مجلة تقنية شهرية

مجلة الرابط

تقرؤون في هذا العدد:

مدخل سريع إلى SOA و تقنية WCF
أسرار برمجية لابد أن تعرفها
قصة ظهور قاصمة الظهور: OOP
لهذه الأسباب يفشل الناس في البرمجة
المختصر النافع حول Anonymous Types
تقرير مفصل حول جدوى هندسة البرمجيات
قراءة في كتاب: C# 2008 Programmer's Reference
سورة الأنعام والإلحاد المادي
قصيدة شعرية: معادن الورى

ومنافع أخرى

رئيس التحرير: خالد السعداني

مجلة الرابط

العدد الأول - تشهر سبتمبر 2015

مجلة الرابط؟

مجلة تقنية شهرية تهتم بمجالات البرمجة والتطوير بشكل خاص، وبالمعلوميات بشكل عام، وتتطرق بشكل ثانوي لقضايا الأدب والدين.

حقل الواقع

الغرور لن يصنع منك مبرمجا..
سيصنع منك مهرجا.
ستخطو على أصابعك حتى لا تلمس
جيف من تراهم دونك..
وسيرونك بهلوانا يخطو على حبال
السيرك

المحتويات

كلمة افتتاحية بعنوان: ومنافع أخرى 2
مدخل سريع إلى SOA و تقنية WCF 3
أسرار برمجية لابد أن تعرفها 6
قصة ظهور قاصمة الظهور: OOP وكيف تهضمها؟ 9
لهذه الأسباب يفشل الناس في البرمجة 12
المختصر النافع حول Anonymous Types 17
تقرير مفصل حول جدوى هندسة البرمجيات..... 20
قراءة في كتاب: C# 2008 Programmer's Reference .. 29
سورة الأنعام والإلحاد المادي..... 34
قهوة الرابط 37
قصيدة شعرية: معادن الورى 40
وصايا مبرمج حكيم لابنه..... 41
خاتمة العدد 43

مجلة الرابط

ومنافع أخرى

رئيس التحرير

بسم الله، الحمد لله، والصلاة والسلام على رسول الله، وعلى آله وصحبه ومن والاه وبعد:

تغمرني فرحة عارمة وأنا أكتب هذه الكلمات على افتتاحية مجلة الرابط، بعد أن كانت مجرد هاجس يراود مخيلتنا فأصبحت حقيقة ماثلة أمام أعيننا.

لم يكن سهلا قط، لكنه كان أمرا ممتعا للغاية أن أخوض هذا التحدي بمفردي متوكلا على الله عز وجل، واضعا نصب عيني هدفا شريفا لا يذعرني عنه شيء حتى أبلغه، سيما بعد أن طرحت الفكرة على متابعي صفحتي على الفيسبوك فراقت لهم أيما روقان.

وددت لو أسهموا معي فيها لنشكل معا فريقا يضع لبنة جديدة في صرح أمتنا الغالية، فطرحت الفكرة عليهم، فوافقوا بشدة وأبدوا استعدادهم، لكن لما نشرت استمارة الانخراط وفتحت باب إرسال المشاركات لتحضير أول عدد، تراجع جلهم القهقرى وقليل منهم من حافظ على حماسه وراسلني.

فلما كانت بعض المقالات المرسلة مسروقة من منابر أخرى، وبعضها مليء بالأغلاط، وبعضها عبارة عن فقرات قليلة وصغيرة، قلت في نفسي: لا بأس، سوف أستغني عن المقالات غير الصالحة وسوف أصحح المقالات المغلوطة وأثريها بمزيد من المعلومات وأنشرها في المجلة، لكنني توقفت متسائلا: ماذا لو تم إصدار العدد الأول من المجلة، ثم قوبلت بالخذلان في العدد التالي سيما وأن المشاركات كانت شحيحة جدا؟

فشرعت أفكر في حل، فوجدت أن لا سبيل إلى إصدار هذه المجلة والاستمرار فيها، إلا إذا ارتكزت على عناصر ثابتة أستطيع الاعتماد عليها، ولما كانت هذه العناصر الخارجية غير مضمونة قررت أن أعتمد على العناصر الداخلية، فأوعزت إلى نفسي مهمة الكتابة والتدقيق والبحث والتصميم.

كان من المفترض أن يكون اسم هذه المجلة "عالم البرمجة"، لكنني بعد أن غيرت موقفي من الارتكاز على العناصر الخارجية غيرت أيضا اسمها ليصبح "الرابط"، لما تحمله هذه الكلمة من معنى في البرمجة Link، ولما تحمله من معنى في الحياة، فأرجو من الله العلي القدير أن تكون هذه المجلة رابط خير بيني وبين جمهور المبرمجين المسلمين، وأن يجعلها المولى عز وجل خالصة لوجهه الكريم.

وأخيرا، فإنني أتقدم بالشكر الجزيل والثناء الوفير إلى كل الإخوة والأخوات الذين شجعوني على خوض غمار هذه المجلة، خاصة متابعي صفحة العبد الفقير على الفيسبوك.

دام لكم البشر والفرح.

تقنية WCF: ثورة الخدمات

(الجزء الأول)

مدخل إلى بيئة الأنظمة الموجهة خدماتيا

Service Oriented Architecture (SOA)

خالد السعداني

شهدت بنيات الأنظمة المعلوماتية تطورا ملحوظا مع مرور الزمن، بدء من البنية التقليدية التي كانت تقدم للمستخدمين أنظمة أحادية الخدمة، عبارة عن تطبيقات انفرادية يشتغل عليها مستخدم واحد، بل ولا تقدم له كل ما يحتاجه في عمله، إذ يضطر أن يوزع مهامه بين عدة برامج، فيقوم بإدخال المعطيات في برنامج خاص بقواعد البيانات، ثم بعد ذلك يقوم بالاستعلام عن هذه البيانات من خلال برنامج آخر مخصص لهذا الغرض، فإذا أراد أن ينشىء التقارير Re-ports توجه إلى برنامج آخر وهكذا دواليك. أمام هذه التعقيدات، كان لزاما على المعلوميات أن ترقى بمجال الأنظمة، فظهر جيل جديد من البرمجيات يعتمد على الشبكات Networking platforms، بحيث تسمح هذه البنية الجديدة بتقسيم المهام بين الأجهزة والربط فيما بينها عبر الوسيط الشبكي، فعلى سبيل المثال لو أردنا نظاما متصلا بقاعدة بيانات، نقوم بإعداد جهاز سيرفر نضع عليه برنامج إدارة قواعد البيانات، ثم نقوم بتثبيت التطبيق على مختلف الحواسيب العميلة التي نريدها أن تتعامل معه، ومنه سميت هذه البنية ببنية خادم / عميل Client /Server -.

وظهرت كذلك بنيات أخرى، لعل أبرزها بنية الأنظمة الموجهة خدميا Service Oriented Architecture والمعروفة اختصارا ب SOA، هذه الأنظمة التي تكون عادة في حاجة إلى الاتصال بخدمات الويب بغرض الاستفادة منها عبر الوصول إلى الوظائف Methods التي تقدمها فتستخدمها.

واعتمدت الأنظمة الموجهة خدميا على تقنيات عديدة تسمح بنقل البيانات البرمجية (الكائنات Objects) بين مختلف مكونات النظام عبر الانترنت أو الشبكات المحلية، لأنه ليس لزاما في هذه البنية أن تكون مكونات نظام معين موجودة على نفس الجهاز، إذ أن هذه البنية معدة خصوصا للأنظمة التي تتميز بتباعد مكوناتها جغرافيا، وتسمى هذه الأنظمة بالأنظمة الموزعة Distributed Systems.

من غير شك سبق لك وأن سمعت بالأنظمة الموزعة أو التطبيقات الموزعة Distributed Applications، فهي بكل بساطة أنظمة مركبة وعناصرها متفرقة مكانيا، وتتواصل هاته العناصر فيما بينها عبر الآليات الشبكية، إما عبر الانترنت أو عبر الشبكة المحلية للمؤسسة.

عملية تبادل الكائنات بين مختلف مكونات النظام الموزع تحتاج إلى بعض التقنيات المتقدمة، مثل تقنية الويب سيرفيس، تقنية net remoting. - تقنية RMI في لغة جافا، تقنية COM (Component Object Model) التي تسمح بالتواصل بين التطبيقات والمكونات الموجودة على نفس الجهاز، وتقنية DCOM (Distributed COM) التي تعتبر النسخة الموزعة من تقنية COM بحيث تسمح بالتواصل بين التطبيقات والمكونات الموجودة على عدة أجهزة عبر استخدام الشبكات، ويوجد ماهو أحدث منها كتقنية COM+، وتقنيات أخرى كثيرة إلا أننا سنغض الطرف عنهم كلها ونركز فقط على صديقتنا الغالية WCF، وسوف نتناولها بإذن الله في عدة مقالات سيتم نشرها على التوالي في باقي أعداد مجلة الرابط التقنية، فما هي تقنية WCF؟ ومتى ظهرت؟ ولماذا سنحتاجها؟ وإذا احتجناها كيف نستخدمها ؟

بداية، فإن تقنية WCF تعد اختصارا ل Windows Communication Foundation، وقد صاحبت إصدار الدوت نيت فريموورك 3.0، وطرحتها ميكروسوفت كتقنية بديلة للتقنيات المستخدمة في الأنظمة الموزعة لإنهاء مشاكل عدم التوافقية بين مكونات النظام.الجميل في التطبيقات الموزعة أن مكوناتها قابلة لإعادة الاستخدام، إضافة إلى كونها مهيأة للاستخدام المتعدد من طرف عدة تطبيقات في نفس الوقت، والأجمل من هذا وذاك أن هذه المكونات مخفية عن عين المستخدم النهائي End User الذي لا يعلم ما الذي يجري في خفايا النظام الذي يستعمله، إذ كل ما تقع عليه عيناه يتمثل في واجهات الاستخدام، إما نماذج ويندوز Windows Forms أو صفحات ويب Web Forms.

الهدف الرئيسي من تقنية WCF هو السماح بنقل الكائنات Objects عبر الشبكات المحلية للمؤسسات أو عبر الانترنت من أجل تمكين التطبيقات التي تحتاج هذه الكائنات من الوصول إلى وظائفها واستعمالها.

لعمل ذلك، لابد من استضافة خدمات WCF على جهاز سيرفر، وإعداده بكيفية تجعله قادرا على التجاوب مع استعلامات العملاء Clients.

من جهة العميل Client Side، ونتحدث هنا عن التطبيق المثبت على الأجهزة العميلة التي ستستخدم خدمات WCF المرفوعة على سيرفر، فلا بد أن نوفر لهذا التطبيق العميل جميع المعلومات الأساسية التي تسمح له بالاتصال بخدمات WCF واستعمالها بشكل سليم.

يتبع

فائدة

بعض الأنواع الرقمية في البرمجة تكون مسبوقة بالحرف U مثل UInteger و ULong وهي اختصار ل Unsigned Integer و Unsigned Long إلخ.. الكلمة Unsigned تعني أن هذا النوع الرقمي خاص باحتواء القيم الإيجابية فقط، بمعنى لو أنك ستحتاج إلى متغير لتخزين القيم الإيجابية فقط دون القيم السلبية، فقم بالإعلان عنه من نوع Unsigned لأنه سيأخذ نفس مساحة التخزين في الذاكرة التي يأخذها النوع الأصلي إضافة إلى أنه يتيح لك مجالا رقميا يضاعف المجال الرقمي الخاص بالنوع الأصلي. مثلا النوع Short يسمح باستعمال المجال الرقمي من من -32,768 إلى 32,767، **لو استعملنا النوع Unsigned** من هذا النوع سيسمح لنا بتخزين ضعف هذا المجال لأنه لا يقبل القيم السلبية وبالتالي سيتيح لنا مجالا رقميا أكبر للاعداد الإيجابية، أي أن النوع UShort سيسمح لنا بتخزين القيم الرقمية من 0 إلى 65,535.

أسرار برمجية لابد أن تعرفها

دعني أصدقك القول من الأول، لا يوجد سر للنجاح في أي مجال، وفي البرمجة تحديدا لا توجد خلطة سحرية تقوم بإعدادها فتمطر عليك السماء أكوادا، بل توجد وصايا - سنسميها أسرارا لنجر انتباهك ونشد تركيزك - عليك الأخذ بها لتختصر طريق الاحتراف وإلا فإنك ملزم بالوقوع في عدة أخطاء تستنزف وقتك وتمتص حماسك، وتعيدك باستمرار إلى نقطة البداية.

خالد السعداني

السر الأول: لغة من كل فن، خير من فن من كل لغة

ليس مهما عدد اللغات التي تتقنها. فاللغات في آخر المطاف تبقى مجرد وسائل لإنتاج البرامج. فمهما حاولت أن تستكثر اللغات فإن ذلك لن ينفعك بشيء بل سيضيع عليك وقتك لتكرر نفس العمل على أكثر من لغة وأنت بين هذه القفزات والقفشات واهم. تتخيل أنك حزت مقاليد البرمجة والحقيقة المرة أنك لم تحز من البرمجة إلا جزء يسيرا قمت بتغيير حلته من لغة إلى لغة.

ما قيمة أن تتقن أساسيات البرمجة بأكثر من لغة؟ ما معنى أن تطبع جملة "أهلا بالعالم" على جافا. باسكال. كوبول. دلفي...؟ لماذا هذا التباهي إلى حد التماهي؟ شوف. بدل أن تعيد صناعة العجلة. انتقل إلى نمط جديد من البرمجة فذلك أنفع لك. الوقت الذي ستقضيه في تعلم أساسيات لغة سي شارب وأنت متمكن من أساسيات لغة جافا يمكنك قضاؤه في إتقان مكتبة Swing أو AWT أو غيرها.

الوقت الذي ستضيعه في برمجة الواجهات بلغة دلفي وأنت متمكن منها في لغة الفيجوال بسيك. يمكنك قضاؤه في احتراف برمجة الويب بتقنية asp.net.

يمكنك تعلم لغة جديدة فقط حينما تعجز لغتك الأم عن تقديم النمط المنشود. هاجر إلى لغة جافا مثلا حينما تعجز لغتك عن إنتاج تطبيقات الموبايل. وهاجر إلى لغة سي شارب مثلا حينما تعجز لغتك عن إنتاج مشاريع الويب.

وتذكر جيدا: نفس اللغة البرمجية من كل فن. خير من نفس الفن من كل لغة !

السر الثاني: استسهل الصعاب، تبلغ الرغاب

البرمجة كغيرها من المجالات. فيها السهل وفيها الصعب. ومقدار الصعوبة نسبي حسب القدرة الذهنية لكل مبرمج. وهنالك سبب آخر أراه أشد وقعا وأثرا من هذا. وهو نظرة المبرمج إلى الأمر الصعب.

نعم إنها النظرة. أهي نظرة تخوف يتلوها جمود وبرود. أم نظرة استصعاب تتلوها المدارسة والممارسة.

إن كنت ترى في البرمجة الكائنية مثلا بحرا شاسعا لا تستطيع السباحة فيه. ثم رضيت بتجاهلها فلا تحلم أن تستيقظ في إحدى الأيام وأنت متمرس فيها.

الأمور تمشي هكذا ولا خوارم لسنن الله الكونية. حتى النملة لو نظرت إلى حبة القمح نفس نظرتك. لهلك النمل وانقرض نسله.

والله عز وجل يطالبنا ببذل الأسباب من أجل تحصيل الرغاب. والاجتهاد من صميم التوكل ولا تعارض في ذلك. أما من يزعم أنه توكل ولم يبذل جهدا فلن يحصد إلا الريح. والذي قام به ليس توكلا وإنما تواكلا.

انظر إلى كل صعب نظرة استسهال. واعلم أن كل مركب أساسه بسيط. فابدأ من الأساس متدرجا تبلغ المركب دون أن تستشعر ذرة تعب ذهني.

ألخص لك: استصغر ما يستعظمه الناس تتفوق عليهم. ابدأ من البسيط يهون المركب. مادام غيرك يتقن هذا الصعب فما المانع لديك !

السر الثالث: نظم عملك، تحقق أملك

التنظيم ينبغي أن يلزم كل تفاصيلك البرمجية. بدء من اختيار نوع المشروع واللغة البرمجية المناسبة له. مرورا بأسماء المتغيرات والملفات. وانتهاء بالتحزيم وإعداد كافة الملفات اللازمة لاشتغال البرنامج عند العميل بكفاءة عالية.

حينما تنظم عملك. لن تبذل جهدا إضافيا في أمور قد تشتت ذهنك في حال العمل الفوضوي. حتى تنظيمك لأجزاء ووحدات المشروع وبأيها تبدأ وبالكيفية الواجب نهجها للإنجاز يلعب دورا أساسيا في تسهيل عملية الإنتاج. الشيء الذي سيدفعك من غير شك إلى إنجاح مشاريعك.

هنالك أمور صغيرة لا ننتبه إليها لكنها قد تعبث بأفكارنا وتحول بيننا وبين الصفاء الذهني. كأن تستصغر مثلا دور الورقة والقلم في تدوين ملاحظات تطرأ لك في وقت من الأوقات. أو تعمل في غرفة يضيق صدرك بها. حاول أن تكون منظما في أدق التفاصيل لينعكس هذا الصفاء على برامجك.

السر الرابع: اضبط نومك، تربح يومك

النوم مهم جدا للجهاز العصبي. نعم لا يوجد في ذلك شك. لكن بقدر محدد. أما أن تنام لساعات طويلة بحجة أن جهازك العصبي يحتاج لذلك. فلعمري إنها فرية !

كلما نقصت عدد ساعات النوم. كلما ربحت أوقاتا إضافية يمكنك قضاؤها في التعلم أو العمل. وستلاحظ أنك كنت ظالما لنفسك بتضييع أوقات كثيرة في النعاس وسترى كيف سيزداد إنتاجك وكيف ستتحسن حياتك.

لن أقول لك ماهي الساعات الكافية للنوم لأنني لست متخصصا في الجهاز العصبي. ولن أقول لك ماهي الفترة التي ينبغي أن تستغلها في البرمجة لأن ذلك متغير حسب ميولات كل شخص. لكن ما سأقوله

نفسك كالطفل إن أقنعتها أن أربع أو خمس ساعات من النوم كافية جدا. ثم أتبعت قولتك هذه ببرنامج عملي تلزمها فيه بالنوم لهذا القدر دون أن تتجاوزه. فسوف تربح وقتا كثيرا. والوقت يعني الإنتاج. والإنتاج يعني الأرباح.

أما متى تبدأ يومك. فإنني أنصحك أن يكون ذلك بعد صلاة الفجر. لأن البركة تحل في الوقت إن بدأته باكرا وسترى أن اليوم لم يكتمل بعد وقد أنجزت الكثير.

وتذكر جيدا: اضبط نومك. تربح يومك.

السر الخامس: دَوْنَ كل جديد، تصنع ذاكرة من حديد

لا تثق بذاكرتك اللحظية فقد توهمك بضبط الشيء. لكن بعد مرور الوقت إن سألتها عنه قلبت لك ظهر المجن !

ثق بما تخطه بيدك. فإنه يبقى وكلما احتجته وجدته في المتناول. أما ما نتوهم أنه رسخ في الذهن فإنه غير مأمون وهو من غير شك إلى زوال إذا لم يراجع ويدارس.

كم هي المعلومات التي قرأناها وحسبنا أننا لن ننساها لكن بعد مضي الزمن سئلنا عنها فكأننا لم نقرأها. وكم هي المفاهيم البرمجية التي فهمناها وطبقناها لكن مع تقادم الوقت تنكرت لنا وكأننا لم نداعب شفراتها في الماضي !

العيب ليس فيما ينسى. ولكن العيب فيمن يعجز عن التدوين.

السر السادس: التنظير بلا تطبيق، مجرد "تَخَرْبِيق"

لا جرم أن الاكتفاء بالجانب النظري دون إخضاعه للممارسة والتطبيق ضرب من الجنون. وبالمناسبة كلمة "تخربيق" هي كلمة باللهجة المغربية تعني "عبث". فالتنظير من غير تطبيق عبث وسوء تقدير. بنفس الميزان يمكننا قياس التطبيق من غير تنظير. فهو أيضا "تخربيق" و عبث. لأنك قد تنجز الشيء وأنت لا تفهمه أساسا. فما بالك لو طلب منك تعديله أو تحسينه؟ بل ما بالك لو حصل في برنامجك خطأ فكيف ستصلحه وأنت أساسا أهملت الجانب النظري واكتفيت بنسخ سورس كود ولصقه ظنا منك أن التطبيق هو كل شيء. الموازنة بين التنظير والتطبيق مطلب أساسي في مجال البرمجة. إذ لا عمل من غيرعلم. ولا علم من غير عمل.

فصل تمهيدي

قصة ظهور قاصمة الظهور (البرمجة الكائنية التوجه)

قصة ظهور قاصمة الظهور

البرمجة الكائنية التوجه

Object Oriented Programming

خالد السعداني

في بداية عصر المعلوميات، لم تكن البرامج سوى أوامر متسلسلة مكتوبة بلغة الأسمبلي، يمر عليها المعالج أمرا أمرا ليقوم بتنفيذها، ولم تكن تلك البرامج قابلة للنقل Portable لتعمل على مختلف أنظمة التشغيل، بل حتى على نفس نظام التشغيل كانت تختلف أوامر لغة الأسمبلي حسب نوع المعالج Processor Family مما يجعل البرنامج الواحد المكتوب بلغة الأسمبلي حكرا على عائلة واحدة من المعالجات، لذلك ستسمع عدة نسخ من لغة الأسمبلي حسب نوع المعالجات مثل لغة الأسمبلي الخاص بالمعالج Motorola 68x00، أو الخاصة ب Intel IA-32، أو الخاصة ب IBM-370 وغير ذلك..

أما عملية تنفيذ البرامج بواسطة المعالج فكما نعلم جميعا، يتم تخزين الأوامر في الذاكرة الرئيسية على شكل بيانات ثنائية، فيأتي المعالج ليقوم بتنفيذها بدء بأول أمر ثم بالذي يليه ثم الذي يليه إلى أن يصل إلى نهاية البرنامج فيقوم بإرسال النتيجة إلى جهاز الإخراج المخصص لذلك (الشاشة، الطابعة، مكبرات الصوت..)، وهو نفس مايعرضه الشكل التالي:

فلو أحببنا أن نلخص المهام التي كانت تقوم بها البرامج فيمكن أن نعدها على رؤوس الأصابع:

- قراءة البيانات وتخزينها في الذاكرة
- تغيير عناوين البيانات Adresses على الذاكرة عبر التعامل مع المتغيرات
- القيام بالعمليات الحسابية (الجمع، الضرب...)
- القيام بالعمليات المنطقية (المقارنات، تحقق من نتيجة الشرط..)

مع تطور المعلوميات وزيادة الحاجة إلى برامج وأنظمة معقدة، والحرص على التركيز على حل المشاكل البرمجية بدل التركيز على حل مشاكل كتابة الأوامر، ظهر نوع جديد من لغات البرمجة متقدم نسبيا، غير مرتبط بنوع معين من المعالجات (لأن عملية التحويل من هذه اللغات إلى اللغة الثنائية يتم عبر برامج وسيطة تسمى مترجمات Compiler / مفسرات Interpreter) ويقدم أوامر قريبة نوعا ما من اللغة البشرية، إضافة إلى كونه يسمح بتقسيم البرنامج إلى أجزاء، وعند استدعاء جزء ما باسمه يتم تنفيذ الأوامر المنضوية تحته، وسمي هذا النوع من اللغات باللغات الإجرائية Procedural Languages، وجاءت هذه التسمية كناية على النمط البرمجي الذي تعتمده هاته اللغات وهو النمط الإجرائي، حيث يتم تجميع الأوامر داخل نفس البرنامج على شكل إجراءات Procedures بغرض تفادي إعادة كتابة الأوامر من جهة، وبغرض تسهيل عملية كتابة وصيانة الكود من جهة أخرى.

وظهرت مع هذا النمط البرمجي لغات عديدة، لعل أبرزها على الإطلاق لغة سي C، ثم ما يردفها من لغات تمشي على نفس المنوال مثل لغة باسكال Pascal، فورتران Fortran، ولغة البسيك Basic وغيرها...

واستمرت الأنماط البرمجية في التقدم إلى أن ظهر نمط جديد يسمى البرمجة الكائنية التوجه Object Oriented Programming، والمعروف اختصارا ب OOP، وكان أول ظهور له في بداية الستينات من القرن المنصرم سنة 1960 في مجال الذكاء الصناعي arti-ficial intelligence (وهو مجال معلوماتي متفرع من علم الحاسب ويرتكز بالأساس على صناعة أنظمة ذكية تهدف إلى رفع مستوى النجاح البرمجي).

جاءت البرمجة الكائنية كنمط برمجي يهدف إلى حل مشكلة الأنظمة المعقدة عبر اعتماد مجموعة من التدابير البرمجية التي تهدف إلى تقسيم المشاكل الكبيرة إلى مشاكل صغيرة ليسهل حلها، وكذلك من خلال لاعتماد على مفهوم الكائن Object في تجميع البيانات، بحيث يتم اعتبار كل وحدة مستقلة بذاتها نوعا شاملا يصطلح عليه بالفئة Class، هذه الأخيرة تشمل جميع الخصائص Propeties والوظائف Methods والأحداث Events التي يتميز بها الكائن، الذي بدوره يعد نسخة In-stance من الفئة.

يتبع...

لهذه الأسباب يفشل الناس في البرمجة

مضت سنون طوال و "شادي" يداعب لوحة المفاتيح ويعبث بالأكواد، بعزيمة متقدة لا تفتر، وبهمة عالية لا تكل، كان يسعى إلى بلوغ قمة النجاح البرمجي دون أن يستبين ملامح هذا النجاح، كان يجري فقط نحو صورة هلامية رسمها في ذهنه يخلط فيها بين نبوغ دنيس ريتشي وثراء بيل غيتس، كان يسعى إلى أن يجمع بين هاتين الملكتين، عقل قوي يقهر الخوارزميات، وجيب ثري يحقق أسباب العيش الكريم

بقلم: خالد السعداني

لهذه الأسباب يفشل الناس في البرمجة

خالد السعداني

مضت سنون طوال و "شادي" يداعب لوحة المفاتيح ويعبث بالأكواد، بعزيمة متقدة لا تفتر، وبهمة عالية لا تكل، كان يسعى إلى بلوغ قمة النجاح البرمجي دون أن يستبين ملامح هذا النجاح، كان يجري فقط نحو صورة هلامية رسمها في ذهنه يخلط فيها بين نبوغ دنيس ريتشي وثراء بيل غيتس، كان يسعى إلى أن يجمع بين هاتين الملكتين، عقل قوي يقهر الخوارزميات، وجيب ثري يحقق أسباب العيش الكريم.

كان مقبلا على هذا النجاح التجريدي إقبال الظامىء على الماء البارد في اليوم القائظ، فلا يرى دورة برمجية على اليوتيوب أو على غيره من المنصات التعليمية إلا وسارع إلى تحميلها وحفظها عنده، ولا يرى كتابا يستفز نهمه المعرفي إلا وانقض عليه انقضاض الليث الهصور على الغزال اللاغب، حاسوبه الخاص مكتبة غنية بالمصادر، بينما عقله منشفة تبتل بالأكواد ثم تجف..

ينسخ السورس كود ويغير تصميم الواجهات بحثا عن إرضاء نفسه وإبهار أقرانه، فيبتسم في حبور وفخر وفي داخل قرارة نفسه يردد: تبا لي من غبي، ألهذا أنهكت عقلي وبدني؟

استمر به الحال حينا من الزمن حتى ضجر من وضعه، وكره أن يستمر في خداع نفسه، وآن له أن يصدع بما يختلجه من قلق لعل المحاسبة تفضح العيوب، فتنفرج الكروب.

توجه إلى كوخ حكيم البرمجة، طرق بابه بأدب ففتح له و أذن له بالدخول.

لم يمهل "شادي" الحكيم حتى يرجع إلى مجلسه، بل خاطبه بشدة:

ألم تقل لي أن البرمجة سهلة؟ ألم تردد على الدوام أنها مدرة للأرباح؟ للأسف صدقتك وقطعت زمنا ليس باليسير ماضيا على الخط الذي رسمته لي فما ظفرت بشيء مما كنت قد خططت له.

انظر إلى الناس كيف صاروا مبرمجين أفذاذا، منهم من أنشأ مؤسسته الخاصة، منهم من ترقى في عمله، ومنهم من يعمل حرا وقتما يشاء وحسابه المصرفي يزخر بالدولارات، أكلهم سلكوا الخط الذي رسمته؟ لا أظن ذلك أبدا، لأنهم لو سلكوه لوجدتهم معي الآن ساخطين يصبون عليك جام غضبهم، تكلم يا حكيم فقد خابت أكوادي وضاع معها مرادي !

ابتسم الحكيم بهدوء، ثم تخلل لحيته الطويلة البيضاء بأصابعه في برود و"شادي" أمامه تصطك فرائصه كالصريع.

رفع الحكيم رأسه و نظر في عيني شادي وقال له هامسا:

لعلك لم تفهم الدرس يا بني.

ثم واصل بعد أن غير نبرة صوته ليشد انتباه "شادي" إليه:

البرمجة عالم خاص يقع بين الفن والعلم، فمن ظنها فنا صرفا أمضى عمره ينمق الأكواد ويزين الواجهات فيتفاجأ حينما يخيب مسعاه ليعود منها بمثل مايعود به دارس الرياضيات صاحب الخط الجميل والفهم العليل، ومن ظنها علما بحتا أمضى سني عمره يضع المعادلات، ويصوغ المخططات فينصدم لما ينجلي السواد ليعود منها بمثل مايعود به دارس الرسم ذو الفهم السليم والذوق العقيم.

لا يستقيمان يا بني، البرمجة فن وعلم، وإنما أصابك ما أصابك من فشل وسقوط بسبب لزومك لست خصال، فإن أنت تخلصت منها كان بينك وبين هدفك شعرة، وإن أنت لازمتها فإنني أضمن لك أنك لو أنفقت عمرك ومعه مثله لما تحركت من موضعك قيد أنملة.

سكت شادي في اندهاش، ثم قطع صمته متسائلا:

وما تلك الخصال الستة يا حكيم؟

رفع الحكيم سبابته مشيرا وهو يقول:

أولا: العجلة

لقد رأيتك وأنت تنتقل من لغة إلى لغة مسرعا وفي عجلة من أمرك، فلا يكاد يهنأ لك بال مع لغة جافا حتى تقفز إلى لغة بايتون، ولا يكاد يستقيم حالك مع برمجة قواعد البيانات في سي شارب حتى تنط إلى برمجة الألعاب في سي بلس بلس، لم تستوعب فريموورك سامفوني فتركته مستهترا وتوجهت إلى لارافيل فأنت ملول عجول، تبرمج لتتباهى أمام الأقران فتتناهى كحطب الأفران، وتتعلم لتماري نفسك فتداري نقصك، أما بلغك أن في التأني السلامة وفي العجلة الندامة، أما وقد أدركت الآن أصل الخلل فاعزم على ترك العجلة في البرمجة فإنها مدعاة إلى الفشل.

ثانيا: كثرة الموارد

فلما نهجت للبرمجة نهج المستعجل، أكثرت من جمع الدورات والمحاضرات والدروس والكتب والمقالات، ولو سألتك الآن كم كتابا قرأت في البرمجة لأجبتني أنك لم تكمل كتابا واحدا، ولو سألتك كم دورة برمجية تابعتها من بدايتها حتى نهايتها ودونت ماجاء فيها من معلومات نظرية وأنجزت ما ورد فيها من تطبيقات وأمثلة وتوسعت فيها بالبحث والتنقيب لكان جوابك صفرا، لأنك تفتح المحاضرة وتمرر شريط القراءة لتسمع فقط ما تحب سماعه وما يغص به حلقك من معلومات مهمة تتجاهلها أو ترجئها إلى حين فتحرم نفسك من خير كثير لو صبرت عليه لبلغت ما تصبو إليه، تريد مثالا: تدرس لغة سي شارب وتشاهد فقط دروس برمجة الواجهات وتهرب من المفوضات Delegates و تقنية Linq وتعدد المسارات Multi-Threading هروب السليم من الأجرب، ولو أدركت مافي هذه المفاهيم - وغيرها مما تهرب منه - من الخير لعضضت عليها بالنواجذ.

قم بحذف جميع الكتب والدورات والموارد المعرفية من حاسوبك واحتفظ بمصدر واحد فقط لأن الوفرة تدفع إلى الانتقائية، والانتقائية تدفع إلى المماطلة.

ثالثا: غياب الخطة والرؤية الواضحة

لعل غياب التخطيط يعد من أعظم أسباب الفشل في البرمجة، لما له من أثر سريع على أداء المبرمج، لأن غياب التخطيط يعني عدم الجدية في العمل، فكيف تكون جادا في تعلم جافا وأنت لم تخصص لها وقتا؟ بل كيف تكون جادا في برمجة ألعاب الأندرويد وأنت لم تضع مخططا يوضح لك من أين تبدأ وإلى أين تنتهي؟ وقياسا على قول المفكر الأمريكي بنيامين فرانكلين: "إذا فشلت في التخطيط فقد خططت للفشل" يمكننا القول: "إذا غاب عنك التخطيط فقد غاب عنك النجاح".

اكتب أهدافك دائما، وقم بالتخطيط لكل هدف عبر كتابة الخطوات المفضية إليه لكي تتضح الرؤية وإلا فإنك ستبقى تضحك على نفسك.

رابعا: السعي إلى الربح السريع

قد تلتزم بالأناة في التعلم، وقد تحتفظ بمصدر معرفي مكتمل، وقد تضع مخططا سليما يشمل مختلف خطوات بلوغ الهدف الذي حددته، لكن بمجرد ما تفتح صفحتك على الفيسبوك تطالعك صورة أمين رغيب وهو يتوشح وساما من حاكم دبي أو صورة لسيمو لايف وهو ينشر شيكا وصله من كليك بانك قيمته آلاف الدولارات أو غيرهم، حتى تختل موازينك وترمي ما كان في يدك وتركض إلى غير هدف وتنسى كل ما عاهدت نفسك بملازمته لأن الطيش يجري فيك مجرى الدم في عروقك، فتبدأ بالبحث عن كيفية الربح من التسويق الالكتروني ثم تدخل في دوامة جديدة لن تخرج منها بطائل ولو أنك تمهلت قليلا وواصلت سيرك لتفوقت على كل من تراهم قد بلغوا ذروة النجاح، ولكنه الطيش يأبى إلا أن يلازم صاحبه حتى يحيله إلى فاشل محبط.

خامسا: النقاشات الجانبية

لا تشغل بالك بالنقاشات الجانبية التي لن تضفي إلى صرحك المعرفي لبنة واحدة. ركز في هدفك وثق في الأدوات التي تستخدمها ولا تدافع عنها بعصبية أمام خصومها لأنك لن تقنعهم بأي حال من الأحوال. فهم أيضا متعصبون إلى أدواتهم ويرونها أبهى من ضفائر بثينة.

إن كنت تبرمج بلغة سي شارب فلا تنتقص لغة جافا. لأن ذلك سفاهة منك و سوء تقدير بل تصرف صبياني سيفتح عليك باب الويلات. خلاص أنت مقتنع بلغتك؟ طيب واصل سيرك ولا تلتفت إلى أي نقاش جانبي لأن ذلك من خوارم المروءة قبل أن يكون من خوارم البرمجة.

سادسا: التقليد

في درب النجاح لا مكان للنسخ المكررة. إما أن تكون نفسك أو تترك سبيل النجاح وتنهج سبيل المقلدة الذين لن يضيفوا شيئا جديدا. وأقصى ما يستطيعون تقديمه لن يتجاوز بأي حال من الأحوال سقف ما قدمه من سبقوهم. لأن التقليد يقتل الإبداع. وإذا مات فيك الإبداع فلا تفكر في النجاح أبدا.

استفد من الكل ولا تقلد أحدا. حتى يجتمع فيك ما تفرق في غيرك.

مغارات نخشى وطأها

المختصر النافع حول الأنواع المجهولة
Anonymous Types

خالد السعداني

مقدمة:

الأنواع المجهولة Anonymous Types هي طريقة برمجية في لغة سي شارب تتمثل في استخدام الكلمة var عند التصريح مكان نوع البيانات. وذلك بغرض تمكين المترجم Compiler من تحديد نوع بيانات المتغير بناء على القيمة البدئية المسندة له.

وهذه طريقة استخدام الكلمة المحجوزة var:

```
var name = "Ahmed Dahbi";
var age = 25;
var isMarried = false;
```

المترجم سيستنتج أن المتغير الأول name نوع بياناته نصي string بحكم قيمته الأساسية. وبنفس الكيفية سيعلم أن المتغير الثاني رقمي int والثالث منطقي bool. للتحقق من ذلك يمكننا استخدام الوظيفة GetType التي ترجع لنا نوع بيانات الكائن كما يلي:

```
Console.WriteLine(name.GetType());
Console.WriteLine(age.GetType());
Console.WriteLine(isMarried.GetType());
```

وستكون النتيجة عند التنفيذ كما يلي:

إلى هنا الأمور عادية جدا، وقد تتساءل وأنت غاضب، هل خالد أصبح مجنونا؟ أما كان يكفيه أن يعلن عن تلك المتغيرات المشؤومة بالطريقة العادية، فيريحنا من عناء المصطلحات 😎 ؟

سأمضي في الشرح وكأنني لم أقرأ تعليقك 😅 لنواصل يا صديقي

كان من الممكن أن نعلن عن تلك المتغيرات بالطريقة المألوفة كما يلي:

```
string name = "Ahmed Dahbi";
int age = 25;
bool isMarried = false;
```

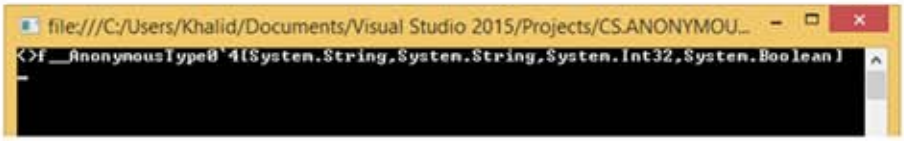
لكنني ارتأيت أن أبسط الأمر لكي نتعرف على دور الكلمة var، أما الآن وقد استوعبت ذلك فتعال بنا نتعرف على الأنواع المجهولة بشكل أعمق قليلا.

تسمح لنا الأنواع المجهولة بإنشاء كائنات Objects في نفس زمن التصريح دون الحاجة إلى إضاعة الوقت في إنشاء الفئات Classes، خصوصا حينما نحتاج إلى استخدام كائن لمرة واحدة في البرنامج فلا داعي لإنشاء فئة كاملة له ثم استنساخه منها، بل بكل بساطة نستخدم الكلمة new ونضع بعدها ما نشاء.

سأقوم في المثال التالي بإنشاء كائن يشمل عدة حقول دون أن أحتاج إلى تحديد أنواع البيانات ولا إلى إنشاء الفئات الرئيسية، تابع معي:

```
var Person = new { name = "Hamid Makboul", Country = "Morocco", Age = 26, isLocked = true };
```

هكذا بكل بساطة أنشأت كائنا أسميته Person دون أن أقوم بإنشاء الفئة أو أحدد أنواع الحقول، لو استدعيت الوظيفة GetType الآن بالنسبة لهذا الكائن فسوف تعيد لي القيمة التالية:



لو قرأت في ثنايا القيمة المطبوعة، ستجد نوع بيانات كل حقل من حقول الكائن المعلن عنه، نوع بيانات الحقلين الأول والثاني string، والثالث نوعه رقمي int، والأخير نوعه منطقي bool.

هذا من جهة، ومن جهة أخرى سنحتاج إلى استخدام الأنواع المجهولة حينما يكون عندنا استعلام على مصدر بيانات معين ونجهل نوع بيانات النتيجة المرجعة.

على سبيل المثال لو عندنا في البرنامج مصفوفة نصية، وأردنا الاستعلام عن النصوص التي تبدأ بحرف M عبر استخدام تقنية Linq، فإن الأمر سيكون كما يلي:

```
string[] Values = { "May", "Red", "KeyBoard", "Monitor", "Alice" };
var list = from value in Values where value.StartsWith("M") select value;
foreach (var item in list)
{
    Console.WriteLine(item);
}
Console.ReadLine();
```

كما تلاحظ معي، فنحن استخدمنا الكلمة var من أجل الحصول على نتيجة الاستعلام، ثم قمنا بطباعة النتيجة بواسطة تكرار من نوع foreach، إذن فالنتيجة ستكون على الشكل التالي:

تفريغ نصي لمحاضرات خالد السعداني

محاضرة بعنوان: هندسة البرمجيات
Software Engineering

خالد السعداني

بسم الله الرحمن الرحيم، الحمد لله رب العالمين والصلاة والسلام على الرحمة المهداة للعالمين، سيدنا محمد وعلى آله وصحبه أجمعين، وعلى التابعين، ومن تبعهم بإحسان إلى يوم الدين، ثم أما بعد:

أيها الاخوة الأفاضل، أيتها الأخوات الكريمات:

السلام عليكم ورحمة الله تعالى وبركاته.

بداية، اسمحوا لي أن أتوجه بالشكر الجزيل والثناء الوفير لأخي الحبيب يونس معدان الذي بذل جهدا كبيرا في سبيل أن يتحقق هذا الجمع المبارك، وإلى سائر أفراد إدارة المدرسة العليا للأساتذة، ثم إلى سائر السادة الحضور من أساتذة أجلاء وطلبة فضلاء.

في الحقيقة، كنت أنتوي الحديث عن موضوع من موضوعات الدوت نيت بحكم طبيعة تخصصي، كأن نتحدث عن الفروق بين لغة الفيجوال بزيك دوت نيت وبين لغة سي شارب، أو أن نتحدث عن التقنيات الحديثة التي رافقت إصدارات الدوت نيت الأخيرة بدء من النسخة 3.5 مثل تقنية Entity Frame-work، WPF، WCF، LINQ وغيرهم، لكن بلغني أن أخي الفاضل أنس صحري سيتناول في محاضرته نطاق الدوت نيت عبر عرض أوجه الاختلاف والائتلاف بينه وبين نطاق جافا.

ففكرت في الأمر مليا، فخرجت بنتيجة مفادها أن الحديث عن نفس الموضوع ولو من زوايا مختلفة قد يسبب نوعا من الرتابة والملل لدى المتابعين الكرام، فارتأيت أن أغير الوجهة صوب موضوع شامل يخص كل لغات البرمجة، ويكون له تأثير جوهري على عملية صناعة البرمجات فوجدت أن أفضل موضوع في هذا الشأن هو هندسة البرمجيات Software Engineering / Génie Logiciel.

طبعا المحاضرة ستكون عبارة عن رحلة استطلاعية لهذا التخصص الرائع، سنلقي عليه نظرة من فوق: بانوراما، من خلالها نفهم ماهية هذا التخصص وأهميته البالغة في عملية صناعة البرمجيات.

قبل ذلك اسمحوا لي أن أستعرض أمامكم مجموعة من المعلومات التقنية كتمهيد لهندسة البرمجيات.

نشرت مجلة سيو ماغازين CEO Magazine في شهر ديسمبر من عام 1998 خبرا مفاده أن مؤسسة ستانديش غروب، وهي مجموعة مهتمة بدراسة طرق إدارة المشاريع البرمجية Software Project Man-agement ways كما تعرف نفسها على موقعها الرسمي.

هذه المجموعة (ستانديش غروب) أشارت في بحث لها أن 46% من المشاريع البرمجية تجاوزت المدة المتفق عليها، بل وتجاوزت حتى الميزانية المخصصة لها، بل ولم تلبي حاجيات العملاء..

يعني أن المؤسسات البرمجية التي أشرفت على هذه المشاريع لم تفلح أن توفق بين الميزانية والوقت المخطط لهما وبين الميزانية والوقت الفعلي الذي استغرقته عملية صناعة البرمجيات.

وذكرت نفس المجلة (سيو ماغازين) أن مجموعة ستانديش غروب قد أحصت كذلك نسبة 28% من المشاريع التي فشلت تماما، ولم يقدر لها لا النجاح ولا حتى أن يتم إنتاجها بصرف النظر عن الميزانية والوقت الذي تم التخطيط له، يعني خلاص فشلت بشكل كامل.

وذكرت المجموعة أن 24% فقط من المشاريع البرمجية هي التي كتب لها النجاح.

دقق معي جيدا، نحن الآن لا نتحدث عن مبرمجين هواة أو عن مؤسسات برمجية صغيرة، بل نتحدث عن مؤسسات كبيرة وتضم مهندسين وأطر عليا ومع ذلك قدر لمشاريعها أن تموت في المهد، أو تكتمل بشكل شائه دون احترام للميزانية المخصصة لها أو الوقت المزمع أن تستغرقه.

فيما يلي رسم بياني يعرض نفس البحث الذي تقوم به مجموعة Standish Group لكن لعام 2012:

نتيجة أبحاث مشروع CHAOS التابع لـ Standish Group
Project resolution from 2012 CHAOS research.

وفيما يلي نتائج الأبحاث التي تمت بدء من عام 2004 وانتهاء بالعام 2012:

نسبة حالات المشاريع البرمجية من عام 2004 إلى عام 2012
حسب Standish Group

وذكرت المؤسسة العسكرية الأمريكية US Army أنها أنفقت على مشروع برمجي ما قيمته 1 بليون دولار (مليار دولار) زيادة على الميزانية المخصصه له. ومع ذلك مايزال هذا المشروع غير شغال.

قبل أحداث الحادي عشر من سبتمبر من عام 2001 بالولايات المتحدة الأمريكية وتحديدا عام 2000. قام المكتب الاتحادي للتحقيقات The US Federal Bureau of Investigation ((FBI بتكليف المؤسسة الدولية للتطبيقات العلمية Science Applications International Corp (SAIC) بتطوير برنامج اسمه Virtual Case File software (VCF).

دور هذا البرنامج هو إدارة ملفات القضايا الكترونيا. بمعنى أن أي مستخدم له صلاحيات في هذا النظام يمكن له أن يطلع على معلومات القضايا.

تم تحديد ثلاث سنوات غطاء زمنيا لإنجاز هذا المشروع. أي في عام 2003 يتعين أن يكون النظام كاملا بكل ما تم الاتفاق عليه من مهام وعمليات.

انتهت المدة المحددة. وتم تسليم نسخة من هذا النظام للمكتب الاتحادي للتحقيقات The US Federal Bureau of Investigation في شهر دجنبر من عام 2004. وكانت هذه النسخة علاوة على التأخير الزمني مليئة بالعيوب والنقائص. من بين العيوب التي تم رصدها في نسخة النظام المسلمة:

أن ما تم إنجازه لا يمثل سوى واحد من عشرة من البرنامج الذي تم الاتفاق عليه

ما يقارب 170 مليون دولار ضاعت هباء منثورا.

في شهر أبريل من عام 2005 توقفت عملية تطوير المشروع وتم التصريح بفشله.

قصص كثيرة جدا تحكي فشل مشاريع برمجية كبرى يمكنكم الاطلاع عليها على الانترنت. فقط ادخلوا إلى أي محرك بحث. غوغل مثلا. وقوموا بكتابة: Software Fail-ure stories وستشاهدون النتائج.

هذه القصص تحيلنا مباشرة إلى سؤال جوهري وهو: ماهو سبب أو ماهي أسباب فشل المشاريع البرمجية؟

نفس السؤال أطرحه عليكم: في نظركم ماهو السبب أو ماهي الأسباب التي تؤدي إلى فشل المشاريع البرمجية ؟

من يجيب؟

أسباب فشل المشاريع

يمكننا أن نجمل أسباب فشل المشاريع البرمجية في النقاط التالية:

1. أهداف المشروع غير حقيقة أو غير واضحة (في الغالب الأعم يكون العميل بعيدا عن مجال صناعة البرمجيات. لذلك لا تتوقع منه أن يخاطبك بأسلوب تقني كأن يقول لك: أريد من البرنامج أن يفعل كذا وكذا. وأن تكون به شاشات لإدخال كذا وكذا. أن تكون التقارير من نوع Jasper Report. أن تكون قاعدة البيانات أوراكل... لا تتوقع منه ذلك. لأنه سيخاطبك بلغة عادية بعيدة عن الحقل البرمجي. وربما يذكر في طلباته أشياء بعيدة عن الواقع

البرمجي، وأن يطلب أشياء لا يمكن إنجازها برمجيا. فلا توافق على كل شيء وإلا لفشل مشروعك بسبب عدم تحقيق جميع رغبات العميل التي ذكرها ووافقته عليها مع أنها غير مهيئة أن تكون بصيغة برمجية)

2. سوء تقدير الموارد اللازمة من ميزانية ووقت (الوقت وما أدراك ما الوقت. لا يمكنك السيطرة عليه. لكن يمكنك إدارته. ولكي لا يواجه مشروعك البرمجي الفشل. قم بإعطاء مرحلة التخطيط Planning وقتا كافيا. لا تبدأ مباشرة في كتابة الكود حتى وإن كنت متيقنا من أن مايدور برأسك هو نفسه ما يريده العميل. خذ كل وقتك في مرحلة التخطيط لتفهم النظام ومتطلباته بشكل دقيق. هذا الأمر سيسهل عليك كل العمليات التي تأتي بعده من كتابة كود واختبار وتحزيم وغيرهم. أما بخصوص الميزانية. فلا تعتقد أنك إن خفضت تكلفة النظام ستربح العميل. لأن مايهمه في الأخير هو نظام برمجي ينقص عليه المصاريف أو يزيد له المداخيل.

البعض منا لديه أفكار خاطئة حول تقييم تكلفة المشروع. ويظن أنه لو ذكر ثمنا مناسبا سيرفض العميل بحجة أن الثمن مرتفع. فيقوم بتخفيض المبلغ مخافة أن يفر العميل من بين يديه. وهذا خطأ فادح. قم بدراسة مشروعك وصمم له الحل بهدوء وخذ وقتك الكافي في ذلك. ثم قم بتحديد الميزانية اللازمة لإنجاز المشروع في المدة الزمنية التي حسبتها ولا تفكر في أي شيء آخر وإلا فإن مشروعك سيفشل وستدفع من جيبك لتعويض نقص الميزانية. ركز دائما على مرحلة التخطيط لأنها أول مرحلة فإن وضعتها بشكل سليم سلم مشروعك من الفشل. وهنا أستحضر مقولة بنيامين فرانكلين وهو فيلسوف أمريكي: إذا فشلت في التخطيط فقد خططت للفشل If you fail to plan, you are planning to fail)

3. متطلبات النظام تم تحديدها بشكل سيء (تأكد جيدا أنك لو أرسلت نفس العميل إلى مجموعة من المؤسسات البرمجية. فكل واحدة ستقوم بكتابة المتطلبات بشكل مختلف عن نظيراتها. إذن فالعملية تستلزم الدقة والتركيز في كل كلام العميل وصياغته بشكل يتلاءم مع طبيعة المشاريع البرمجية. كما يجب عليك أن تقدم اقتراحات للعميل لتحسين النظام. وإعطاء بديل لأهدافه الخيالية التي لا يمكن إنجازها برمجيا. بمعنى لو قال لك: أريد من البرنامج أن يقرأ الوثائق الورقية بواسطة الطابعة. تقول له: الطابعة دورها هو إخراج الوثائق فقط. لكن يمكننا قراءة الوثائق الورقية عبر جهاز الماسح الضوئي Scan-ner)

في هذه المرحلة ستحتاج إلى تصميم بعض النماذج من نوع Use case إن كنت تستخدم لغة UML. من أجل تحديد المهام Actions التي يتيحها النظام ومن هم الفاعلون Actors الذين يقومون بها.)

4. تقارير ناقصة حول المشروع (عند الانتقال من مرحلة إلى مرحلة أخرى في المشروع يفترض إعداد تقارير. مثلا عند أول مرحلة في المشروع والمعروفة باسم مرحلة التخطيط Planning علينا وضع تقرير يحتوي على دراسة الجدوى الخاصة بالمشروع فإن كان هذا التقرير ناقصا فإن ذلك سيؤثر بالسلب على المرحلة الموالية المعروفة باسم مرحلة التحليل Analysis وهكذا دواليك. فأي تقرير ناقص أو يحتوي على معطيات مغلوطة سيفضي بالمشروع إلى شط الفشل)

5. عدم الأخذ بعين الاعتبار تعقيدات النظام System Complexity (عليك تحديد صعوبة النظام الذي تود إنجازه. ماهي الأجزاء الصعبة في هذا النظام؟ كيف سأقوم بإنجازها؟ وكيف يمكنني إصلاحها إن أخطأت في إنجازها؟ مع العلم أن تصحيح الكود أصعب من كتابته. ومن العبارات الشهيرة في مجال دراسة تعقيدات الأنظمة "قراءة الكود أصعب من كتابته It's harder to read code than write it.

تعقيدات النظام أو صعوبة النظام من المواد الأساسية في هندسة البرمجية والتي تعنى بحساب ودراسة درجة صعوبة النظام. لكي نعرف النظام بشكل عميق. وتشمل تعقيدات النظام الجانب المادي للبرنامج بمعنى كم هي حاجة البرنامج إلى الموارد المادية مثلا: ماهو حجم الذاكرة الذي يتطلبه البرنامج ليشتغل؟ وتشمل تعقيدات النظام كذلك الجانب البشري بمعنى كم هي الموارد البشرية التي سيحتاجها البانرمج. مثلا: عدد المبرمجين. عدد الأفراد الذين سيجرون الاختبار على البرنامج.

لذلك يهتم المهندسون البرمجيون بحساب وقياس تعقيدات النظام. لأنه بحساب كل عملية ننفي احتماليات الخطأ والشك. التعقيدات تحسب. الجودة تحسب. إعادة الاستخدام تحسب. كل عملية من عمليات هندسة البرمجيات تحسب.

وأذكر هنا مقولة الفيزيائي البريطاني ذي الأصول الايرلندية ويليام تومسون الشهير باللورد كيلڤان إذ يقول:

> حينما تستطيع حساب ماتتكلم عنه والتعبير عنه بالأرقام، يمكنك معرفة بعض الأشياء حوله، لكن إن لم تستطع حساب ماتتكلم عنه والتعبير عنه بالأرقام، فمعرفتك به فقيرة جدا وغير كافية.

6. ضعف التواصل بين العملاء والمطورين والمستخدمين النهائيين (على المبرمجين الذين يشتغلون على نفس المشروع أن يكونوا على تواصل فيما بينهم، ليوضحوا المهام لبعضهم البعض، حتى لا يكرر أحدهم عمل الآخر، أو يظن أحدهم أن جزئية ما تدخل في تكليفات مبرمج آخر.
أيضا ينبغي أن يكون هنالك تواصل مع العميل ومع مستخدمي النظام خاصة في مرحلة جمع المتطلبات من أجل بناء نظام يناسبهم ويلبي رغباتهم.)

7. استخدام تكنولوجيا لا تفي بالغرض (لغة برمجية لا تلبي الغرض، أجهزة موصولة بالنظام لا تشتغل بكفاءة جيدة...)

8. ممارسات برمجية غير مسؤولة ولا تتعامل مع المشروع بالجدية المطلوبة (بعد تحديد الميزانية والوقت الخاصين بالمشروع، على الفريق البرمجي أن ينكب على العمل بجدية دون تهاون أو تخاذل من أجل ضمان تسليم المشروع في المدة المحددة، لأن كل زيادة زمنية هي نقصان في الميزانية، وخسارة للمؤسسة، وسقوط للسمعة.)

9. سوء إدارة المشروع (إذا ما حدد مدير المشروع Project Manager لفريق البرمجة موعدا نهائيا غير واقعي لتسليم العمل، فإن ذلك سيؤثر سلبا على المشروع، لأن العمل لن ينجز في المدة المحددة وبالتالي اهتزاز نفسية الفريق البرمجي مظنة أنهم ليسوا أكفاء زيادة على تأخر موعد التسليم، بينما يكون الأمر جيدا لو قام مدير المشروع بتحديد مدة زمنية معقولة، أو قام بتقسيم العمل إلى أجزاء صغيرة وتحديد مدة لكل جزء.
نفس الكلام ينطبق على طريقة العمل، حينما يكون الفريق البرمجي يشتغل بشكل غير مسؤول، وليست لديهم أهداف محددة، فطبيعي جدا ألا يتم إنجاز المطلوب بالشكل المطلوب في الوقت المطلوب بالثمن المطلوب.)

10. عدم تلبية حاجيات العميل: (الغرض من إنشاء المشروع هو تلبية متطلبات وحاجيات العميل، فمنطقيا أي برنامج لا يحترم هذه النقطة فهو حتما برنامج فاشل ولا يتناطح فيها عنزان ولا يختلف فيها اثنان كما يقال.

هندسة البرمجيات

هندسة البرمجيات هي مجمل الأنشطة والعمليات التي من شأنها تصور النظام، وتوظيف تقنيات ووسائل من أجل الرفع من جودة وكفاءة عملية إنتاج وصناعة البرمجيات وكذلك عمليات التتبع والمراقبة، وتضمن لنا هندسة البرمجيات احترام أربعة معايير أساسية من معايير عملية صناعة البرمجيات وهي:

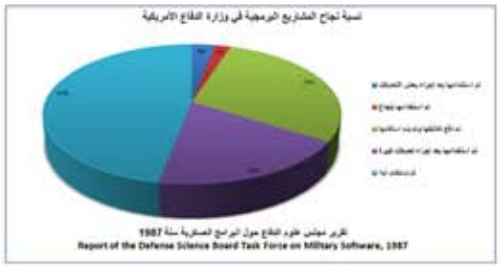
1. ينبغي أن يشمل المشروع جميع الوظائف والمهام التي اقترحها العميل
2. ينبغي الأخذ بعين الاعتبار جودة المشروع
3. ينبغي احترام التكلفة المالية التي تم

تحديدها قبل خوض غمار المشروع

4. ينبغي احترام المدة الزمنية التي تم الاتفاق عليها

تاريخ هندسة البرمجيات

ظهرت هندسة البرمجيات في سبعينيات القرن المنصرم 1970 من أجل حل أزمة البرمجيات Software Crisis / La Crise du Logiciel. حيث عرفت عملية تطوير البرمجيات مشاكل جوهرية متمثلة في عدم احترام مواعيد التسليم، وعدم احترام الميزانية المخصصة للمشاريع، وأن هاته المشاريع لا تفي بالغرض ولا تلبي احتياجات العميل، إضافة إلى أنها تكون صعبة الاستخدام وصعبة الصيانة والتصحيح، فضلا عن إمكانية إعادة تطويرها وتحسينها.



أهداف هندسة البرمجيات

تلبية حاجيات العميل meeting user's needs.

تكلفة إنتاج منخفضة low cost of production.

كفاءة عالية high performance.

قابلية النقل والاشتغال من جهاز إلى جهاز آخر portability.

جودة عالية مما يضمن الاتكالية والاعتماد على النظام reliability.

احترام موعد التسليم delivery on time.

يمكننا اعتبار كل هدف من الأهداف السابقة إكراها وتحديا في عملية صناعة البرمجيات، وغياب عنصر من تلكم العناصر قد يفضي إلى إفشال المشروع، مثلا عدم تلبية حاجيات ومتطلبات العميل=برنامج فاشل، عدم احترام الوقت أو الميزانية=برنامج فاشل، غياب الجودة أو احتواء البرنامج على أخطاء =برنامج فاشل...إلخ.

دورة حياة البرامج

حينما نتحدث عن دورة حياة البرامج System Development Life Cycle، قد يتبادر إلى أذهاننا لأول وهلة أن الأمر يتعلق بإنشاء أول ملف من ملفات البرنامج وتنتهي هذه الدورة عند حذف آخر ملف من ملفات البرنامج، وهذا ليس صحيحا لأن المقصود بدورة حياة البرامج أو دورة حياة النظم هو بداية طلب النظام من طرف المستخدم وانتهاء بتسليمه إياه.

حينما نقول العميل Customer / Client أو المستخدم User / Utilisateur فنحن نتحدث عن نفس الفاعل الذي سيستفيد من خدمات النظام، حتى وإن كان العميل Customer غالبا ما يشير إلى الشخص الذي سيدفع تكلفة النظام / ممول النظام.

ومصطلح المستخدم يشير إلى الشخص الذي سيستعمل النظام / مستخدم النظام. إلا أنه في هندسة البرمجيات يتم اعتبارهما واحدا في الغالب الأعم لذلك ستجد في المواد الإنجليزية يتم التعبير عنهما بنفس المسمى StackeHolder.

تمر عملية صناعة البرمجيات من مجموعة من المراحل التي تعرف بدورة حياة النظم البرمجية Sytem Development Life Cycle. والمتمثلة فيما يلي:

1. التخطيط Planning / Planification: خلال هذه المرحلة ينبغي أن نحدد بنية المشروع. والهدف من إنشائه. ودراسة المتطلبات الاقتصادية. والزمنية. والتنظيمية. لكي نقوم بوضع دراسة جدوى سليمة للمشروع.

2. التحليل Analysis / Analyse: خلال هذه المرحلة ينبغي أن نعرف من الذي سيستخدم هذا النظام. ومتى يمكن لكل مستخدم أن يستخدم النظام. وأين يستطيع كل مستخدم أن يتعامل مع النظام حسب طبيعته. في هذه المرحلة علينا جمع المعلومات بدقة وتحديد المهام والمتطلبات بناء على دراسة الجدوى التي حصلنا عليها في مرحلة التخطيط.

3. التصميم Design / Conception: خلال هذه المرحلة يتم تصميم النظام بمختلف مكوناته. من واجهات وملفات وقواعد بيانات. والتصميم ليس مقتصرا على الشاشات والواجهات الرسومية GUI. وإنما يعني كذلك التصميم العملي والوظيفي Functional Design. بمعنى تصميم طرق التعامل مع البرنامج واختيار أفضل السبل لإنجازه بالشكل المطلوب وبجودة عالية. إذن فالتصميم يجمع بين الشكل والوظيفة Ap-pearance and functionality.

يلعب التصميم في عملية صناعة البرمجيات نفس الدور الذي يلعبه في مجالات أخرى. فكما أن الميكانيكي لا يستطيع تجميع قطع غيار سيارة لم يسبق له أن تعامل معها إلا من خلال دليل تجميع. وكما أن صاحب محل صيانة الحواسيب لا يستطيع تركيب جهاز جديد عليه دون دليل تجميع. فكذلك شأن إنشاء برنامج دون العبور من مرحلة التصميم. فالتصميم هو ذلك التصور الذي سيقودك إلى بناء برنامج يلبي حاجيات العميل. فإن أتيت وبدأت بكتابة الكود مباشرة فذلك أشبه بشكل كبير بتجميع قطع لعبة Puzzle لم تشاهد صورتها الأصلية فنسبة الفشل إذن كبيرة جدا.

أيضا خلال هذه المرحلة. يفضل تقسيم النظام إلى أنظمة جزئية Sub-Systems لكي يتم تقسيم المشاكل الرئيسية إلى مشاكل فرعية sub-problems ليسهل حلها. لأن التعامل مع مشاكل صغيرة أهون من التعامل مع المشاكل الكبيرة. فلكي نتمكن من تجاوز المشاكل المعقدة نقوم بتفكيكها وتفتيتها.

4. الترميز Coding / Codage: يتوجب علينا في هذه المرحلة صياغة الحلول والنماذج والمخططات التي حصلنا عليها في مرحلة التصميم Design بشكل برمجي، عبر الاعتماد على إحدى لغات البرمجة المرنة التي بإمكانها أن تسع كل ما تم تسجيله من متطلبات وأغراض.

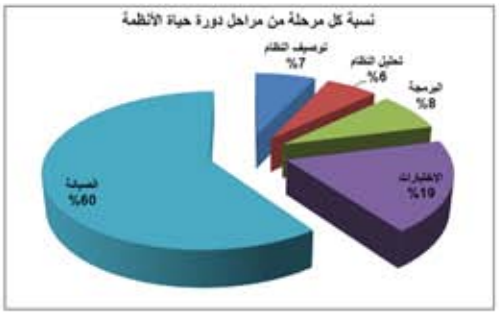
في غضون مرحلة الترميز Coding يتم القيام بمجموعة من الاختبارات الوحدوية التي تعنى بكل وحدة على حدة بحيث يتم اختبار كل وظيفة Function وكل فئة Class وكل مكون Component ليتم التحقق من صحة الشفرات المكتوبة، ويسمى هذا النوع من الاختبارات ب: الاختيارات الوحدوية Les tests unitaires / unit testing

5. اختبارات التكاملية: عند الانتهاء من برمجة واختبار جميع وحدات المشروع من قبل فريق العمل، يتم تجميع هذه الوحدات من أجل الحصول على النظام بشكله الكامل، ويتم إجراء مجموعة من الاختبارات عليه للتأكد من استيفائه لكافة المتطلبات التي تم طرحها بواسطة العميل، وليتم التأكد من خلو النظام من أية مشاكل أو أخطاء.

6. التوثيق Documentation: تعتبر هذه العملية من أهم العمليات التي لا ينبغي الاستهانة بها أثناء الانكباب على تطوير نظام ما، لأنها توثق لكافة مراحل صياغة الحل البرمجي وتعرض كافة المشاكل التي ظهرت في مختلف مراحل دورة النظام وحلولها، مما يسهل عملية الصيانة وإعادة الاستخدام وتطوير النظام، لأن كل جزئية لها توثيق خاص بها مما يضمن سهولة متابعة النظام واستكناه تفاصيله وبالتالي تسهيل السيطرة عليه.

غياب عملية التوثيق في المشاريع البرمجية الكبيرة قد يتسبب في توقيف عملية التطوير لأن المشروع سيصل إلى مرحلة متقدمة يصبح فيها من الصعب صيانة ومعالجة ما مضى منه.

الصيانة Maintenance: تأخذ هذه المرحلة الحيز الأكبر من دورة حياة النظام، في غضونها يتم تصحيح النظام من كافة الأخطاء والشوائب، وإضافة تحسينات وزيادات تساهم في رفع جودة النظام.



خاتمة

أرجو من الله العلي القدير أن أكون قد وفقت في تقديم هذه المحاضرة بشكل يضمن إفادتكم بشيء من المعلومات ولو بالنزر القليل، وأرجو منه تبارك وتعالى أن يجعل هذا العمل خالصا لوجهه الكريم وألا يجعل فيه للنفس ولا للشيطان حظا.

أكتفي بهذا القدر، أدعو لكم بالتوفيق والسداد ودام لكم البشر والفرح.

سبحانك اللهم وبحمدك، أشهد أن لا إله إلا أنت، أستغفرك وأتوب إليك

قراءة في كتاب C# 2008 Programmer's Reference

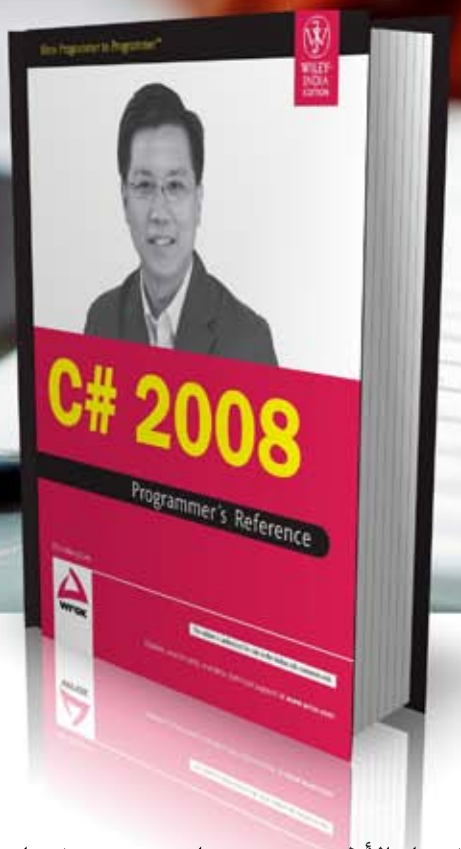
للكاتب: Wei-Meng Lee

سنتحدث اليوم عن كتاب محترم، كان له علي – بعد الله عز وجل – فضل عظيم في استيعاب عدة أمور برمجية في نطاق الدوت نيت عموما وفي لغة سي شارب خصوصا.
كتاب يجمع بين الشرح النظري الماتع، والعرض العملي النافع، يتميز بالدقة في شرح المفاهيم المتناولة دون الخروج عنها، وكذلك بشموليته لكافة تفاصيل لغة سي شارب تقريبا في إصدارها الثالث الذي رافق نسخة الفيجوال ستوديو 2008.

بقلم: خالد السعداني

قراءة في كتاب C# 2008 Programmer's Reference

خالد السعداني



سنتحدث اليوم عن كتاب محترم، كان له علي – بعد الله عز وجل – فضل عظيم في استيعاب عدة أمور برمجية في نطاق الدوت نيت عموما وفي لغة سي شارب خصوصا.

كتاب يجمع بين الشرح النظري الماتع، والعرض العملي النافع، يتميز بالدقة في شرح المفاهيم المتناولة دون الخروج عنها، وكذلك بشموليته لكافة تفاصيل لغة سي شارب تقريبا في إصدارها الثالث الذي رافق نسخة الفيجوال ستوديو 2008.

الكتاب من الحجم الضخم، يشمل بين دفتيه ثمان مائة وأربعة وأربعون صفحة (844)، مقسم إلى ثلاثة أقسام، قسم خاص بأساسيات لغة سي شارب، وقسم ثان خاص ببرمجة التطبيقات المتنوعة (ويندوز، ويب، سيلفر لايت، موبايل، خدمات) موزع على عشرين فصل أصغرها الفصل الأول وهو يمتد على تسع صفحات، وأكبرها الفصل ما قبل الأخير (التاسع عشر) وهو يمتد على سبعة وسبعين صفحة، أما القسم الثالث فهو ملحق يعرض قاموس الكلمات المحجوزة في لغة سي شارب، وكيفية فحص فئات الدوت نيت عبر أداة Object Browser، إضافة إلى كيفية توثيق التطبيقات Documentation of C# Applications، أما عنوان الكتاب فقد اختار له صاحبه: "مرجع المبرمجين بلغة سي شارب 2008".

قبل أن نتطرق إلى محتويات فصول الكتاب، دعونا نتعرف على من أبدعها وكتبها: المبدع واي مين لي Wei-Meng Lee.

حول الكاتب:

واي مينغ لي، كاتب تقني بارز، حاصل على شهادة أكثر مبرمجي ميكروسوفت تقييما MVP (Most Valuable Programmers)، مؤسس موقع www.learn2develop.net الخاص بتدريس آخر تقنيات ميكروسوفت و أبل وتطبيقات أندرويد.
يكتب واي مينغ لي في عدة منابر إعلامية مثل مجلة Code Magazine و مجلة asp.net PRO Magazine، كما أنه يكتب لفائدة مؤسسة O'Reilly Network و DevX.com.

ما الذي يحتويه هذا الكتاب؟

1. يتطرق الفصل الأول من الكتاب إلى إطار العمل دوت نيت فريموورك، باعتباره ورشة العمل التي سيحتاجها المبرمج لإنشاء المشاريع بلغة سي شارب، ويتعرض إلى المكونين الأساسيين للدوت نيت فريموورك، وهما الآلة الافتراضية CLR (Common Language Runtime) التي تسمح بتنفيذ البرامج المكتوبة بلغات الدوت نيت، ومكتبة الفئات FCL (Framework Class Library) التي تشمل كافة مكتبات الدوت نيت التي يحتاجها المبرمج.
ويتحدث الكتاب كذلك عن اللغة الوسيطة MSIL (Microsoft Intermediate Language) الناتجة عن عملية الترجمة والتي تسمى كذلك CIL (Common Intermediate Language) ولها أسماء أخرى مثل IL اختصارا ل Intermediate Language.
2. الفصل الثاني يتناول برنامج الفيجوال ستوديو 2008 باعتباره البرنامج الذي يسمح بإنشاء البرامج بلغات الدوت نيت مثل الفيجوال بيسك وسي شارب وغيرها...
في هذا الفصل سوف تتعلم كيف تقوم بالتحكم في إعدادات الفيجوال ستوديو، والقيام بعمليات التصحيح Debug ونقاط التوقف Breakpoints وإجراء الاختبارات الأحادية Unit Testing.
3. الفصل الثالث يتحدث عن أساسيات لغة سي شارب، مثل الكلمات المحجوزة Keywords، المتغيرات Variables، الثوابت Constants، التعليقات Comments، أنواع البيانات Data Types، البينات الشرطية والتكرارية Flow Control and Looping، الروابط بأنواعها Operators، الموجهات Preprocessor Directives، وغير ذلك من المفاهيم الأساسية في لغة سي شارب.

4. الفصل الرابع يدخل إلى البرمجة الكائنية OOP، فيتعرض للفئات Classes، الكائنات Objects، وغير ذلك من أصول البرمجة الموجهة كائنيا.
5. أما الفصل الخامس فموضوعه هو الواجهات Interfaces، بحيث يعرض كيفية إنشائها واستخدام عناصرها في الفئات، بالإضافة إلى كيفية استخدام عدة واجهات في فئة واحدة لتعويض الوراثة المتعددة : Multiple Inheritance، ويتعرض هذا الفصل أيضا إلى أمور أخرى مثل

الروابط is و as، وغير ذلك..

6. في الفصل السادس يتحدث واي مينغ لي عن مفهوم الوراثة Inheritance، وما يترتب عنها من مفاهيم أخرى مثل مفهوم الفئات المجردة abstract classes التي تمنع عملية الاستنساخ وتسمح فقط بالوراثة، ومفهوم الفئات المغلقة Sealed Classes التي تسمح بالاستنساخ وتمنع الوراثة، ويتحدث الكاتب أيضا عن مفهوم زيادة التحميل Overloading وعن الوظائف الافتراضية Virtual Methods.

7. هذا الفصل مقسم إلى جزأين، جزء أول يتحدث عن المفوضات Delegates وكيفية إنشائها واستخدامها ومفهوم التفويض المتعدد Multicast Delegates.

وفي الجزء الثاني يتطرق الفصل إلى مفهوم الأحداث Events، وكيفية اقتناصها وتطبيقها، والفرق بينها وبين المفوضات.

8. الفصل الثامن يشرح كيفية التعامل مع النصوص String Manipulations والعبارات النظامية Regular Expressions.

9. يتطرق الفصل التاسع إلى الأنواع التشعيبية Generics Types، أهميتها وكيفية استخدامها.

10. الفصل العاشر خاص بالتعامل مع المسارات المتعددة Multi-Threading، وكيفية بدء المسار وإلغائه وتمرير البرامترات إليها وغير ذلك من المفاهيم المتقدمة المتعلقة بالمسارات.

11. يتناول هذا الفصل كل ما له علاقة بالملفات Files وتدفق البيانات Streams، وما يترتب عن ذلك من عمليات كالتعامل مع الوحدات والمجلدات، وتشفير البيانات، وضغطها والتعامل مع البيانات الثنائية.

12. يتحدث هذا الفصل عن إدارة الاستثناءات التي تقع في البرامج، وكيفية تفاديها عبر استخدام آليات معالجة الاستثناءات مثل try catch و throw.

13. في هذا الفصل ستتعرف على مفهوم مجموعات البيانات Collections والمصفوفات Arrays، وكيفية التعامل معها وإجراء العمليات عليها، وفي ختام الفصل سوف تتعرف على باقي بنيات البيانات Data Structures مثل Dictionary، Stacks و Queues.

14. أما هذا الفصل فهو مخصص لتقنية Linq الخاصة بالاستعلام عن البيانات والكائنات بلغة سي شارب، وسينتقل بك الكاتب إلى أبرز دواعي استخدامات تقنية Linq مع مصدر البيانات Data-set، مع ملفات XML، مع لغة SQL ومع الكائنات Objects.

15. يركز هذا الفصل تحديدا على مفهوم المجمعات Assemblies باعتبارها الوحدة الناتجة عن مشاريع الدوت نيت، عارضا بنيتها، وفحص محتواها، والفرق بين المجمعات الخاصة والمشتركة Pri-vate and Shared Assemblies.

16. الفصل السادس عشر سينقلك إلى فضاء برمجة تطبيقات الويندوز، وسيستعرض معك مثالا قويا يوضح كيفية رفع الملفات والمجلدات بواسطة بروتوكول FTP، بعد ذلك سينقلك إلى

عالم الطباعة وكيفية استخدام أدوات الطباعة. وفي الجزء الأخير من هذا الفصل سوف تتعلم كيف تقوم بنشر البرنامج وإعداد حزمة بواسطة الوسيط ClickOnce لكي تقوم بتثبيت برامجك المكتوبة بلغات الدوت نيت على حواسيب أخرى.

17. في هذا الفصل الرائع سيأخذك واي مينغ لي إلى عالم برمجة الويب مع تقنية asp.net. سيشرح لك أولا كيف تعمل هذه التقنية وما الذي ستحتاجه لاستخدامها. وكيفية الربط بين صفحات الويب وبين مصادر البيانات بواسطة تقنية Linq to SQL. إضافة إلى شرح تقنية أجاكس عبر التطرق إلى بعض أدواتها الجاهزة التي تأتي مع مكتبة Ajax Toolkit.

18. في هذا الفصل سوف يشرح لك الكاتب كيف تقوم بإنشاء تطبيقات للهواتف Mobile Appli-cations الخاصة ببيئة الويندوز عبر استخدام .Net Compact Framework. وكيفية إعداد بيئة التطوير. ثم سيأخذك مع مثال عملي يشرح كيفية إنشاء تطبيق موبايل يقرأ RSS بدء من التطوير وانتهاء بالتحزيم.

19. يتناول هذا الفصل تقنية Silverlight التي تهتم بإنشاء تطبيقات انترنت غنية Rich Internet Applications والمعروفة اختصارا ب RIA. بحيث تسمح باستعمال التأثيرات المرئية وتقنيات الرسم كما هو الحال مع أدوبي فلاش.

ويتطرق هذا الفصل أيضا إلى كيفية تصميم صفحات Silverlight بواسطة تقنية XAML.

20. في هذا الفصل ينقلك الكتاب إلى تقنيات الخدمات. وتحديدا مع تقنية ميكروسوفت الجديدة WCF التي تعد اختصارا ل Windows Communication Foundation. سيشرح لك هذا الفصل كيف تقوم بإنشاء خدمات WCF وكيفية استعمالها.

{فلمسوه بأيديهم}

سورة الأنعام والإلحاد المادي

كلما أقبلت على القرآن الكريم متعلما لاحت لك الأنوار الربانية كاشفة لك كل لبس و غموض، فتستشعر لذة الأنس بالله تلامس قلبك ثم تفيض لتملأ صدرك كله سعادة وبهجة، إنه نور القرآن الذي تشفى به الصدور.

بقلم: خالد السعداني

{فلمسوه بأيديهم}

سورة الأنعام والإلحاد المادي

خالد السعداني

كلما أقبلت على القرآن الكريم متعلما لاحت لك الأنوار الربانية كاشفة لك كل لبس و غموض، فتستشعر لذة الأنس بالله تلامس قلبك ثم تفيض لتملأ صدرك كله سعادة وبهجة، إنه نور القرآن الذي تشفى به الصدور.

انقسم الناس أمام كلام الله عز وجل ثلاثة أقسام، أما القسم الأول فهم المؤمنون، الذين آمنوا وصدقوا بكل ما جاء به القرآن الكريم فلم يمتروا في ذلك ولم يرتابوا، وأما القسم الثاني فهم المشركون، وهم مؤمنون بالله لكنهم ظنوا في غيره الندية والمثل، فتوجهوا بدعائهم لغير الله، وأقبلوا على شركائهم يسألونهم كشف الضر وجلب النفع وقبول الأعمال، متعامين عن صريح القرآن الكريم الذي يصدع أن لا شريك ولا ند لله، وأن من اجترح معصية الشرك فهو على خطر عظيم إذ صرح الله تبارك وتعالى أنه يغفر الذنوب جميعا إلا ذنب الشرك "قُلْ يَا عِبَادِيَ الَّذِينَ أَسْرَفُوا عَلَى أَنفُسِهِمْ لَا تَقْنَطُوا مِن رَّحْمَةِ اللَّهِ إِنَّ اللَّهَ يَغْفِرُ الذُّنُوبَ جَمِيعًا إِنَّهُ هُوَ الْغَفُورُ الرَّحِيمُ [الزمر:53]".

أما القسم الثالث، فهم الكفار، وهم ليسوا بمؤمنين ولا مشركين، إذ ينكرون وجود الله أساسا وكل ما يترتب على وجوده من دين ونبوة وكتب وغير ذلك، أو يؤمنون بوجود الله لكنهم يظنون الألوهية في غيره كما هو حال النصارى والذين ورد كفرهم في القرآن الكريم، وذلك لاعتقادهم الألوهية في المسيح عيسى عليه السلام.

القسم الثاني والثالث عرفا انتشارا واسعا هذه الأيام على مختلف الوسائط الإعلامية، وشرعا يروجان لضلالاتهما ويزعمان أنهما أهل الحق وأن أهل الإيمان في ضلال بعيد.

سأتحدث عن القسم الثالث أولا وهم الكفار، الذين يكفرون بالله عز وجل أو يظنون الألوهية في غيره، وهذا النوع يكفر غالبا بسبب غياب الدليل المادي على وجود الله حسب زعمهم، متغاضين عن حقيقة أن الإيمان ظاهرة غيبية بها يتمايز المؤمنون من الكافرين، فمن خشع لله قذف في قلبه نورا يغنيه عن كل دليل مادي، أما الخراصون فحتى لو جاءهم المولى تبارك وتعالى بالأدلة المادية فإنهم لن يؤمنوا أبدا بسبب تمركزهم حول ذواتهم وبسبب زهوهم بعقولهم و عجبهم بأفكارهم، وفيهم قال الله تعالى: "وَلَوْ نَزَّلْنَا عَلَيْكَ كِتَابًا فِي قِرْطَاسٍ فَلَمَسُوهُ بِأَيْدِيهِمْ لَقَالَ الَّذِينَ كَفَرُوا إِنْ هَذَا إِلا سِحْرٌ مُبِينٌ(7) - الأنعام"

هذا النوع نجده كثير الاستهزاء والتكذيب والجدال وهي الأوصاف التي ذكرتها سورة الأنعام : "وَلَقَدْ اسْتُهْزِئَ بِرُسُلٍ مِنْ قَبْلِكَ فَحَاقَ بِالَّذِينَ سَخِرُوا مِنْهُمْ مَا كَانُوا بِهِ يَسْتَهْزِئُونَ(10)قُلْ سِيرُوا فِي الأَرْضِ ثُمَّ انظُرُوا كَيْفَ كَانَ عَاقِبَةُ الْمُكَذِّبِينَ(11) - الأنعام" وفي الآية "وَإِنْ يَرَوْا كُلَّ أية لا يُؤْمِنُوا بِهَا حَتَّى إِذَا جَاءُوكَ يُجَادِلُونَكَ يَقُولُ الَّذِينَ كَفَرُوا إِنْ هَذَا إِلا أَسَاطِيرُ الأَوَّلِينَ" تمعن جيدا كيف وصفهم الله تعالى، فهم يسخرون ويستهزؤون من كل من يدعوهم إلى عبادة الله، ويكذبون بالغيب ورواتبه، وحينما يأتون لمجادلتك، فتحيلهم إلى القرآن الكريم يقولون لك: إن هذا إلا أساطير الأولين، وهي نفس الجملة التي يرددها الملحدون كلما دعوا إلى الإسلام، بعد هذه الآية الكريمة تأتي آية أخرى تعطيك وصفا دقيقا للملحدين، فهم لا يحتفظون بكفرهم لأنفسهم فيبتعدون وينأون عن القرآن وعن اتباع الرسول، لا بل يدعون الناس إلى غيهم، وفي ذلك تقول الآية الكريمة :"وَهُمْ يَنْهَوْنَ عَنْهُ وَيَنْأَوْنَ عَنْهُ وَإِنْ يُهْلِكُونَ إِلا أَنفُسَهُمْ وَمَا يَشْعُرُونَ(26) - الأنعام"، سبحان الله فهم ينهون الناس عن القرآن الكريم وعن اتباع الإسلام، وينأون بأنفسهم عن الصراط المستقيم، وهم بذلك يهلكون أنفسهم دون أن يشعروا بذلك لأنهم مأخوذون بسكرة عقولهم التي تصور لهم أنهم نوابغ وأن غيرهم من المؤمنين على ضلال وإن يتبعون إلا أساطير الأولين.

تنقلنا الآية الكريمة من مشهد الكفار وهم أعزة في غمرة زهوهم بأفكارهم ودعوتهم الناس إليها، إلى مشهدهم وهم أذلة في حضرة النار يتمنون لو يردون إلى الدنيا فيسلكون مسلك المؤمنين ثم لا يكذبون بآيات الله "وَلَوْ تَرَى إِذْ وُقِفُوا عَلَى النَّارِ فَقَالُوا يَا لَيْتَنَا نُرَدُّ وَلا نُكَذِّبَ بِآيات رَبِّنَا وَنَكُونَ مِنَ الْمُؤْمِنِينَ(27)بَلْ بَدَا لَهُمْ مَا كَانُوا يُخْفُونَ مِنْ قَبْلُ وَلَوْ رُدُّوا لَعَادُوا لِمَا نُهُوا عَنْهُ وَإِنَّهُمْ لَكَاذِبُونَ(28) - الأنعام" وتأمل كيف ختم الله عز وجل الآية الكريمة بحقيقة صادعة يرددها ملحدو زماننا "وَقَالُوا إِنْ هِيَ إِلا حَيَاتُنَا الدُّنْيَا وَمَا نَحْنُ بِمَبْعُوثِينَ(29) - الأنعام" أليست هي نفس العبارة التي يعيدونها باستمرار، سبحان الله "تِلْكَ آيَاتُ اللَّهِ نَتْلُوهَا عَلَيْكَ بِالْحَقِّ فَبِأَيِّ حَدِيثٍ بَعْدَ اللَّهِ وَآيَاتِهِ يُؤْمِنُونَ (6) - الجاثية" !

وتأتي الآيات بعد ذلك لتبين حال الكفار الذين كذبوا بلقاء الله وظنوا أن الحياة ستطوى بعد الموت ثم لا يبعثون، لتتبعها ذكرى كريمة للمتقين مفادها أن الحياة الدنيا ليست سوى لهو ولعب، وأن الدار الآخرة خير لهم، لتختم الآية الكريمة بسؤال استنكاري: "وَمَا الْحَيَاةُ الدُّنْيَا إِلا لَعِبٌ وَلَهْوٌ وَلَلدَّارُ الآخِرَةُ خَيْرٌ لِلَّذِينَ يَتَّقُونَ أَفَلا تَعْقِلُونَ(32) - الأنعام"، وكأني بهذه الآية الكريمة تقول للكافرين الذين أعجبوا بما توهموه حقا: هذا هو ميزان الخيرية وفيه فلتستخدموا عقولكم، أي الدارين أحق بالحياة؟

قهوة
الرابط
فضاء للاستراحة

ماهي المراحل التي ينبغي أن يسلكها المبرمج لإنجاز برنامج ناجح؟

1. مرحلة فهم المشكل وتكوين حلول شاملة:
ينبغي على المبرمج أن يفهم الأمر المطلوب جيدا، وأن يحلله تحليلا دقيقا، ثم يسعى إلى إيجاد الحل الملائم له، من خلال وضع تصور منطقي للخطوات الواجب اتباعها عمليا للوصول إلى النتيجة المتوخاة، هذه الخطوات التي يصطلح عليها في عرف المبرمجين ب **الخوارزميات**.

2. مرحلة التكويد (كتابة الكود):
تُعنى هذه المرحلة بإخضاع الخطوات المنطقية التي تم تصورها في المرحلة الأولى إلى الواقع العملي، من خلال تحويل الخوارزميات الناتجة إلى برنامج تطبيقي بواسطة إحدى لغات البرمجة المناسبة.

3. مرحلة التجريب والمتابعة
في هذه المرحلة يتم قطف ثمار البرنامج من خلال استخدامه وإخضاعه للتجربة، وإصلاح ما به من أخطاء، وتعديل مواصفاته لتتوافق مع احتياجات العميل.

ماهي سمات وعلامات المبرمج الناجح؟

1. الخلفية المعرفية:
لا يكون المبرمج مبرمجا ناجحا، إلا إذا كان على دراية شاملة بكل ما يتعلق بحقله البرمجي، فإن كان يشتغل على تطبيقات سطح المكتب بلغة الجافا مثلا، فعليه أن يستوعب أن نجاحه في هذا المجال مرتبط ارتباطا وثيقا بمدى اطلاعه على كافة آليات وتقنيات إنشاء هذا الضرب من التطبيقات.

2. الضبط البرمجي:
المقصود بالضبط البرمجي، قدرة المبرمج على التوفيق بين خلفيته المعرفية **(الجانب النظري)** وبين إنزالها على واقعه البرمجي **(الجانب التطبيقي)** بشكل سليم يخول له استغلال معلوماته البرمجية على الوجه العملي المطلوب **(الجانب البيداغوجي)**

3. العطاء:
المبرمج الناجح يستفزه أن يرى الناس يبحثون عن معلومة ما فلا يجدونها، لذلك يكون من سماته وأوصافه أنه جزيل العطاء وفير البذل، لا يحتكر ولا يكتم ما وفقه الله إلى تعلمه واكتسابه إيمانا منه أنه لو لجأ من سبقه إلى كتم المعلومة لما وصلت إليه.

الأسد الحكيم و Try...Catch

في إحدى حقول السافانا بإفريقيا كان يقطن أسد متعصب للغة سي شارب، وفي إحدى الأيام جمع رعيته وقال لهم:
- أتدرون أن شركة سان Sun Microsystems سرقت فكرة سي شارب من ميكروسوفت وأنشأت على غرارها لغة جافا.
كتم الحضور ضحكاتهم بعد أن أدركوا حجم كذبة الأسد، فتقدم إلى المنصة سنور شجاع وخاطب الأسد:
- ولكن يا مولاي تاريخ إصدار لغة جافا أقدم من تاريخ لغة سي شارب ؟
أصيب الأسد بإحراج شديد، وحاول الخروج من اللقاء بأقل الفضائح فهتف به:
- نعم أعلم ذلك يا قليل الأدب لكنني تعمدت الخطأ لأصطاد عديمي التربية أمثالك، وأعرف الخائنين من رعيتي.
ثم التفت إلى الجموع وقال لهم مكشرا:
- أتدرون ماذا يسمى هذا في عرف المبرمجين؟
قالوا بصوت واحد:
- لا ندري.
ابتسم الأسد في كبرياء وقال:
- يسمى معالجة الاستثناءات Exceptions Handling، لقد أوقعت السنور في الكاتش بلوك.

طموح الثعالب وصدمة HTML

زعموا أن ثعلبا أراد أن يتعمق في البرمجة، فلما سأل عشيرته وأخلاءه أنبؤوه أن للذئب باعا طويلا في علوم الحاسب، فحمل الثعلب كراسته ويمم شطر كهف الذئب.
ولما وصل إلى باب الكهف، استأذن الذئب فأذن له في الدخول وقال له:
- ما الذي أتى بك يا ماكر؟
رد الثعلب في استحياء:
- سمعت أنك من أهل البرمجة فأتيتك لتعلمني.
فقال له الذئب باستكبار واستعلاء:
- إنني مبرمج محترف وأتقن أصعب لغة برمجية في العالم: لغة HTML
ضحك الثعلب بقوة ثم هتف بالذئب:
- يا أحمق HTML ليست لغة برمجية ! ويح سعدي ومن قصدت !

مَعَادِنُ الْوَرَى

قصيدة لخالد السعداني

أَ جِئْتَنِي الآنَ وَقَدْ تَعِبْتُ يا وَطَنِي؟ ... لَسْتُ أَرْضَاكَ حَبِيبًا وَقَدْ مَضَى زَمَنِي

مَضَيْتُ عَلَى شَوْكِ وِهَادِكَ سَاعِيًا ... لِمجدك فَأَمْضَيْتَ سَيْفَكَ فِيَّ لِوَهَنِي

وَكَمْ من مَوَاجِعَ ضَاقَ الْفُؤَادُ بِهَا ... لَكَ فِيهَا أَسْبَابٌ وَلِي فِيهَا شَجَنِي

وَإِذَا عَزَّ أَهْلُ الفَسَادِ فِي بَلَدَةٍ ... دَقَّتْ لِلصَّالِحِينَ بِهَا سَاعَةُ الظَّعَنِ

مَعَالِمُكَ الَّتِي أَقَامَهَا الصَّلاَحُ هَوَتْ ... وَأَرْدَاهَا الْفَسَادُ رَسْمًا عَلَى دِمَنِ

تَكَلَّمَ السَّفِيهُ فِيكَ بِنَقصٍ فَعَذَلْتُهُ ... فَأَثْقَلَتْنِي ذُلاًّ حِينَ أَنْتَ أَخْرَسْتَنِي

وَهَلْ يَهونُ الْوِدَادُ بَيْنَ أَصْحَابِه ... فَقَدْ أَهَنْتَهُ وَغَادَرَ الْكَرَى جَفْنِي

مَعَادِنُ الْوَرَى سَاعَةَ الرَّخَاءِ ذَهَبٌ ... يَنْكَشِفُ زُورُهَا عِنْدَ سَاعَةِ الْفِتَنِ

وَكَمْ مِنْ بَادٍ يَغْشَى الْحُسْنُ قَوْلَه ... فَإِنْ شَدَّ الْحَزْمُ قَلَبَ ظَهْرَ الْمِجَنِ

للهِ دَرُّ الْخُطُوبِ مَا أَفْصَحَ قَوْلَهَا ... كُلَّمَا نَطَقَتْ فَضَحَتْ أُولِي الدَّخَنِ

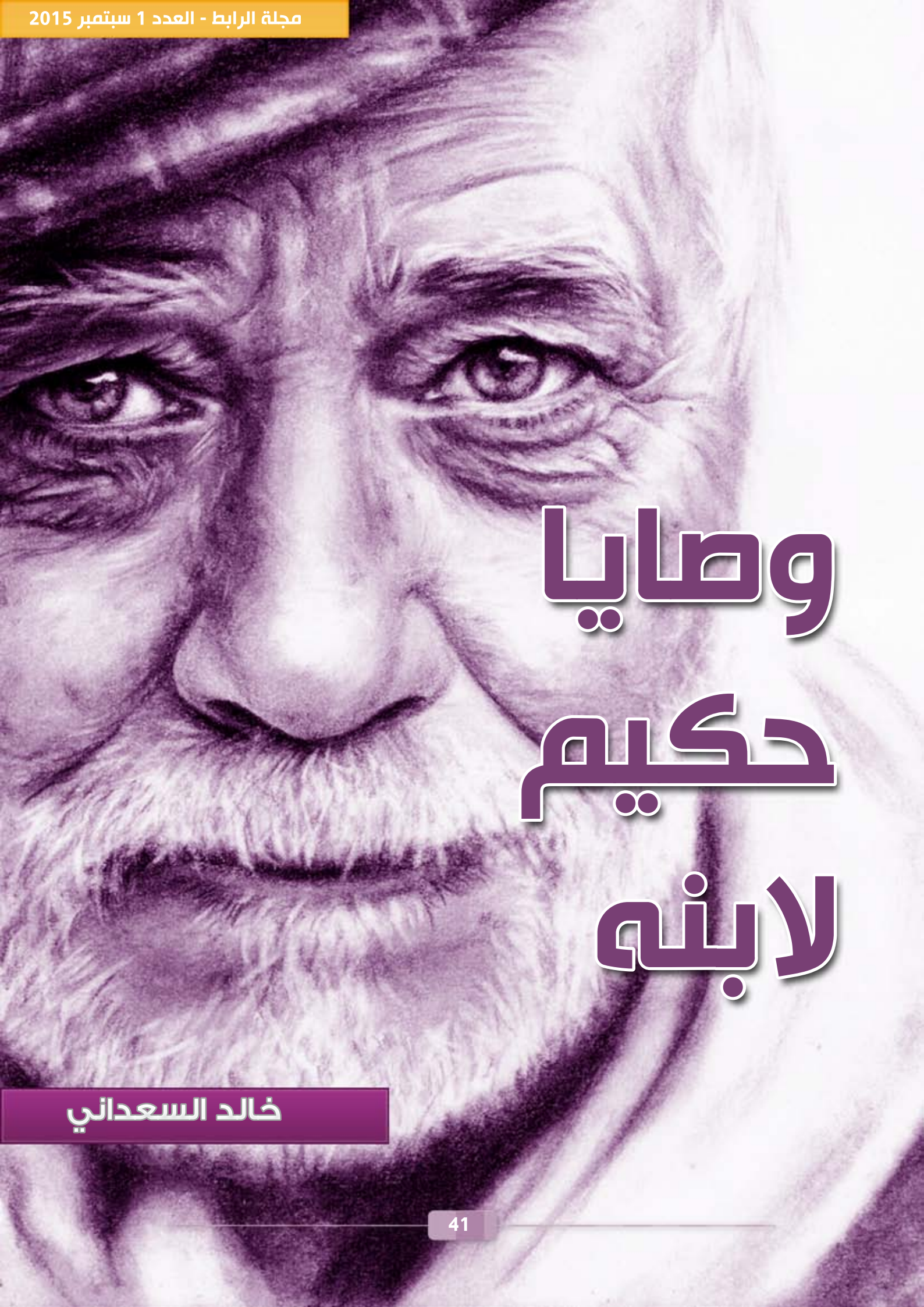
وَلِلْفَتَى طِبَاعُ سُوءٍ لاَ جَلاَءَ لَهَا ... إِلاَّ إِذَا غَسَلَتْهَا يَدُ الدَّهْرِ بِالْمِحَنِ

وَكَذَا نَفْسُ الْعَزِيزِ لاَ رَاحَةَ لَهَا ... فِي عَيْشٍ يَرَى الذُّلَّ رَأْيَ الْوَثَنِ

فَلاَ هَنَاءَ لِنفُوسِ الْكِرَامِ إِلاَّ إِذَا ... كَسَرَتْ قُيُودَ الأَسْرِ وَالسَّجَنِ

رَاحَةُ الأَحْرَارِ فِي قَلْبٍ مُتَدَبِّرٍ ... وَرَاحَةُ الْعَبِيدِ نَوْمٌ وَرَخَا بَدَنِ

إِنَّ الْوَطَنَ أُذُنٌ تَصْغَى فَإِنْ أَبَى ... سَمَاعَكَ اسْتَحَالَ إِلَى كُوخِ سَكَنِ



وصايا حكيم لابنه

خالد السعداني

وصايا حكيم لابنه

#أساطير_برمجية

يا بني، حينما أكلمك اصغ إلي جيدا وحبلك على غاربك...

فأنا لا أطالبك بالتنفيذ Execution لأننا فقدنا الأمل فيه...

فحتى الأكواد التي نسطرها بأناملنا تخذلنا ساعة التنفيذ Runtime.

يا بني، لا تتعلق بأحد مهما استحسنته فإن الناس نُسَخٌ Versions...

وقد تعجبك نسختهم الآنية فترتبط بها....

ثم يصدمونك عند أول تحديث Update أو إصدار جديد.

يا بني، لقد أدركت أكوادا لا أفهم عاليها من سافلها...

لكنني تحاملت على نفسي وسايرتها باجتهاد حتى صرت أراها أبهى من ضفائر بثينة...

وأجمل من لوحات بيكاسو...

فلا تحكم على أحد من أول وهلة قبل أن تعايشه وتعاشره.

يا بني، سيحيط بك الفاشلون والمُثَبِّطُون وسيقولون لك: لن تستطيع إنجاز شيء...

فاضرب بكلامهم عرض الحائط فقد قالوا نفس الكلام عن لغة الأسمبلي Assembly...

لكن تأمل الآن كم من اللغات الرائعة التي خرجت من رحمها.

يا بني، عِشْ على أمل، فالدنيا حلقة تكرارية من نوع while مرتكزة على شرط: ما دمت متفائلا ستحيى....

فإن أنت يئست كنت أقرب لباطن الأرض منك إلى ظاهرها.

يا بني، شَرُّ القومِ قَوْمٌ مَرُّوا بِأَرْضٍ فَلاَةٍ...

فلما تمكن منهم الظمأ واستيأسوا من طلب الورد...

أدركوا بئرا فشربوا منها حتى قضوا وطرهم...

و وردوا إبلهم وملؤوا قرابهم...

ثم طمروا البئر وواصلوا سفرهم لئلا ينتفع منها غيرهم...

وما علموا أن لو سبقهم إليها أمثالهم لكانوا قد هلكوا عطشا...

فكذلك مثل أهل الأكواد المحتكرين...

الذين كلما عثروا على سورس كود استحوذوا عليه وَضَنُّوا به عن غيرهم مخافة أن ينتفعوا منه أو يسابقوهم فيه...

فبئس القوم هم...

جانبهم وباعد بينك وبينهم...

فإنهم أناس لا يصطلى بنارهم ولايرتجى نفعهم.

الخاتمة

جيل من النوابغ

الحمد لله ابتداء وانتهاء، وبعد:

من جميل ما نسب إلى نابليون بونابرت قوله: "أحسن وسيلة للتغلب على الصعاب اختراقها"، وقريب منه قول القائل: "الطريق المهمل مليئة بالأشواك"، وأجمل منهما قول المتنبي:

على قدر أهل العزم تأتي العزائم وتأتي على قدر الكرام المكارم
وتعظم في عين الصغير صغارها وتصغر في عين العظيم العظائم

دائما ما يتملكنا الخوف عندما نقف على أعتاب طموحاتنا فنجدها محفوفة بالمكاره والغمرات..
آن لنا أن نخوض غمار ما نحب، أما الصعوبة والتعقيد فهي أمور ثانوية ستسهل مع مرور الزمن.
لننظر إلى الأمور الصعبة على أنها نعمة ينبغي أن نظفر بها، لا على أنها أثقال تضيق صدورنا بها.
إن كنت تخشى لغة جافا فاعلم أنك بحيازتها ستبرمج تطبيقات ومواقع وألعاب وغيرها..
إن كنت تخشى شيئا فاعلم أن غيرك بارعٌ فيه، واسأل نفسك بهدوء: مالذي أُعْطِيَهُ هذا الغير وحرمته أنت؟ لا تتشبث بالإمكانيات المادية فإنها تُقْضَى ولو بالترقيع، ولكن تشبث بالإمكانيات الذهنية التي من الوارد جدا أن يكون نصيبك منها أكبر وأجل من نصيب ذاك الغير الذي حاز ما هِبْتَ أنت خَوْضَهُ.
سيبقى هدفنا في "الرابط" بعون الله هو السعي نحو التألق والتميز لصناعة جيل من النوابغ لا يرضى دون القمة بديلا.

دام لكم البشر والفرح

الحقيبة البرمجية لاحتراف لغة الفيجوال بيسك 2015 من الصفر

الجزء الأول

باقة تعليمية من الدروس القيمة لتعلم إنشاء البرامج من الصفر

خالد السعداني

مبرمج ومدرس

محتوى الحقيبة:

- باقة من الفيديوهات التعليمية تمتد إلى أكثر من 180 فيديو عالية الجودة بأسلوب علمي يجمع بين الشرح النظري والممارسات التطبيقية.
- كتاب الإكليل (نسخة الكترونية PDF):

الجزء 1: الشامل في أساسيات البرمجة بلغة الفيجوال بيسك دوت نيت

الجزء 2: المختصر المفيد في البرمجة الكائنية التوجه

معلومات الاتصال:

www.alhaqiba.com
mobarmijoun@gmail.com
www.facebook.com/EssadaniPage
www.youtube.com/EssaadaniTV
@K_essaadani
+212 0673 075 105

تنبيه

- الفيديوهات بصيغة HD وسيتم تزويد الطالب بروابط تحميل مباشرة
- الكتب بصيغة PDF وسيتم تسليمها للطالب مع الفيديوهات
- هذا العمل متعوب عليه وأخذ من صاحبه وقتا وجهدا فأي تصرف في المادة دون مشورة صاحبها هو ظلم وتجاوز.