مراحل تطور البرنامج
program development stages

1- مرحلة التعريف والتحليل .

2- مرحلة تصميم الخوارزمية .

3- مرحلة كتابة البرنامج .

4- مرحلة التنفيذ وإيجاد الاخطاء .

5- مرحلة الاختبار.

6- مرحلة التوثيق والصيانة .

1- مرحلة التعريف والتحليل:

ان تعريف المسالة هو عبارة عن دقة التعبير في تطبيق المسالة بحيث يجعلها معروفة ومفهومة و بصورة واضحة وبدون أي غموض لجميع الاشخاص العاملين ضمن مجال الاختصاص الذي تخضع له المسالة.

اما تحليل المسالة ووضع طريقة الحل فهو من المراحل الصعبة، ومن اجل الوصول الى حل صحيح فأن كثير من القوانين والطرق الرياضية المناسبة لحل المسالة يجب ان تستعمل ، ففي هذه المرحلة يجب تحديد:

1- مخرجات البرنامج : هى النتائج والمعلومات المراد التوصل اليها.

2- مدخلات البرنامج : البيانات اللازم الحصول عليها لمعرفة النتائج، وتحديد نوعها وتنظيم ادخالها.

3- عمليات المعالجة : العمليات الحسابية والخطوات المنطقية التي نجريها على المدخلات للحصول على المخرجات.

2- مرحلة تصميم الخوارزمية :

بعد اختيار الحل الامثل وتحديد كل ما تشمله من علاقات رياضية ، يتم تنظيم طريقة الحل المقترحة على شكل خطوات متسلسلة ومترابطة منطقياً ، دقيقة الوصف ، يعبر عنها بالغة العادية ألانه يحكمها المنطق الرياضي ، ومجموع هذه الخطوات يسمى بالخوارزمية، وقد يكون للحل الواحد اكثر من خوارزمية واحدة إلا ان جميع هذه الخوارزميات تودي الى نفس النتيجة ولكن بطرق مختلفة ومتفاوتة .وقد يكون في كثير من الاحيان وضع مخطط وصفي تسلسلي يسمى المخطط الانسيابي (flowchart) وذلك باستخدام مجموعة من الاشكال الرمزية التي تكون اكثر وضوحا وانسب فهما.

وبواسطة المخطط الانسيابي يمكن ملاحظة كيفية تتبع الحل والعلاقات المنطقية التي تظهر في الحل .

وقد يقوم المخطط الانسيابي مقام الخوارزمية ويعتبر وثيقة مهمة من وثائق حل المسالة يمكن الرجوع اليها عند الضرورة وفي غياب المبرمج او المحلل الذي قام بتصميمها ومن ناحية اخرى فان استخراج الخوارزمية من خريطة سير العمليات (المخطط الانسيابي) يكون اسهل بكثير من كتابة الخوارزمية نفسها .

3- مرحلة كتابة البرنامج :

بعد عرض المسالة وحلها بصيغة خوارزمية او مخطط انسيابي يتم تحويل صيغة الحل هذه الى صيغة يمكن للحاسبة ان تتعامل معها أي تحويلها الى (برنامج) بإحدى لغات البرمجة كلغة فورتران (Fortran) او لغة باسكال (Pascal) او لغة (c++) وغيرها . اما امر تحديد اللغة المراد استخدامها فيتوقف على طبيعة المسالة المراد حلها وحسب توفر اللغة وامكانياتها ، تقسم لغات البرمجة الى قسمين :

أ- لغات ذات المستوى الواطئ :
Low Level Languages

تشمل لغة الماكنة (Machine Language) التي يعبر عنها بـ(0,1) ، ولغة التجميع (Assembly Language) التي تعتمد على الرموز المعبرة في كتابة البرنامج ، وتتميز هذه اللغات بالصعوبة في كتابة البرامج ، اضف الى ذلك اعتمادها الماكنة فعندما ينتقل برنامج مكتوب بلغة الماكنة او بلغة التجميع من حاسبة الى اخرى فان ذلك يتطلب اجراء تغييرات عديدة ، إلا أن هذه اللغات تمتاز بسرعتها في تنفيذ الايعازات لأنها لا تحتاج الى مترجم لأنها لغة الماكنة.

ب- لغات ذات المستوى العالي :

High Level Language

تشمل اللغات التي تتميز عن اللغات التي سبقتها ، بكونها قريبة من المستفيد واعتمادها على الماكنة اقل حيث ان عملية نقل البرنامج من حاسبة الى اخرى لايستوجب إلا اجراء بعض التغييرات الطفيفة اضف الى ذلك ان البرمجة بهذه اللغات هى اسهل بكثير من اللغات ذات المستوى الواطئ ومن امثلة هذه اللغات لغة بيسك وباسكال و++C ..... الخ.

4- مرحلة التنفيذ وإيجاد الاخطاء:

تتلخص هذه المرحلة بعملية ادخال البرنامج الى الحاسبة عن طريق احدى وسائل الادخال ومن ثم الايعاز للحاسبة بتنفيذ البرنامج ، فإذا كانت اللغة التجميعية او احدى لغات ذات المستوى العالي يتم استدعاء المترجم لتحويلها الى لغة الماكنة لان الحاسبة لأتفهم غيرها . ويتخذ المترجم احدى الصيغ التالية :

1- المؤلف compiler:

هو برنامج يقوم بتحويل البرامج المعبر عنها بإحدى اللغات ذات المستوى العالي الى لغة الماكنة.

2- المفسر Interpreter:

هو برنامج مهمته نفس مهمة المؤلف ألا أنه يقوم على اساس ترجمة الايعاز وتنفيذه ثم الانتقال الى ايعاز اخر على خلاف ما يقوم به المؤلف الذي يترجم البرنامج بشكل كامل قبل تنفيذه .

3- المجمع Assembler :

هو برنامج يقوم بتحويل البرامج المعبر عنها باللغة التجميعية الى لغة الماكنة .

وعند تنفيذ البرنامج ستظهر انواع من الاخطاء البرمجية (Types Of Errors) وهى :

1- الاخطاء الاملائية والقواعدية (Syntax errors).

2- الاخطاء التنفيذية (Run Time errors).

3- الاخطاء المعنوية (Semantic errors).

1- الاخطاء الاملائية والقواعدية (Syntax errors):

هى الاخطاء الاكثر شيوعا وانتشارا ولكنها سهلة العلاج ، وهى التي تظهر اثناء كتابة البرنامج عندما لايتم استخدام اللغة البرمجية بالشكل الصحيح .

فالأخطاء الاملائية تحدث غالبا بسبب وجود خطا املائي لغوي (طباعي) في كتابة الاوامر الخاصة باللغة البرمجية المستخدمة ، مثل كتابة كلمة محجوزة بشكل اخطاء او استخدام غير صحيح لدالة معينة (يعني خطا في اسم الدالة).

اما الاخطاء القواعدية :

هى التي تحدث نتيجة عدم فهم المبرمج لقواعد اللغة البرمجية المستخدمة وعدم التزامه وخروجه عن القواعد المتبعة في صياغة وتركيب الجملة مثلا نسيان اغلاق قوس و قد تم فتحه مسبقا او نسيان علامات الاقتباس في النصوص وغيرها .

2- الاخطاء التنفيذية (اخطاء وقت التشغيل) : (Run Time errors)

وهى اخطاء التي تظهر بعد ترجمة البرنامج اثناء التنفيذ، فيحصل شي غير متوقع وغير قادر على المعالجة من تلقاء نفسه ، وبالتالي يتوقف العمل، مثلا وجود نقص في المدخلات .مثال اخر لو كتبت برنامج لقراءة البيانات من ملف معين فعند تشغيل البرنامج رغم ان البرنامج صحيح سيظهر خطا اذا لم يكن الملف موجود او اذا كان الملف موجود وتالف . وغالبا يمكن حل هذه الاخطاء عن طريق اعادة صياغة البرنامج وتشغيل البرنامج من جديد .

3- الاخطاء المعنوية (Semantice errors):

في لغات البرمجة الخطأ المعنوي يوضح سلوك الحاسوب في تنفيذ برنامج معين مكتوب بإحدى لغات البرمجة ، هذا السلوك يوضح تنفيذ البرنامج خطوة بعد خطوة للحصول على النتائج النهائية والمنطقية لذلك البرنامج.

في أي لغة برمجية بعد تصحيح الاخطاء الاملائية والقواعدية من البرنامج يمكن ان يظهر خطا اخر بعد تنفيذ البرنامج يعطي سلوك غير صحيح لتنفيذ البرنامج ويبين ان النتائج غير صحيحة وغير متوافقة مع ما صمم من اجله البرنامج.

هذا النوع من الاخطاء صعب الاكتشاف وعندما تكتشفه فقد تصرف الكثير من الوقت والجهد لمعالجته وتصحيحه ، وهو عادة يظهر كنتيجة غير صحيحة وليس كخطأ واضح .

وهذه الاخطاء تحدث عندما يرتكب المبرمج خطأ في بناء سلسلة التعليمات اللازمة للإعطاء النتيجة المطلوبة أي ان الخطأ يكون في منطق الحل .

وتحدث الاخطاء المعنوية في عدة حالات منها مثلا محاولة قسمة عدد على صفر،او في حالة الحلقات التكرارية الغير المنتهية ،او قراءة متغيرات معينة وهى ليست معرفة، او لدينا عملية رياضية معقدة تتطلب في احد اجزاءها جمع رقمين ثم جاء المبرمج من دون قصد و وضع اشارة عملية الطرح بدل من الجمع،وغيرها من الحالات الاخرى.

* اسباب الخطأ في البرنامج:

هناك عدة اسباب تؤدي الى فشل البرنامج وعدم تحقيق الهدف الذي كتب من اجله البرنامج وهي:

1. عدم الفهم الكامل للمشكلة من خلال دراستها وتحليلها بشكل صحيح.
2. الخوارزمية غير صحيحة فإذا كانت المسالة معقدة نوعا ما فمن الافضل ان توضع الخوارزمية من قبل المبرمج وتدقق من قبل شخص اخر للتأكد من صحتها قبل تحويلها الى برنامج.
3. عدم الإلمام بلغة برمجية بشكل جيد والذي يسبب بعض الاخطاء.
4. الاخطاء المطبعية التي ترتكب اثناء ادخال البرنامج الى الحاسبة .
5. وجود اخطاء في مترجم اللغة مع ان هذا الخطاء نادر الحصول

* ادوات التنقيح (التصحيح):

عند اكتشاف خطا في البرنامج يجب على المبرمج ان يتتبع الخطأ في البرنامج ليكتشف موقعه ومن ثم تصحيحه وأكثر الاساليب شيوعاً في تتبع الخطأ هي:

1. وضع عبارات Print في مواقع متعددة في البرنامج حيث تطبع قيم المتغيرات التي هي موضع الشك ويستمر الفحص لحين اكتشاف الاخطاء .

2. تستخدم عبارة التوقف Stop :

لتتبع الاخطاء في البرنامج حيث توضع في مواقع متعددة ومهمة في البرنامج وعند توقف البرنامج نتيجة لتنفيذ هذه العبارة يعطى بهذه الحالة للتعرف على قيم المتغيرات بان تستخدم عبارة الطبع Print وبعد ذلك يمكن المباشرة باستمرار تنفيذ البرنامج باستخدام مفتاح F5 (cont) .

3. في بعض انواع الحاسبات توجد عبارة F7 (TRON) حيث تستخدم في اكتشاف موقع الاخطاء.

5- مرحلة اختباروفحص البرنامج Program Testing :

لايمكن اعتبار البرنامج على انه في وضعه النهائي بعد كتابته مباشرة وقبل ان يتم فحصه وتصحيح الاخطاء الموجودة فيه. ففي بعض الحالات فحص البرنامج وتصحيحه قد يحتاج الى وقت اطول من وقت تطوير البرنامج ذاته. ففي كل الاحوال يجب فحص البرنامج بعد تطويره للتأكد من خلوه من الاخطاء فهنالك اعتقاد بان حوالي 50% من الوقت المصروف في تطوير البرنامج يعرف عادة في فحص البرنامج في حين تأخذ الخطوات الاخرى كالتصحيح والتوثيق 50% من الوقت .

كلما كان البرنامج اكثر تعقيدا ازدادت صعوبة فحصه تبعا لذلك ويتم في هذه المرحلة تنقية البرنامج من الاخطاء المنطقية بشكل كامل ويتحقق ذلك بإدخال مجاميع افتراضية مختلفة من البيانات كمدخلات للبرنامج ومن ثم التحقق من صحة البرنامج ومن الافضل ان تكون البيانات في مجاميع مختلفة لضمان اختبار جميع مقاطع البرنامج والتحسب لجميع الاحتمالات المختلفة.

وبعد اثبات صحة طريقة الحل المقترحة يمكن تنفيذ البرنامج على البيانات الحقيقية وطباعة النتائج المطلوبة.

6- مرحلة التوثيق والصيانة:

أ- مرحلة التوثيق Decumentation:

المقصود بالتوثيق هو ضرورة وجود قائمة بالبرنامج ووصف للمعلومات الداخلة والخارجة ومخطط او خوارزمية توضح طريقة الحل وقائمة بالأخطاء المتوقع حدوثها ومرشد للمستفيد الذي يدل على كيفية تشغيل البرنامج وكذلك تضمين البرنامج على بعض الشروحات الضرورية التي الغرض منها تعريف المتغيرات المستخدمة وتوضيح عمل كل جزء من اجزاء البرنامج . وأحيانا خلال البرنامج تكتب ملاحظات باستخدام عبارة Rem وتعتبر جزء من التوثيق .

الوثائق المرفقة للبرنامج

تزداد الوثائق المطلوبة للبرنامج وتتنوع في حالة كون المستفيد معرفته بالبرنامج محدودة وتختلف الوثائق من برنامج الى اخر حيث انها تعتمد على تعقيدات البرنامج واستخداماته . وكما يجب عدم الاسراف في تصميم المعلومات والمستندات الموثقة لأنها قد تصبح عبئا على المستفيد وعندئذ تقل فعاليتها، ويمكن تصنيف مستويات مختلفة تحت قائمة اعداد الوثائق وتشمل :

1-المواصفات الوظائفية التي توضح مايلي :

أ- الاعمال التي يقوم بها البرنامج .

ب- المشكلة التي يحلها .

ج- الطريقة المستخدمة في الحل .

د- البيانات المدخلة والمخرجات الناتجة .

2. مواصفات البرنامج وتوضيح كيفية استخدام البرنامج .

3. اجراءات التشغيل وتشمل :

أ- حاجة الماكنة.

ب- نوع الحاسبة.

ج- اقل واكبر حجم الخزن .

د- الاجهزة الخارجية .

هـ- الخزن الخارجي أي الاضافي (الاقراص).

و-حاجات تهيئة المعلومات .

ز- تعليمات حول تشغيل البرنامج.

4 . اجراءات الاختبار والصيانة :

التي توضح كيفية اختبار البرنامج وتشمل ايضا بيانات الاختبار ونتائج استخدام بيانات الاختبار وقائمة بالمتغيرات المستخدمة في البرنامج.

ب- مرحلة صيانة وإدامة البرنامج:

تشمل هذه المرحلة على الاضافات المستقبلية التي سببها ظهور حالات جديدة ثم الاخذ بنظر الاعتبار هذه الاحتمالات اثناء تصميم البرنامج بالإضافة الى معالجة الاخطاء التي لم تعالج في مرحلة الاختبار .