

4. أنواع البرمجيات

البرمجيات تحيط بنا في كل مكان : في المناطق الصناعية ، وفي الاستعمالات الخاصة ، وفي أنظمة الاتصالات ، وفي أنظمة النقل ، وفي مجالات أخرى، فالبرامج تأتي مختلفة في أشكالها وأحجامها، فمنها ما يدمج في الهاتف المحمولة ، ومنها ما تقوم بتصميم مركبات الفضاء. أما عن تصنیف البرمجيات ، نستطيع أن نميز نوعان رئيسیان:

- **برمجيات النظام (System Software)** : هي البرامج التي تمثل الأدوات التي تساعده في بناء أو دعم البرمجيات التطبيقية. وتنتمي برمجيات الأنظمة بالتفاعل الكثيف مع عتاد الحاسب الآلي وباستعمال كثيف والمشاركة في الموارد من قبل المستخدمين ، مثل : أنظمة التشغيل ، ومتراجم اللغات ، وخلافه.

- **البرمجيات التطبيقية (Application Software)** : هي البرامج التي تساعده في تأدية بعض المهام المفيدة أو الممتعة بشكل مباشر وأمثلة ذلك : الألعاب ، برامج الصرافة الآلية (ATMs) ، برامج التحكم في الطائرة وبرامج البريد الإلكتروني وبرامج معالجات النصوص ، وبرامج الجدولة.

وضمن تصنیف البرمجيات التطبيقية ، من المفيد أن نميز أصناف البرمجيات التالية:

- **برمجيات الألعاب (Games Software)**.

- **برمجيات نظم المعلومات (Information Systems Software)** : وهي من أهم وأوسع مجالات التطبيقات البرمجية ، حيث تقوم هذه النظم بالاتصال بقاعدة بيانات كبيرة أو أكثر تحتوي على كميات ضخمة من البيانات والمعلومات الخاصة بهذه النظم وبطريقة تسهل الحصول على المعلومات واتخاذ القرارات ، والتفاعل مع المستخدم لإتمام عملياته بيسر وموثوقية كبيرة ، ومن هذه الأنظمة على سبيل المثال : نظام حجز مقعد بالخطوط الجوية ، نظام المعاملات البنكية ونظام شئون الموظفين والرواتب ، خلافه من الأنظمة.

- **برمجيات نظم الوقت الحقيقي (Real – Time Systems Software)** : هي الأنظمة

التي يجب أن تعطي إجابة ضمن شروط زمنية محددة ، وتألف مكونات برمجيات الزمن الحقيقي من : م Jasat لجمع وتصفيه البيانات من المحيط الخارجي ، ومحول يقوم بتحويل البيانات إلى الشكل الذي يقبله النظام ، بالإضافة إلى مكون تحكم وإخراج إلى المحيط الخارجي ، ومراقب تحكم ينسق بين جميع هذه المكونات لتستمر المحافظة على الاستجابة بالزمن الحقيقي. ومثال على ذلك برمجيات التحكم في مراكز الطاقة ، والأقمار الصناعية ، والعمليات الصناعية المختلفة.

- **البرمجيات المدمجة بالأجهزة (Embedded Software)** : تقوم هذه البرمجيات بأداء وظائف خاصة ضمن الأجهزة الذكية من خلال برمجة ذاكرات القراءة فقط بهذه البرمجيات وتركيبها ضمن هذه الأجهزة. مثال على هذه الوظائف : التحكم في سرعة محرك الغسالة الكهربائية ، وكذلك التحكم في اختيار برامج تشغيلها.
- **البرمجيات المكتبية (Office Software)** : وهي برمجيات موجهة لأجهزة الحاسوب الشخصية (PC Computer) لأنماط الأعمال المكتبية مثل : مجموعة برمجيات ميكروسوف特 أوفس كبرنامج معالجة الكلمات وتنسيق النصوص ، وبرنامج الجداول الحسابية ، وتقديم العروض التقديمية ، وخلافه . وكذلك برمجيات الإنترنت والبريد الإلكتروني ، وخلافه من هذه البرمجيات.
- **البرمجيات العلمية والهندسية (Engineering and Scientific Software)** : تتميز هذه الأنظمة بمحاكاة الأنظمة وتعتمد على الخوارزميات العددية ، وتتراوح تطبيقاتها من إيجاد جذور معادلة من الدرجة الثانية إلى تحليل إحداثيات مركبات القضاء وإجراء التفاعلات النووية .
- **برمجيات الذكاء الاصطناعي (Artificial Intelligence Software)** : تعتمد برمجيات الذكاء الاصطناعي على مجموعة من الحقائق (Facts) والعلاقات (Relationships) . وتقوم هذه البرمجيات بتحديد الحل المناسب من خلال الاستنتاج والاستدلال المنطقي (Logical Deductions) وبكفاءة عالية ، حيث يتم تمثيل العلاقات بين الأشياء وتجميعها وتنظيمها للوصول إلى استنتاج منطقي للحقائق التي تمثلها تلك

العلاقات. ومثال ذلك برمجيات التعرف على الأشكال (الصور والصوت).

• **البرمجيات المغلفة (Shrink-wrap Software)** :

يطلق هذا المصطلح (البرمجيات المغلفة) على البرمجيات التطبيقية المعدة للتوزيع التجاري (Commercial Software Packages) (مثل : ميكروسوفت أوفس). ويتم تحويل البرمجيات التطبيقية المعدة للاستخدام الداخلي (In-House Application Software) إلى برمجيات تطبيقية معدة للتوزيع التجاري (برمجيات مغلفة) بإضافة العديد من الوحدات البرمجية الجديدة الصغيرة إليها مثل برمجية حماية الشيفرة ، وبرمجة التثبيت وبرمجة الرخص ، وخلافه ، وبعد ذلك يتم وضعها كحزمة برمجية تطبيقية تجارية قابلة للتنشيط على أقراص مدمجة مغلفة تُعد للبيع التجاري. ومن أهم الخصائص أو الاعتبارات الهامة التي يجب أن تؤخذ في الاعتبار عند تطوير هذه البرمجيات ما يلي :

① **سهولة التكيف مع منصات تشغيل الحاسب الآلي (Multi-Platform Resilience)** :

لا بد أن يراعى في عملية التطوير للحزمة بأن تكون صالحة للعمل على منصات التشغيل المختلفة للحاسوب (Platform Independent) ، وإذا كانت غير موافقة لمنصات معينة ، يجب العمل على تكييفها للعمل معها عن طريق إضافة وحدات برمجية تقوم بهذا الغرض.

② **الخيارات الإقليمية وخيارات اللغة (Localization)** :

يجب أن تتوافق هذه البرمجيات مع منصات التشغيل ذات اللغات المتعددة (Multi-Languages Platforms) عن طريق إضافة وحدة برمجية Languages Service يتم من خلالها اختيار الخيارات الإقليمية وخيارات اللغة بسهولة تامة مما يساعد على انتشار البرمجية وبالتالي على تسويقها.

③ **إدارة عملية الترقع (Patch Management)** :

وهي العملية الخاصة بتنسيق عمليات التحديثات اللازمة للإصدارات ، وكيفية توزيع الرقيعات (Patches) إلى العملاء لسد الثغرات التي تنشأ في الحزمة أثناء عمليات التشغيل (مثل الرقيعات الأمنية التي توزيعها شركة ميكروسوفت لسد الثغرات الأمنية التي

تكتشف بعد توزيع نظام التشغيل) ، وكيفية حماية العملاء من أي فقد للبيانات أو بيانات التشكيل ، وكيفية تحديث الحزمة.

④ رخصة الاستخدام وعملية التنشيط المنتج (Activation License Control and Product) : عملية تنشيط المنتج البرمجي هي عبارة عن إجرائية للتحقق من صلاحية رخصة استخدامه (**License Activation Procedure**) لمنع استخدام المنتج لغير المرخص لهم ، لذا يجب تضمين هذه العملية في البرمجيات التجارية لحفظ على حق الملكية ، ويجب أن تتضمن أيضاً طريق سهلة لعملية التنشيط من قبل العملاء ، ويوجد العديد من أنواع رخص التشغيل وهي رخص محددة بوقت معين وتم لنسخ المنتج التقويمية (**Time Limited Evaluation Edition**) (مثل البرمجيات المشتركة التي توزع من خلال الإنترنت للاستخدام لمدة شهر) ، أو رخص استخدام غير محددة الوقت (مثل رخص التشغيل الخاصة بحزمة ميكروسوف特 أوفس).

⑤ حماية البرنامج المصدر (Source Code Protection) : لا بد من حماية برنامج المصدر من العابثين الذين يحاولون الحصول عليه ، وذلك باللجوء إلى وسائل التشفير والحماية المختلفة.

⑥ وجود وسائل المساعدة (Help) : لا بد من احتواء الحزمة على طرق لمساعدة العميل في استخدام البرنامج بمعظم اللغات العالمية المشهورة ، وكذلك إرشادات في حالة حدوث مشاكل (**Troubleshooting**) ، ويجب أيضاً تخصيص موقع على شبكة الإنترنت للمنتج لتقديم المساعدة للعملاء.

⑦ التثبيت (Installation) : وفقاً للأبحاث الحديثة الخاصة بتنشيط البرمجيات وجد أن حوالي 30% من فشل حزم البرمجيات ينشأ أساساً من عدم التثبيت الصحيح ، لذا لا بد من التفكير جدياً في طريقة قوية لعملية التثبيت لتجنب الأخطاء التي تؤدي إلى فشل التثبيت ، ولا بد من الأخذ في الاعتبار أنواع الحاسوبات المختلفة ، والتفكير في العمليات غير العادية في عملية التثبيت كوضع مفاتيح للتسجيل والرخص ، وعملية التنشيط.

أسئلة تقويم ذاتي



1. ما هي البرمجيات المغلفة؟ وكيف يمكنك تحويل البرمجيات التطبيقية المطورة للاستخدام الداخلي لتصبح برمجيات مغلفة.

2. الجدول التالي يحتوي العمود الأول منه على الأنماط المختلفة للبرمجيات والعمود الثاني منه على تعاريفات لهذه الأنماط وهي غير مرتبة، أعد بناء الجدول بحيث يكون هناك تواافق بين الأنماط والتعريفات.

تعريفها	أنماط البرمجيات
• برمجيات موجهة لأتمتة الأعمال المكتبية.	• برمجيات النظام
• برمجيات لمحاكاة الأنظمة وتعتمد على الخوارزميات العددية.	• برمجيات تطبيقية
• برمجيات تعتمد على مجموعة من الحقائق وال العلاقات حيث يتم تحديد الحل المناسب من خلال الاستنتاج والاستدلال المنطقي.	• برمجيات الألعاب
• برمجيات المدمرة.	• برمجيات نظم الوقت الحقيقي
• برمجيات تعطي إجابة ضمن شروط زمنية محددة.	• البرمجيات المدمجة
• تقوم هذه البرمجيات بالاتصال بقواعد البيانات بطريقة تسهل الحصول على المعلومات واتخاذ القرارات.	• برمجيات بالأجهزة
• برمجيات تقوم بأداء وظائف خاصة ضمن الأجهزة.	• البرمجيات المكتبية
• برمجيات للتسليم وزيادة الإدراك والمعرفة.	• البرمجيات العلمية والهندسية
• برمجيات تساعد في بناء أو دعم البرامج التطبيقية.	• برمجيات الذكاء الاصطناعي

5. أدوات هندسة البرمجيات بمساعدة الحاسوب

(CASE Tools)

CASE : هي كلمة مكونة من أولى حروف عبارة Computer Aided Software Engineering (هندسة البرمجيات بمساعدة الحاسوب) وهي تشمل طائفة واسعة من الأنواع المختلفة من البرامج التي تُستخدم في مساندة الأنشطة الخاصة بالعمليات البرمجية مثل تحليل المتطلبات وإعداد النماذج التحليلية وتصحيح الأخطاء والاختبار ، وتتوافق جميع الطرق في أيامنا هذه مع تقنية CASE ومنها أجهزة التمثيل بالرموز وفي وحدات التحليل المستخدمة في تدقيق نماذج النظام فوق قواعد توثيقية عملية لإنشاء النظام. وقد تشمل أدوات CASE كذلك على معدات الترميز (النوكود) والتي تقوم بصورة آلية بإعداد تكويد للبرامج من واقع نموذج النظام إضافة إلى إعداد الخطوط الإرشادية الخاصة بتنفيذ العملية وتقديم النصائح لمهندسي البرامج حول الخطوة التالية التي يجب القيام بها.

وهذا النوع من أدوات CASE والذي يهدف إلى إسناد ودعم عملية التحليل والتصميم يسمى في بعض الأحيان بأداة CASE العليا ، لأنه يقدم المساعدة للمراحل المبكرة من عملية البرمجة وخلافاً لذلك فإن أدوات CASE المعدة لتقديم المساعدة والدعم لعملية التنفيذ والاختيار كأنظمة تصحيح الأخطاء وتحليل البرامج وإعداد متطلبات الاختيارات وأدوات تحرير البرامج تسمى أحياناً بأدوات CASE السفلية .

ويمكن القول بأن أدوات هندسة البرمجيات بالاستعانة بالحاسوب الآلي هي : عبارة عن برامج مصممة لمساعدة المبرمجين على التغلب على تعقيدات العمليات البرمجية المختلفة والمساعدة على أتمتها ، وهي تمثل مجموعة من الضوابط والآليات التي تساعد في تحليل ، وتصميم ، ونمذجة وترميز ، واختبار ، وتوثيق البرمجيات. ويُعد برنامج محرر النصوص كمثال وأداة قيمة من أدوات تطوير البرمجيات لتنظيم الملف أو البرنامج ، وكذلك برنامج محرر الرسوم كمثال آخر لتوثيق الرسوم الخاصة بالتصميم للنظم الكبرى من البرمجيات ، والمثير للجدل أن هذه الوسائل المساعدة لم يتم اكتشافها وتقديرها بصورة كاملة ، فكلما ظهرت احتياجات برمجية جديدة أخرى أصبح من الضروري اكتشاف وسائل

وأدوات جديدة مناسبة لها.

1.5 ميزات أدوات هندسة البرمجيات بمساعدة الحاسوب

(CASE Tools Advantages)

يمكن تلخيص أهم ميزات أدوات هندسة البرمجيات كالتالي :

① **السرعة الفائقة (Increased Speed)** : تتميز أدوات هندسة البرمجيات بالسرعة الفائقة في أتمتة العمليات وتخفيض الوقت اللازم لاستكمال العديد من المهام خاصة التي تتطلب المخططات الرسومية والبنود الخاصة بها. وتقدر التحسينات في الإنتاجية بعد التطبيق في مدى يتراوح من (200 - 300%).

② **الدقة المتزايدة (Increased Accuracy)** : تساعد أدوات هندسة البرمجيات في اكتشاف الأخطاء والذي تُعد خطوة مهمة في إزالة هذه الأخطاء حيث يوفر اكتشاف الأخطاء وإزالتها الوقت والجهد في المراحل المبكرة من إعداد النظام ، و كلما زاد حجم النظام أصبح من الصعب اكتشاف الأخطاء مما يؤدي إلى زيادة الوقت والجهد المطلوب للتنسيق والإدارة بين فرق العمل المختلفة.

③ **تخفيض الوقت اللازم للصيانة (Reduced Lifetime Maintenance)** : وكنتيجة للتحليل الجيد، التصميم الجيد، توليد الشفرات آلياً، اختبار البرمجيات آلياً، التوثيق الآلي، يتحسن أداء النظام وبالتالي فإن الجهد اللازم للصيانة ينخفض إلى حد كبير. وكذلك يمكن توفير العديد من المنابع لتطوير أنظمة جديدة ، وتقوم أدوات معايدة البرمجيات (برمجيات إعادة الهندسة Re-engineering Programs) باكتشاف الأجزاء من البرمجيات والتي يمكن إعادة استخدامها مما يزيد من كفاءة العمل وتقليل الجهد المطلوب.

④ **الوثيق الأفضل (Better Documentation)** : وباستخدام أدوات هندسة البرمجيات المساعدة هناك العديد من كميات التوثيق التي يتم توليدها لتمثل ملاحظات على كيفية تطوير النظام وعمل صيانة له.

⑤ البرمجة في أيدي غير المبرمجين :

(Programming in the Hands of Non-programmers) :

مع التطور السريع والاتجاه نحو تكنولوجيا البرمجة الشبيهة ، وقواعد بيانات خادم العميل يمكن أن تتم عملية البرمجة بآناس ليس لديهم خلفية كاملة عن البرمجة، حيث يصبح المهم هو فهم الهدف الأساسي من البرنامج والقدرة على تحليل مكونات البرنامج وتفاصيله والتي تستخدم في توليد هذه الأدوات (أدوات التطوير منخفضة المستوى).

⑥ الفوائد الغير ملموسة (Intangible Benefits) : تفيد أدوات التطوير في مشاركة

المستخدم والتي تساهم في قيوله الجيد للنظام الجديد وهذا يساعده إلى حد كبير في تخفيف منحنى التعليم الأولي.

أسئلة تقويم ذاتي

ما هي أدوات هندسة البرمجيات بمساعدة الحاسوب الآلي ؟ وما هي أهم مميزاتها وقصور استخدامها؟.



2.5 قصور أدوات هندسة البرمجيات بمساعدة الحاسوب

(CASE Tools Limitations)

يمكن تلخيص أهم القصور الموجودة في أدوات هندسة البرمجيات بمساعدة الحاسوب في التالي :

① مزج الأدوات (Tool Mix) : من المهم جدا اختيار ملائم لمزيج من الأدوات للحصول على مميزات وتكاليف مخفضة. ولابد من اختيار الأدوات غير المعتمدة على نوعية العتاد وإمكانية مشاركة نتائج أداة تطوير تم استخدامها مع أدوات أخرى لضمان تكاملية الأدوات مع بعضها.

② التكاليف (Cost) : الأدوات المساعدة ليست رخيصة الثمن ، وفي الحقيقة فإن شركات التطوير على نطاق ضيق لا تستخدم هذه الوسائل معتقدين بأنها مفيدة فقط في

تطوير الأنظمة الكبيرة ، حيث أن تكالفة تزويد مطوري الأنظمة بهذه الوسائل مكلفة .
ويعتبر (العتاد ، البرمجيات ، الاستشارات) كلها عوامل تدخل في معادلة التكالفة .

③ **منحنى التعلم (Learning Curve)** : وفي معظم الحالات فإن إنتاجية المبرمج تصل إلى أدنى مرحلة لها في المرحلة الأولية للتطبيق وذلك لاحتياج المستخدمين للوقت الكافي للتعلم ، بالرغم من أن مستشاري هذه الأدوات يقدمون العديد من الخبرات لمستخدميها حيث يقومون بالتدريب وكذلك وجود العديد من مواقع الخدمات الخاصة بهذه الأدوات على شبكة الإنترنت والتي تساعد في تعجيل منحنى التعليم لهذه الأدوات .

3.5 التصنيف العام لأدوات هندسة البرمجيات المساعدة

(CASE Tools General Classifications) :

① أدوات التطوير المساعدة عالية المستوى (Upper Case Tools) :

وهي تمثل أدوات أنشطة العمليات المبكرة للمطلوبات ، التحليل والتصميم . وتساعد المحللون في تخزين ، وتنظيم ، وتحليل نماذج العمل ، وترتيب الأسبقيات التالية :

- إستراتيجيات العمل الحالية والمستقبلية .
- النظم المكملة وإستراتيجيات تطبيقها .
- قواعد البيانات والشبكات المراد تطويرها .
- التطبيقات المراد تطويرها .

② أدوات التطوير المساعدة منخفضة المستوى (Lower Case Tools) :

وهي تمثل أدوات دعم الأنشطة التالية مثل البرمجة واكتشاف العلل والاختبارات حيث تساعد المبرمجين في زيادة الإنتاجية والجودة ، و تمتد هذه الأدوات إلى تفاصيل التصميم للمساعدة في توليد التطبيقات من خلال الخدمات التالية :

- المساعدة في سرعة اختبار البرنامج واكتشاف الأخطاء .
- توليد الكود للبرنامج من مواصفات التحليل والتصميم .
- توليد الشاشات وقواعد البيانات .

② أدوات هندسة البرمجيات بمساعدة الحاسوب لدورة حياة التطوير المتقطعة :

(Cross Life Cycle Case Tools) :

وهي تشمل وسائل إدارة المشاريع التي تساعد المديرين في (الخطيط ، وضع الجداول الزمنية ، وإعداد التقارير ، وتوزيع الموارد).
والجدول رقم (1) يوضح أهم تصنیفات أدوات هندسة البرمجيات بمساعدة الحاسوب وأمثلة عليها.

جدول(1) : تصنیف أدوات هندسة البرمجيات بمساعدة الحاسوب وأمثلة عليها

نوع الأداة (Tool Type)	أمثلة على الأداة (Examples)
أدوات التخطيط (Planning Tools)	• أدوات بيرت (PERT Tools)
	• أدوات التقييم (Estimation Tools)
	• الجداول الإلكترونية (Spreadsheets)
أدوات التحرير (Editing Tools)	• محررات النصوص (Text Editors)
	• معالجات الكلمات (Word Processors)
	• محررات المخططات (Diagram Editors)
أدوات إدارة التغيير (Change Management Tools)	• أدوات تتبع المتطلبات (Requirements Traceability Tools)
	• نظم تحكم التغيير (Change Control Systems)
	• نظم إدارة الإصدار (Version Management Systems)
أدوات إدارة التكوين (Configuration Management Tools)	• أدوات بناء النظم (System Building Tools)
	• اللغات عالية المستوى (Very High-level Languages)
	• مولدات واجهات المستخدم (User Interface Generators)
أدوات دعم الطرق (Method Support)	• محررات التصميم (Design Editors)
	• قواميس البيانات (Data Dictionaries)

أمثلة على الأداة (Examples)	نوع الأداة (Tool Type)
• مولدات الشفرة (Code Generators)	Tools
• المترجمات (Compilers) • المفسرات (Interpreters)	أدوات معالجة اللغات (Language Processing Tools)
• مولدات التداخل (Cross reference Generators) • المحللات الإستاتيكية (Static Analyzers) • المحللات الديناميكية (Dynamic Analyzers)	أدوات تحليل البرامج (Program Analysis Tools)
• مولدات اختبار البيانات (Test Data Generators) • مقارنات الملفات (File Comparators)	أدوات الاختبار (Testing Tools)
• نظم التصحيح التفاعلية (Interactive Debugging Systems)	أدوات اكتشاف وتصحيح الأخطاء (Debugging Tools)
• برامج شكل الصفحة (Page Layout Programs) • محررات الصور (Image Editors)	أدوات التوثيق (Documentation Tools)
• نظم التداخل (Cross-reference Systems) • نظم برماج إعادة الهيكلة (Program Restructuring Systems)	أدوات إعادة الهندسة (Re-engineering Tools)

6. التحديات الرئيسية التي تواجه هندسة البرمجيات

(Key challenges Facing Software Engineering) :

تواجه هندسة البرامج ثلاثة تحديات كبرى في القرن الحادى والعشرين ، وهى :

① التحدي المتعلق بالموروث : (The Legacy Challenge)

لقد تطورت غالبية الأنظمة البرمجية الرئيسية المستخدمة حالياً منذ سنوات عديدة خلت ، ورغم ذلك لا تزال تلك الأنظمة تؤدي وظائف حساسة. إن التحدي المتعلق بالموروث هو التحدي الذي يعني أساساً بصيانة وتحديث البرامج بطريقة يمكن معها تفادي

التكاليف الباهظة والاستمرار في الوقت ذاته في تقديم الخدمات الأساسية في قطاع الأعمال.

② تحديد عدم التجانس (The Heterogeneity Challenge) :

تزداد الحاجة وباستمرار إلى أنظمة تعمل كأنظمة موزعة عبر شبكات تتضمن على أنواع مختلفة من الحاسوبات الآلية ذات الأنظمة المساعدة المختلفة وهذا التحدي يعوق تطوير أساليب إعداد برامج يمكن الاعتماد عليها بدرجة كافية للتعامل مع هذا التحدي.

③ تحدي المخرجات (The Delivery Challenge) :

تُعد العديد من الأساليب المتتبعة في هندسة البرامج مضيعة للوقت. كما أن الزمن الذي تستغرقه يحد وبشكل كبير من فرص تطوير البرامج وضمان جودتها . على الرغم من كل ذلك ، فإنه ينبغي على قطاع الأعمال والتجارة في أيامنا هذه أن يتميز بسرعة الاستجابة والتغيير السريع، كما ينبغي أن تتغير وبنفس الوتيرة برامج المساعدة الخاصة بها. إن تحدي المخرجات هذا يرتبط بالزمن اللازم للحصول على مخرجات من أنظمة كبيرة ومعقدة دون التفريط في الجودة والتوعية. وبالطبع فإن هذه المسألة ليست مسألة منفصلة فقد يلزم مثلاً إجراء تغيير سريع في نظام قديم العهد من أجل جعل النفاد إليه من خلال الشبكة أمراً سهلاً.

والتعامل مع هذه التحديات يلزمنا توفر أدوات وأساليب جديدة واستخدام طرق إبداعية من أجل القيام بالبرمجة واستخدام الطرق الحالية في هندسة البرمجيات جنباً إلى جنب مع تلك الطرق الإبداعية.

7. مهنة البرمجة (Programming Career)

مهنة البرمجة من المهن الهامة والمطلوبة في السوق العربية بصفة خاصة والعالمية بصفة عامة ، ولكن بشرط أن يكون المبرمج على كفاءة عالية وقدرة على استخدام معظم الأدوات البرمجية وتوظيفها بصورة مثالية، ومن يريد العمل بمهنة البرمجة يجب أن يكون مستخدماً متربساً للحاسوب الآلي وله خبرة طويلة في التعامل مع شتى أنواع برمجياته ليس فقط كمستخدم عادي ولكن كشخص قادر على فهم كيفية تصميم وإنشاء هذه البرمجيات ،

مع القدرة على اكتساب المهارات الخاصة بأدوات البرمجة وتطويرها ، وكذلك يكون له عقل يجيد التعامل مع الأسس الرياضية ، حيث إن مهنة البرمجة لا تعتمد على مجرد أداء المهام فقط وإنما تتطلب فكراً خصباً وذهناً حاضراً وحباً للإبداع في العمل والمثابرة عليه. ولتصميم وتطوير البرامج والتطبيقات بطريقة قياسية ، يجب أن يتكون فريق العمل الخاص بذلك من التالي :

- **محلو النظم (System Analysts)** : وهم الأشخاص القائمون على دراسة وتحليل متطلبات النظام ومدخلاته ومحركاته ، وكذلك تحديد الموارد الازمة لتنفيذها ، بالإضافة إلى بيان كيفية التنفيذ وشرح ديناميكية العمل وتنظيم العلاقات المختلفة بين الكائنات الموجودة بالنظام .
- **صممو النظم (System Designers)** : يأتي دورهم بعد مرحلة التحليل وتحديد الاحتياجات ، حيث يكون النظام بحاجة الآن إلى كيفية التطبيق من حيث الشكل العام وتصميم كائنات ونماذج النظام وبنية كل كائن على حده. وفي هذه المرحلة يتم تصميم نماذج وأشكال الشاشات ومواضعها وطرق عرضها وربطها مع بعضها البعض.
- **المبرمجون (المطورون) (Programmers (Developers))** : ويأتي دورهم بعد مرحلة التحليل والتصميم حيث يتم التنفيذ الفعلي للنماذج والشاشات المصممة وكتابة الشفرات (Source Code) المسئولة بدورها عن تشغيل النظام .

فمثلاً إذا كنا بصدد إنشاء نظام لإدارة شركة ما من الناحية المالية والتجارية ، فسيقوم المحللون بدراسة الدورة المستدية لهذه الشركة ، وكيفية تعاملها مع الشركات الأخرى ، وдинاميكية العمل من حيث المستندات المستخدمة في دورات العمل المختلفة ...الخ ، وكيفية تدفق البيانات من مرحلة إلى الأخرى ، وبالتالي تحليل النظام ككل بشكل متكامل ثم. بعد ذلك يأتي دور المصممين حيث يتم تصميم نماذج وأشكال الشاشات ومواضعها وطريقة عرضها وربطها ببعض والتي سيبرمجها المبرمجون ، وبعد ذلك يأتي دور المبرمجين حيث يتم التنفيذ الفعلي لما تم تصميمه سابقاً حيث يتم كتابة الشفرات اللازمة لإنشاء كل النماذج وربطها بعضها ببعض.

ومن خلال ما سبق ، يتضح أن المبرمج هو الشخص القائم على كتابة الشفرات اللازمة لبث روح الحياة في النظام وجعله وحدة واحدة متربطة يؤدي في النهاية - عند تشغيله من قبل المستخدم - جميع المهام الذي صمم من أجلها ، وبالتالي فالمبرمج هو حلقة الوصل بين الحاسوب الآلي والمستخدم ، فكلاهما لا يعرف لغة الآخر ، ولكن المبرمج يعرف لغة الاثنين.

والمبرمج ليس من تعلم لغة برمجية فحسب ، بل المبرمج هو من يعرف فن البرمجة ، أي كيف يضع الإستراتيجية المناسبة في المكان المناسب ، وهذه بحد ذاتها موهبة رياضية كالرسم والنحت ، أما ما تبقى من العمل البرمجي فلا يتعدى تطوير الذخائر والأدوات البرمجية وتشكيل الفكرة في قالب ذي طابع فني برمجي لإنتاج المنتج البرمجي.

ولكي يبدأ أي شخص بامتهان مهنة البرمجة كوظيفة يجب عليه أولاً التعرف على أنواع لغات البرمجة من حيث نقاط القوة والضعف في كل منها ، وكذلك التطبيقات الخاصة بكل منها ، وتلّم مبادئ البرمجة ومفاهيمها الأساسية والمشتركة بين جميع لغات البرمجة ، وهو الهدف الأساسي من هذا الكتاب الذي بين يديك ، فإذا كنت تزيد أن تبدأ في عالم البرمجة فعليك بدراسة هذا الكتاب وبالترتيب المذكور به .

أسئلة تقويم ذاتي

<p>1. ما هي أهم التحديات الرئيسية التي تواجه هندسة البرمجيات؟ وكيف يمكن مواجهتها من وجه نظرك؟</p> <p>2. الجدول التالي يحتوي العمود الأول منه على الفرق الأساسية لتطوير البرمجيات والعمود الثاني على وظائف هذه الفرق، أعد بناء الجدول بحيث يكون هناك توافق بين الفرق ووظائفها</p>	?								
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center; padding: 5px;">فرق تطوير البرمجيات</th> <th style="text-align: center; padding: 5px;">وظائفها</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;"> محللو النظم </td><td style="padding: 5px;"> <ul style="list-style-type: none"> • يقوم بتصميم نماذج وأشكال الشاشات مواضعها وطرق عرضها وربطها مع بعضها البعض. </td></tr> <tr> <td style="padding: 5px;"> مصممو النظم </td><td style="padding: 5px;"> <ul style="list-style-type: none"> • يقوم بتحليل متطلبات النظام ودخلاته ومخرجاته وكذلك تحديد الموارد اللازمة لتنفيذها. </td></tr> <tr> <td style="padding: 5px;"> المبرمجون </td><td style="padding: 5px;"> <ul style="list-style-type: none"> • يقوم بكتابة الشفرات المسؤولة بدورها عن تشغيل النظام. </td></tr> </tbody> </table>	فرق تطوير البرمجيات	وظائفها	محللو النظم	<ul style="list-style-type: none"> • يقوم بتصميم نماذج وأشكال الشاشات مواضعها وطرق عرضها وربطها مع بعضها البعض. 	مصممو النظم	<ul style="list-style-type: none"> • يقوم بتحليل متطلبات النظام ودخلاته ومخرجاته وكذلك تحديد الموارد اللازمة لتنفيذها. 	المبرمجون	<ul style="list-style-type: none"> • يقوم بكتابة الشفرات المسؤولة بدورها عن تشغيل النظام. 	
فرق تطوير البرمجيات	وظائفها								
محللو النظم	<ul style="list-style-type: none"> • يقوم بتصميم نماذج وأشكال الشاشات مواضعها وطرق عرضها وربطها مع بعضها البعض. 								
مصممو النظم	<ul style="list-style-type: none"> • يقوم بتحليل متطلبات النظام ودخلاته ومخرجاته وكذلك تحديد الموارد اللازمة لتنفيذها. 								
المبرمجون	<ul style="list-style-type: none"> • يقوم بكتابة الشفرات المسؤولة بدورها عن تشغيل النظام. 								

8. المسؤولية المهنية والأخلاقية لمهندسي البرمجيات

(Professional and Ethical Responsibility for Software Engineers) :

مثهم في ذلك مثل غيرهم من المهندسين ، فإنه يتوجب على مهندسي البرمجيات القبول بحقيقة أن مهامهم تتضمن مسؤوليات أكبر من مجرد استخدام المهارات الفنية حيث أن عملهم يتم ضمن إطار قانوني واجتماعي معين، ومن الواضح أن هندسة البرمجيات مقيدة بمجموعة من القوانين المحلية والوطنية والدولية، وتبعاً لذلك فإن عليهم أن يتصرفوا بطريقة مسؤولة من الناحتين الأخلاقية والمعنوية إذا كان لهم أن يحوزوا على الاحترام

كمحترفين في مجالات عملهم، ومن الديهي كذلك أن يتمسك هؤلاء المهندسون بالقواعد والمعايير المتعارف عليها في مجال الشرف والتزاهة والأمانة والاستقامة. كما ينبغي عليهم عدم استخدام مهاراتهم وقدراتهم بطريقة غير شريفة أو تسيء إلى سمعة المهنة، ومع ذلك فإن هناك مجالات لا تكون فيها معايير السلوك المقبول ملزمة بموجب القانون بل ترتبط بمقاييس فضفاضة للمسؤولية المهنية، وهذه بعض منها :

- **السرية (Confidentiality)** : يجب على المهندسين في العادة احترام خصوصية وسرية المعلومات الخاصة بعملائهم وأرباب العمل بصرف النظر عما إذا تم توقيع اتفاقية خاصة بالسرية أم لا.
- **الاختصاص (Competence)** : يتوجب على المهندسين الامتناع عن إعطاء معلومات خاطئة عن مستوى تأهيلهم واحتياطاتهم ، كما أن عليهم عدم قبول أي عمل خارج عن نطاق اختصاصهم وهم يعلمون ذلك.
- **حقوق الملكية الفكرية (Intellectual Property Rights)** : يجب أن يكون المهندسين مطلعين على القوانين المحلية التي تحكم استخدام الممتلكات الفكرية مثل براءات الاختراع وحقوق النشر والطبع والتاليف وغيرها، كما يجب عليهم أن يحرصوا على التأكد من حماية الممتلكات الفكرية لأرباب عملهم وعملائهم.
- **إساءة استخدام أجهزة الحاسوب الآلي (Computer Misuse)** : ينبغي على مهندسي البرمجيات عدم استخدام مهاراتهم الفنية في الإساءة إلى أجهزة الحاسوب الآلي التي تعود للآخرين وإساءة الاستخدام تتراوح بين أمور تافهة نسبياً (مثل ممارسة الألعاب على جهاز صاحب العمل) وأمور خطيرة جداً (كنشر الفيروسات مثلًا).
وفي هذا الخصوص تلعب الجمعيات والمؤسسات المهنية دوراً هاماً ، فالمنظمات مثل : "اتحاد أصحاب الآلات الحاسبة" ، و"معهد المهندسين الكهربائيين والإلكترونيين" ، و"جمعية الحاسوبات البريطانية" تقوم بنشر لواح السلوكي المهني أو قواعد أخلاقي المهنة. كما يتعهد أعضاء هذه المنظمات باللتزام بالقواعد المنشورة عند التوقيع على العضوية. وينبغي على مهندسي البرمجيات الالتزام بجعل مهنة تحليل وتصميم وتطوير واختبار وصيانة البرامج مهنة محترمة وذات فائدة وجدوى، ووفق التزامهم بصحة وسلامة

ورفاهية الجمهور فإن على مهندسي البرامج الالتزام بالمبادئ الثمانية التالية :

- ① **الجمهور (Public)** : يجب على مهندسي البرمجيات العمل بما يتوافق مع المصلحة العامة.
- ② **العميل وصاحب العمل (Client And Employer)** : يجب على مهندسي البرمجيات العمل بطريقة تخدم مصالح عملائهم وأرباب الأعمال وتكون متوافقة مع المصلحة العامة.
- ③ **المنتج (Product)** : يجب على مهندسي البرمجيات التأكد من أن منتجاتهم والتعديلات المتعلقة بها تقي بأعلى المستويات المهنية الممكنة.
- ④ **النزاهة (Judgment)** : يجب على مهندسي البرامج المحافظة على التكامل والاستقلالية في أرائهم المهنية.
- ⑤ **الادارة (Management)** : يجب على المدراء والقادة في مجال هندسة البرمجيات المساهمة والتعهد بالالتزام بالمنهج الأخلاقي في إدارة عملية تطوير وصيانة البرمجيات.
- ⑥ **المهنة (Profession)** : يجب على مهندسين البرمجيات تحسين سمعة وصورة المهنة وبما يتوافق مع المصلحة العامة.
- ⑦ **الزملاء (Colleagues)** : يجب على مهندسي البرامج أن يكونوا منصفين وعادلين مع زملائهم وأن يقدموا الدعم والمساندة لهم.
- ⑧ **الذات (Self)** : يجب على مهندسي البرامج المشاركة في برامج التعليم الدائم والمتعلق بممارسة مهنتهم كما أن عليهم تعزيز الأسلوب الأخلاقي في ممارسة هذه المهنة.

أسئلة تقويم ذاتي



1 ضع علامة (✓) أمام الإجابة الصحيحة وعلامة (X) أمام الإجابة الخطأ. صحق الإجابة الخطأ؟ :

- هندسة النظم جزء من هندسة البرمجيات.
- مصطلح هندسة البرمجيات يحتوي على شقين أحدها "هندسي" تطبق فيه النظريات والأساليب للحصول على الحلول المثالية للمشكلات والأخر "برمجي" يهتم بجميع مظاهر إنتاج البرمجيات.
- هندسة البرمجيات لا تهتم فقط بالعملية الفنية لتطوير البرمجيات ولكن تهتم أيضاً بالنشاطات الأخرى مثل إدارة مشروع البرمجيات.
- علوم الحاسوب تركز على أنظمة الحاسوب ولغات البرمجة لدراستها وتطويرها بينما هندسة البرمجيات تهتم بالمشاكل العملية لإنتاج البرمجيات.
- تحديد المتطلبات وقيود تشغيلها وتطوير النظام البرمجي والتحقق من أداء النظام البرمجي عبارة عن الأنشطة الأربع الأساسية في جميع عمليات البرمجيات.
- أدوات هندسة البرمجيات بالاستعانة بالحاسوب عبارة عن أنظمة برمجية مصممة لإسناد الأنشطة الروتينية لعمليات إنتاج البرامج.

2. رتب (ترتيباً تصاعدياً) المهام التالية التي يجب أن تتبع في تطوير المشاريع البرمجية :

إدارة المشاريع ، صيانة البرمجية ، تصميم البرمجية ، اختبار تكاملية البرمجية ، تحديد المتطلبات، توثيق البرمجية، تدبير الموارد، الترميز (عملية البرمجة)، التثبيت وتوزيع البرمجية، التدريب، تحليل المشكلة.

اسئلة تقويم ذاتي



اختر المصطلح المناسب والصحيح لكل من التعريفات التالية :

- تسليم المنتج البرمجي في الوقت المحدد.
- إمكانية التشغيل والتعامل مع أنظمة أخرى عادية أو شبكية.
- يمكن إعادة استخدامها مع مشاريع أخرى لأنها مكتوبة وفقاً لمعايير.
- يمكن اختبارها بسهولة.
- يمكن نقلها إلى أي نظام تشغيل أو حاسب بدون إعادة كتابة أي أجزاء جديدة.
- تعني توفر ربط يبني مناسب للمستخدم ومادة وثائقية كافية.
- يمكن صيانته إذا حدثت أي مشاكل أثناء التشغيل.
- تعني تنفيذ البرمجية في الوقت المحدد من خلال الموارد المتاحة.
- تعني عدم تذبذب أداء البرمجية من وقت لآخر.
- يتصرف النظام بشكل يتوافق مع المتطلبات الوظيفية.