt الغنى عن التعريف.

وبعد تعلم اللغة عليك ان تعلم جيدا انها تعطيك الفرصة للسيطرة على جميع موارد الحاسب - ولكن اللغة بادواتها القياسية غير مجهزة لاداء كل المهام المطلوبة بصورة مباشرة - لهذا سوف تحتاج للبحث فى الانترنت عن مكتبات للغة و ايضا عن برامج مفتوحة المصدر لفهم فلسفة البرمجة بلغة سى.

يقول الخبراء ان لغة سى هى لغة يتعلمها من لايعرف ماذا يريد ان يفعل بالتحديد - انها لغة يمكنها ان تفعل كل شىء يمكن عمله باى لغة برمجة اخرى.

فن برمجة النظم



ان عالم البرمجة يرتكز على ثلاث اركان

- علم البرمجة
 - من ابحاث نتجت عن تجارب
 - كتب المفاهيم والالجزومات
- فن البرمجة
 - موهبة
 - مهارة نتيجة الممارسة
- تكنولوجيا البرمجة
 - فريق عمل

○ إدارة ناجحة

وإذا سألت عن فن البرمجة - وبالتحديد برمجة النظم فعليك الاهتمام بالجوانب الأساسية للنظام - والتخصص فى جانب معين

- هناك نموذج سير العمليات
 - سبق التعرض له فى الجزء الاول بإيجاز
- هناك نمط البرمجة
 - سبق التعرض له فى الجزء الاول بإيجاز
- هناك واجهة المستخدم
 - سبق التعرض لها فى الجزء الاول بإيجاز
- هناك ادارة الذاكرة
 - عليك بدراسة كيفية تمثيل البيانات بمختلف انواعها
 - كيفية المحافظة على الاستقرار بحيث لا يتعامل النظام الا مع الذاكرة التى يحتاجها فقط لا غير
- هناك ملفات النظام
 - عليك تعلم كيفية عمل نظام ملفات خاص بك بصرف النظر عن المستوى الذى سوف تعمل فيه
 - التعامل مع القرص الصلب مباشرة
 - الاستناد على نظام التشغيل
 - الاستناد على لغة البرمجة
 - الاستناد على ادوات تطوير
- بخصوص الشبكات
 - عليك الالمام باساسيات الاتصال عبر الشبكة
 - اتقان البرمجة لكل من الشبكة المحلية وشبكة الانترنت

واعلم جيدا ان كل نظام له قصته الخاصة - واعلم انك كى تقوم بعمل تطوير لاي نظام - عليك بعمل دراسة كاملة له حتى تتأكد من عدم وجود تعارضات بين ماسوف تضيفه للنظام - مع مكونات النظام الحالية - حيث ان التشابكات بين مكونات النظام واعتمادها على بعضها البعض غالبا ما تكون كثيرة.

ومن واقع خبرتى العملية التى لا تتخطى برمجة ثلاث نظم
سوف اوجه له بعض الارشادات
لقد عملت فى النظم التالية

- 1 – Event Driven System OOP GUI Design
- 2 – Programming Paradigm – DoubleS Library
- 3 – DoubleS/RPWI – Framework

هذه النظم الثلاث تختلف كثيرا عن بعضها البعض من حيث
الهدف منها – والتكوين الداخلى – والمهارة والخبرة الازمة
لعمل كل منها.

اولا : Event Driven System OOP GUI Design

لهذا النظام متطلبات كثيرة من حيث الخبرة كى يتسنى لك
امكانية برمجته وتطويره
انه يتطلب الالمام بالجوانب التالية :-
١ – برمجة الكائنات

ينبغى ان تكون ملما بكل جوانبها – وان تكون متمرسا –
ولديك القدرة على كتابة نظم تركز تماما عليها – حيث
تستخدم الوراثة بغزارة – بينما التفويض والرسائل بين
الفصائل والكائنات اكثر بكثير – حيث ان النظام يكون
متشابه لدرجة اشبه بالمكرونة الاسبكتى - هذا من
الناحية الداخلية – بينما اذا نظرت له من بعيد – فانه يكون
منظم بحيث يسهل الوصول الى اى جزء فى النظام يختص
بامر معين او مشكلة محددة.

٢ – نموذج سير العمليات

او بالتحديد نظام ادارة الاحداث – وهو يلعب دور خطير فى
النظام حيث انه نقطة الادارة الرئيسية لكل ما يحدث داخل
النظام – وحسن استخدام نظام ادارة الاحداث احد اهم
عوامل النجاح فى برمجة النظام

٣ – واجهة المستخدم

عليك ان تكون ملما بها جيدا لانك سوف تقوم بعمل تحديث
كبير لها – ومن الناحية البرمجية فى الاسهل فى التعلم –
بينما مرحلة التطوير التى سوف تقوم بها – صعبة للغاية

ثانيا : Programming Paradigm – DoubleS Library

فى هذا النظام - قمت بعمل نمط برمجة جديد بديل لبرمجة الكائنات - والصعوبات التى تواجهها فى هذا النمط هى كالتالى

- ١ - هياكل البيانات يرتكز النظام بنسبة تكاد تصل الى ٩٠% على هياكل البيانات التى قد تكون مئات المصفوفات التى تتشابك معا فى العمل حتى تحصل على التمثيل الصحيح للبيانات.
- ٢ - استقرار وسرعة النظام اصبحت اكثر خطورة عامل خطير جدا - حيث ان نمط البرمجة يعد قلب ونقطة انطلاق البرمجيات وسوف تستخدمه مغمورا داخل اى نظام تطوره - اى ان هذا النظام هو نظام مدفون - واذا لم يكن سريعا بما يكفى فسوف تبتعد عن استخدامه - واذا لم يكن مستقرا تماما فتلك كارثة كبيرة - لنفترض مثلا انك تطور نظام اخر وليكن اسمه النظام (ب) وهذا النظام يرتكز على نمط البرمجة الذى قمت انت بعمل نظام له وليكن النظام (أ) - اذا حدث خطأ اثناء عمل النظام (ب) فيفترض ان تكون المشكلة به - بينما لا بد ان يكون النظام (أ) خاليا من المشاكل - ولكن اذا كان النظام (أ) هو سبب المشكلة فتلك مصيبة - يصعب اكتشافها - ولا طاقة لقلوبنا بتحملها.

بخصوص الاستقرار فانه غالبا يعود الى ادارة هياكل البيانات - او نموذج سير العمليات - واخطاء نموذج سير العمليات ليس من السهل اكتشافها نظرا للتشابك - بينما اخطاء هياكل البيانات ليس من السهل اكتشافها نظرا للانتشار.

ثالثا : DoubleS/RPWI Framework

ليس نظام معقد او صعب من ناحية البرمجة - وانما كان كل التعقيد يكمن فى التصميم- ولانه كبير الحجم - فى الواقع انا قلق بخصوص تسميته نظام - انه اشبه بكونه تطبيق - ولكنى اطلق عليه نظام نظرا لانه محيط تطوير Framework ويقدم تكنولوجيا جديدة.

١ - قاعدة البيانات

يشتمل هذا النظام على اكثر من ٤٥ جدول لادارة البيانات الخاصة بالعمل

٢ - نظم الملفات

يلزم محيط التطوير تعريف ملفات خاصة بها - هذه الملفات اثناء انشائها فانه يستخرجها من اكثر من جدول - واثناء فتحها فانه يقوم بتوزيع محتوياتها من جديد على الجداول المناسبة.

٣ - التطبيقات الدولية

محيط التطوير يفترض ان يعمل فى اى مكان فى العالم ولذلك كنت حريصا على امكانية تغير واجهة النظام كاملا لكى تظهر باى لغة (عربى - انجليزى - ايطالى -.....) وقد تعبت فى ذلك وخصوصا لاننى قمت بعمل تلك الامكانية بدون استخدام ادوات جاهزة - وخاصة ان شاشات ونوافذ محيط التطوير كثيرة جدا.

الخلاصة

ان كل نظام على حده يحتاج مجلد كامل لشرح فنياته - التى تتمثل فى الالجورزمات - تنظيم المكونات - ادارة التشابك فيما بينها.

وهذا امر صعب جدا - وليس من المنطق ان تتدرس اى نظام يقابلك حتى تزيد خبرتك - بل عليك اختيار النظم التى تدرسها بعناية - قم باختيار نظم تم برمجتها باللغة التى تتقنها - ثم جرب ان تصمم تلك النظم بدون الاطلاع على الشفيرة المصدرية - حدد الجوانب الصعبة التى تقابلك - ثم تعلم فقط ماتجده جديدا بالنسبة لك.

بخصوص النظم التى قمت بتطويرها فسوف تجدها فى الموقعين

<http://www.sourceforge.net/projects/fglib>

<http://www.sourceforge.net/projects/doublesvsoop>

ومن هنا فعليك باللجوء للمراجع والابحاث حتى تكون القاعدة العلمية الخاصة بك - ثم عليك باتقان التكنولوجيا لانها وسيلة التطبيق - ثم بعد ذلك اطلق العنان لفكرك - وتاكد دائما انه ليس التميز ان تكون مبرمج نظم - بدلا من كونك مبرمج تطبيقات - وانما التميز الحقيقى ان تكون مبرمج

مبدع لديه القدرة على الابتكار - والتعامل مع
المشاكل الطارئة وتذكر جيدا انك لو تعلمت كل
الكتب ودرست كل التكنولوجيا ولم تكن لديك
القدرة على الابداع فانك سوف تظل تتعلم مدى
حياتك - ولن يتعلم منك احد.

المؤلف

محمود فايد

msfclipper@hotmail.com

To Be Continue