

هندسة البرمجيات

الوحدة الرابعة

تصميم البرمجيات

م. ماجد ظاهر

الفصل الدراسي الأول

2022 - 2023

محتويات الوحدة

- أساسيات تصميم البرمجيات
- التصميم على أساس انسياب البيانات
- التصميم الكينوني
- طرق التصميم المستندة إلى البيانات
- التصميم الواقع

أساسيات تصميم البرمجيات

- تصميم البرمجيات هو عملية ابداع وتطوير واختيار الحلول المناسبة وتقديم وصف دقيق للبرامج من أجل تحقيق متطلبات الجهة المستفيدة ضمن حدود التقنيات والإمكانات المسموح بها.
- يمكن اعتباره انه وصف تفصيلي لطريقة بناء البرمجيات يوضح المفاهيم الأساسية ويساعد على اختيار طريقة الحل المناسبة.
- عبارة عن ادراك وتخطيط أو تطوير أو بناء خطة العمل .
- تصميم البرمجيات عبارة عن تخطيط يبين الخصائص الرئيسية للبرنامج المقترح.

... تابع

- التصميم الجيد للبرمجيات هو ذلك التصميم الذي يوازن بين:-
- المهمات التي تؤديها البرمجيات
- مجموعة من المواصفات وهي:-
 - كفاءة البرمجيات
 - امكانية الاستخدام المتكرر للبرمجيات
 - قابلية البرمجيات للتحويل والتطوير
 - تكلفة البرمجيات
 - تسليم البرمجيات الى الجهة المستفيدة في الوقت المحدد
 - المرونة في الاستخدام
 - سهولة الصيانة

... تابع

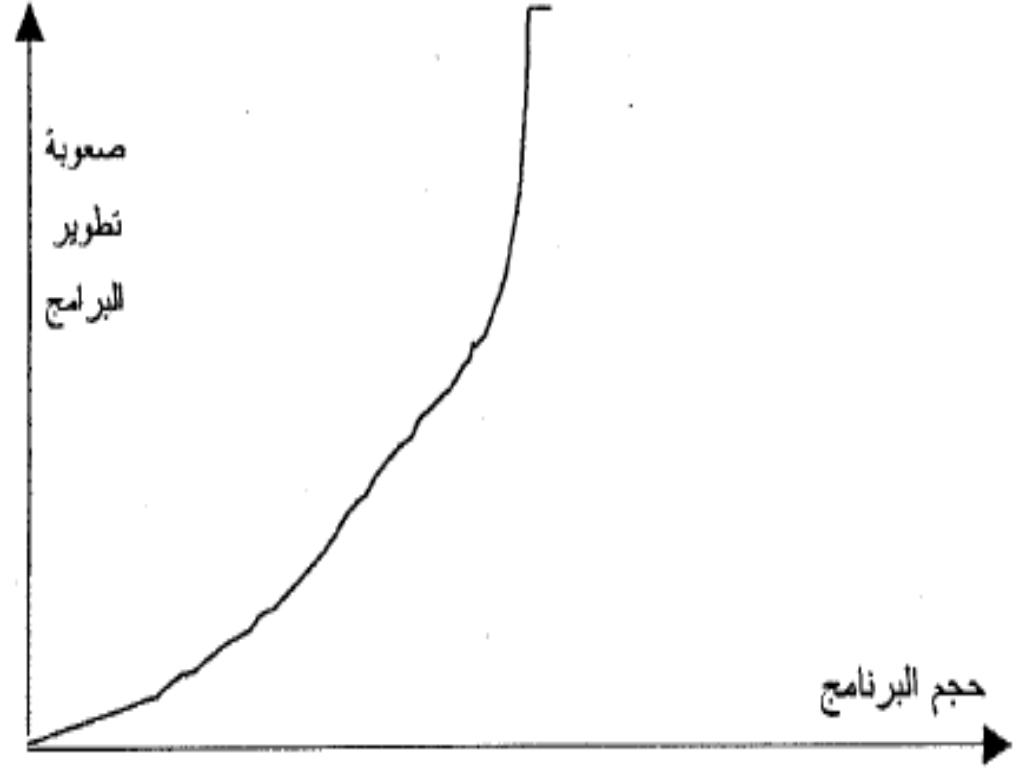
- مدخلات التصميم هي متطلبات الجهة المستفيدة وعلى المصمم معرفة المدخلات والمخرجات والمهمات التي تضعها الجهة المستفيدة.
- يستلزم من المصمم عند تصميم البرمجيات الجودة معرفة التالي:-
 - أهداف النظام المطلوب تصميمه
 - وظيفة الجهة المستفيدة
 - التغييرات المستقبلية المحتملة
 - أجهزة الحاسوب المتوفرة
 - تكلفة التصميم من حيث الوقت، الأموال، الكفاءة، الصيانة، سهولة الاستخدام.

... تابع

- ان مخرجات هذه المرحلة ستكون مدخلات للمرحلة اللاحقة وتتضمن هذه المخرجات ما يلي:-
 - تصميماً هيكلي يبين العلاقات بين أجزاء البرمجيات.
 - مواصفات كل جزء من أجزاء البرمجيات المراد بناؤها
 - تصميم للبيانات الجديدة التي يمكن أن يستفيد منها النظام البرمجي

العلاقة بين حجم البرنامج وصعوبة التطوير

- كلما كان حجم المشروع كبيراً كانت الصعوبات أكثر. ويمكن تقسيم المسألة الي مقاطع يتم ايجاد الحلول لكل مقطع بصورة منفردة. وكلما كانت مقاطع المسألة مستقلة عن بعضها البعض استطعنا تذليل الصعوبات.



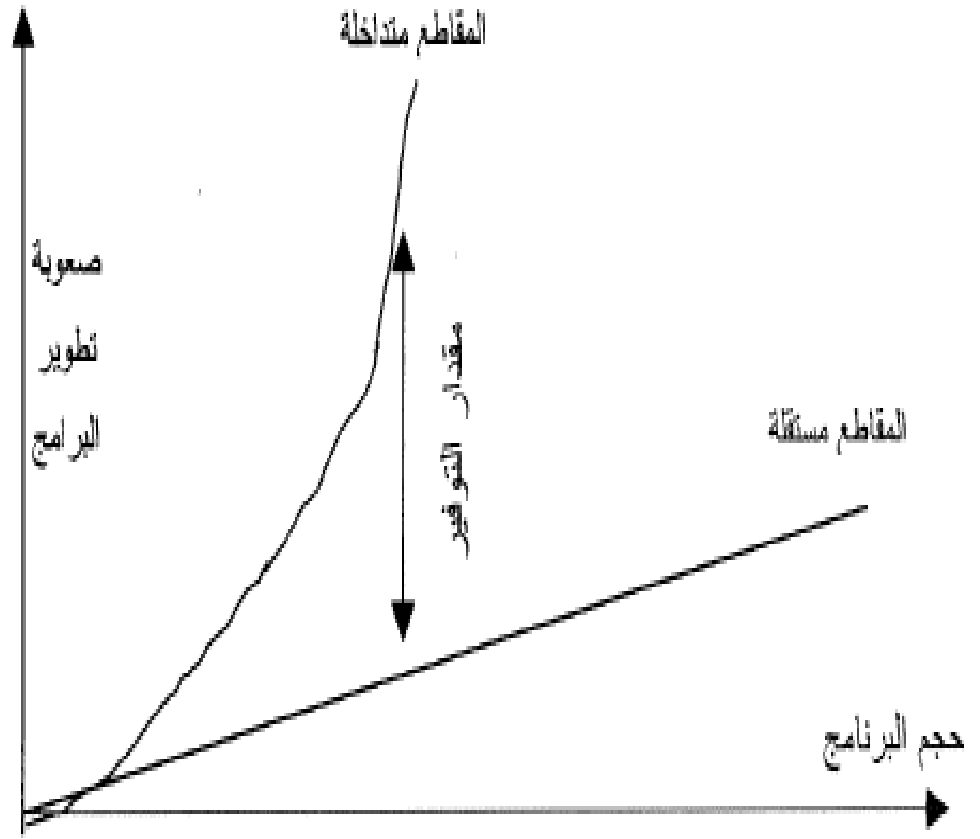
... تابع

- تقسيم المشروع البرمجي الى مقاطع مستقلة يساعد على أداء مجموعة من المهام منها:-

- سهولة فهم جمل المقطع وتعديله
- سهولة اختباره وتصحيحه أو تعديله
- سرعة تطوير البرمجيات وبساطتها
- زيادة الانتاجية ورفع الجودة

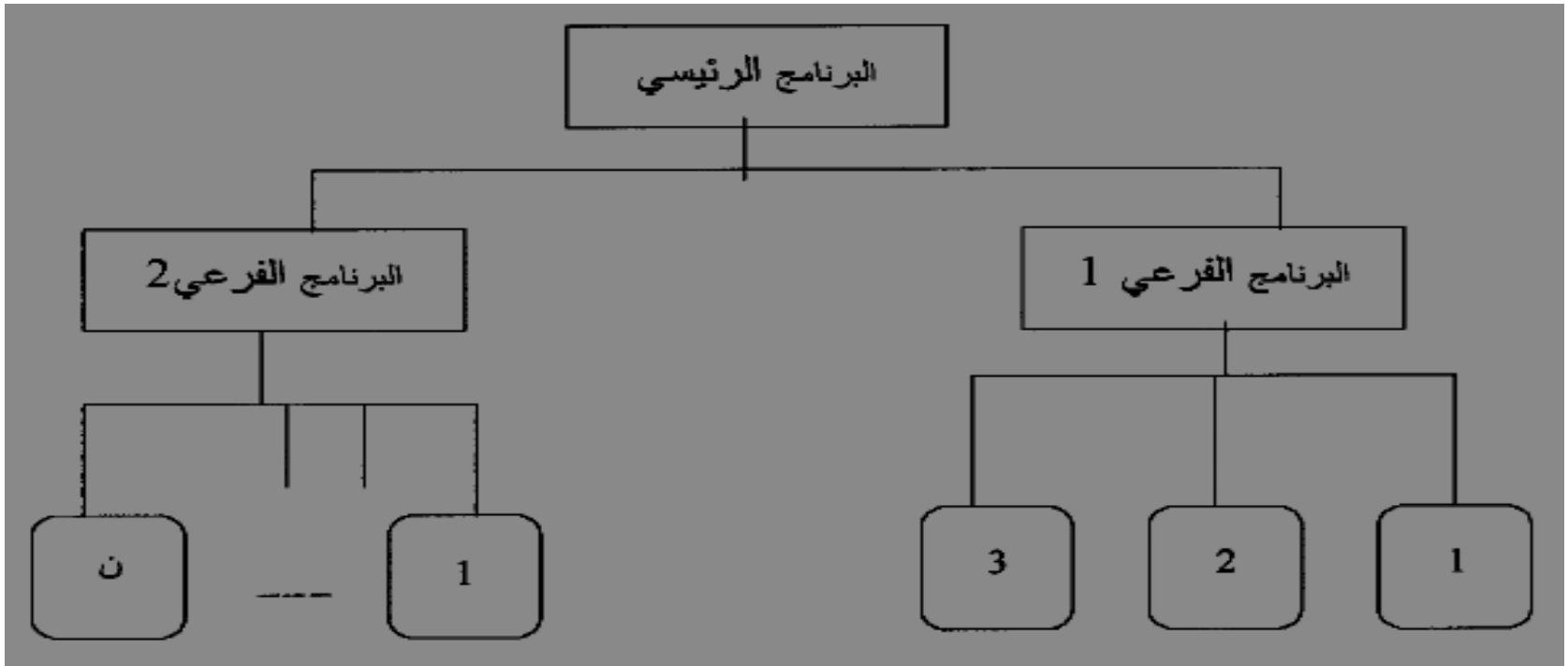
... تابع

- كلما كانت المقاطع مستقلة عن بعضها البعض فإن الصعوبات تتناسب طردياً مع حجم البرنامج.



... تابع

- تحتاج استقلالية البرامج الى تصميم هيكلي من أجل ربط الاجزاء مع بعضها للحصول على المنتج النهائي.
- المخططات هي الوسيلة المرئية التي تبين العلاقات بين مقاطع البرنامج الواحد في شكل يسهل تتبع تدفق البيانات وتحديد مساراتها.



... تابع

- **خاصية التماسك (cohesion)** في المقطع البرمجي : هي قابلية المقطع البرمجي لتنفيذ مهمة واحدة دون الحاجة الى استدعاء برامج أو مقاطع أخرى.

- **مستويات التماسك:-**

- التماسك الوظيفي Functional Cohesion
- التماسك المعلوماتي Information Cohesion
- التماسك الاتصالي Communicational Cohesion
- تماسك البرامج الفرعية Procedural Cohesion
- التماسك المؤقت Temporal Cohesion
- التماسك المنطقي Logical Cohesion
- التماسك العرضي Coincidental Cohesion

- يعد التماسك الوظيفي والتماسك المعلوماتي هما الاقوى بينما التماسك المنطقي والتماسك العرضي هما الاضعف.

... تابع

- التماسك العرضي:

– عندما يقوم البرنامج او المقطع البرمجي بأداء اكثر من مهمة غير مترابطة.

– يصعب صيانة هذا النوع او الاستفادة منه في برمجيات اخرى.

– ويمكن تحسينه بتجزئة البرنامج الى اجزاء اخرى,

– هذا النوع ايضا يعتبر الاسوأ وغير مقبول على الاطلاق.

- التماسك المنطقي: يعد البرنامج متماسكا منطقيا اذا كان يؤدي مجموعة من المهمات المترابطة.

- التماسك المؤقت: يكون البرنامج متماسكا بصورة مؤقتة اذا كان يؤدي مجموعة مهام مترابطة من حيث الزمن.

... تابع

• التماسك الاجرائي:

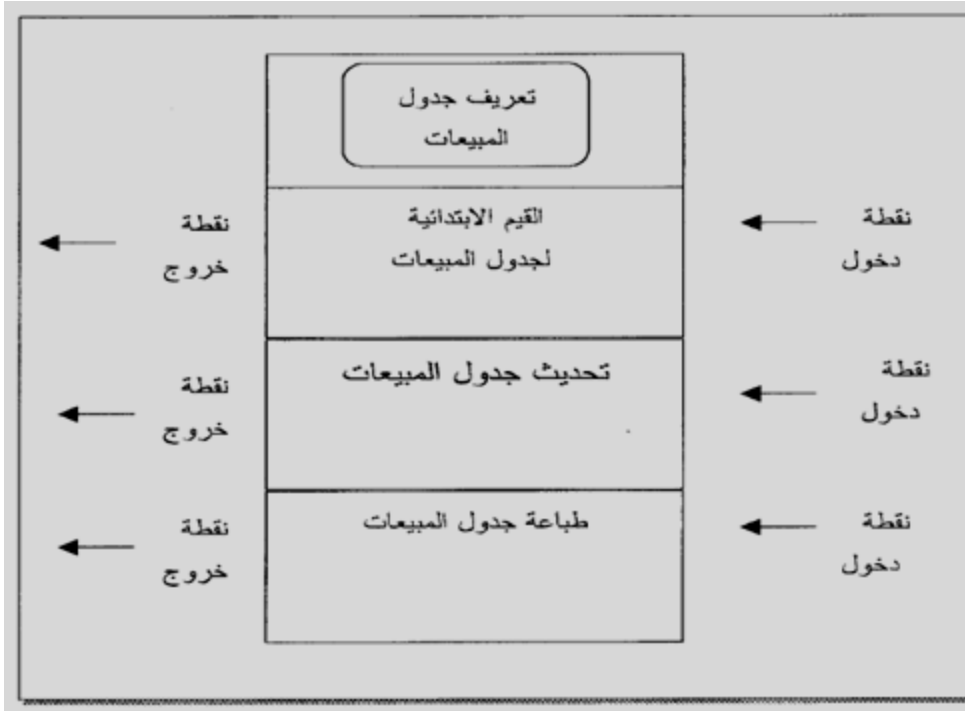
- يعد البرنامج متماسكا فرعيا اذا كان يؤدي مجموعة مهام ذات علاقة بالبرنامج الفرعي نفسه,
- ويعتبر اقوى من التماسك المؤقت لعدة اسباب منها: ان المهام في البرنامج الفرعي ذات طبيعة واحدة بينما في التماسك المؤقت كان عامل الوقت هو الذي يجمع بين المهام.

• التماسك المنطقي:

- يشبه تماسك البرامج الفرعية, يؤدي مجموعة مهام ذات علاقة بالبرنامج الفرعي نفسه, بالإضافة الى ان المهام جميعها تتعامل مع البيانات نفسها.
- ويحدث عندما تتضمن الوحدة الوظيفية مهاما عدة تتبادل البيانات فيما بينها.

... تابع

التماسك المؤقت: يقال اذا كان يؤدي عددا من المهام لكل مهمة نقطة بداية خاصة بها، ويصاحبها برنامج مستقل خاص بها. وتستخدم المهام جميعها تركيب بيانات واحدا. وهو تطبيق لأنواع البيانات الموجزة Abstract Data Types



التماسك الوظيفي:

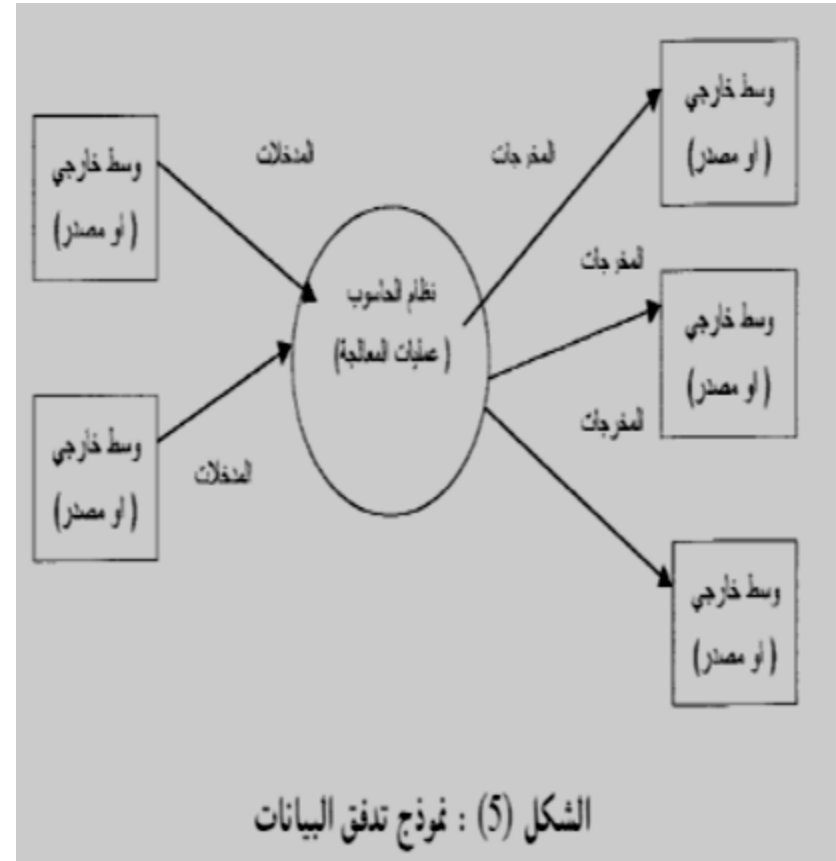
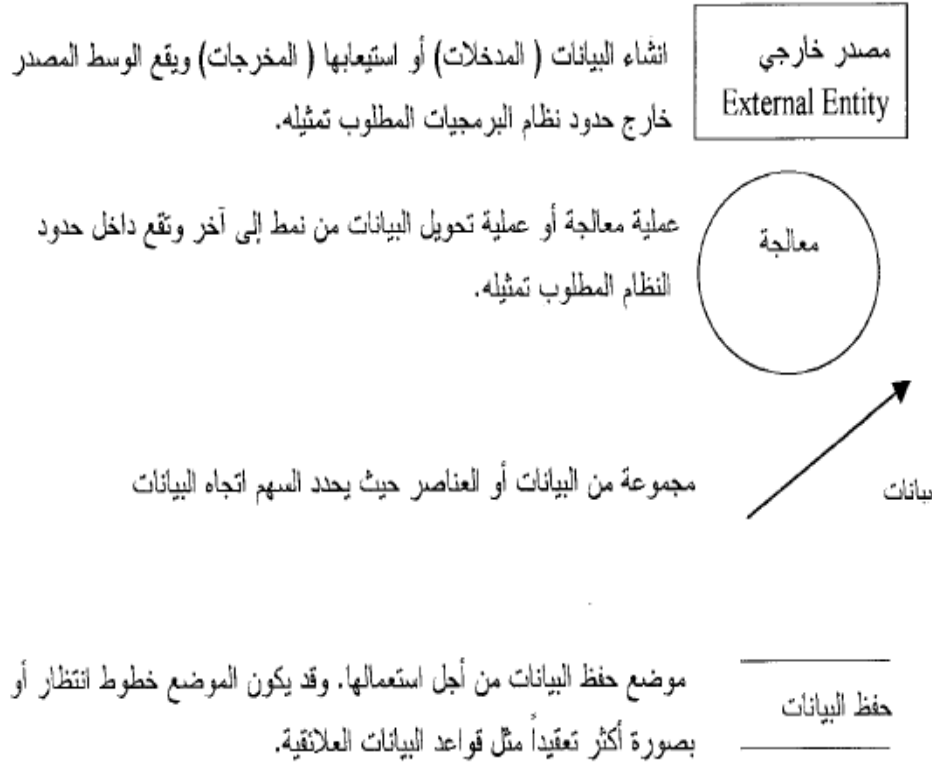
- عندما ينفذ البرنامج مهمة واحدة فقط، او هدفا واحدا، فان هذا النوع يسمى التماسك الوظيفي. ويمكن استخدامه في برامج أخرى.
- ويعد البرنامج ذو التماسك الوظيفي المصمم بصورة جيدة والذي اجتاز الاختبارات والذي يشمل على الملاحظات والتوثيق الشاملين بمثابة الاستثمار الحسن للمؤسسة.

التصميم على اساس انسياب البيانات

Method (DFD) Data-Flow Diagram

- تعتمد هذه الطريقة على انسياب البيانات خلال المنتج مع العمليات التي تجري على هذه البيانات في اثناء انسيابها, ويستخدم عند تصميم برامج ذات حجم متوسط او كبير.
- يبدأ العمل بطريقة التصميم على اساس انسياب البيانات مع العمليات التي تجري عليها. وأول خطوة هي بناء مخطط انسياب البيانات.
- تعتبر خرائط تدفق البيانات من الوسائل المفيدة في توضيح انسياب البيانات, وخرائط تدفق البيانات عبارة عن وسائل مساعدة لتطوير انظمة البرمجيات.
- تدفق البيانات عبارة عن نموذج يمثل حركة المعلومات من مواضيع معينة عبر مجموعة من المعالجات الى مواضع اخرى.

... تابع



الشكل (6) : الرموز الأساسية المستخدمة في خرائط تدفق البيانات

الطرق التي تؤدي الى افضل تصميم

1. البدء بدائرة كبيرة واحدة وتقسيمها الى دوائر اصغر.
 2. البدء من البيانات الخارجة من المنتج وتفحصها بالاتجاه المعاكس.
 3. البدء من البيانات الداخلة الى المنتج وتفحصها بالاتجاه نحو الامام.
- الخطوة التالية بعد الانتهاء من تصميم انسياب البيانات فهي تحويلها الى مخطط هيكلية,والذي يهدف الى تحديد اهم العمليات التي تجري على البيانات.

التصميم الكينوني

Object-Oriented Design

- هو احد انواع تصميم البرمجيات الذي ينظر الى البرمجية على انها مقسمة الى مجموعة من الاجزاء تسمى كينونات, تتعامل وتتفاعل مع بعضها البعض لتحقيق اهداف البرمجية.
- حيث يعتمد هذا التصميم على حالات منها: ان البرمجيات عبارة عن مجموعة من الكينونات تتعامل مع بعضها البعض ولكل كيان استقلاليته وبياناته الخاصة به.
- بناء برامج كينونات يستلزم الامر اتباع اربع خطوات منها:
 - تحديد تركيب الكينون المطلوب
 - تحديد المتغيرات التي يتضمنها كل كيان وتعريف كل متغير
 - تحديد البرامج الفرعية المطلوبة لتنفيذ عمليات الكينون
 - تحديد اهداف كل برنامج فرعي وأهمية المتغيرات الاساسية وتحديد الشروط الاولوية والنتائج المتوقعة.

الركائز الأساسية في التصميم الكينوني للبرمجيات

- 1- الكينون:- هو وحدة برمجية لها استقلاليته، وهو تجسيد لشيء واقعي يؤدي مهمة محددة يحتوي بياناته الخاصة به التي تتعلق بالشيء الذي يجسده، كما يحتوي على العمليات التي يسمح بتنفيذها على تلك البيانات.
- 2- العمليات:- يتطلب الأمر تصميم كينون يحتوي على مجموعة عمليات بالإضافة الى اجراءات تساعد المستخدم على تنفيذ مهامه.
- 3- الاشعارات:- المهام التي يقوم بها كل كينون هي بمثابة الخدمات التي يقدمها للكينونات الأخرى.
- 4- برمجة الكينونات
- 5- مزايا برمجة الكينونات في تصميم البرمجيات

طرق تصميم المستندة الى البيانات

- تعتبر من أول الطرق المستخدمة في تصميم البرمجيات وأهمها ويوفر وصفاً إجرائياً لإنتاج البرمجيات وهي ليست مخططات انسياب البيانات DFD فهي تستخدم الأسلوب المتسلسل أو الهرمي لتمثيل تركيب المعلومات في البرمجة.
- تشترك هذه الطرق مع طرق التصميم الكينوني في أكثر من صفة:
 - كلاهما يهتم بأنماط وكينونات التطبيقات على أرض الواقع للبرمجية
 - كلاهما يعتمد على المعلومات أساساً للتنفيذ
 - كلاهما يعتمد تمثيل البيانات لبناء تمثيل مناسب للبرمجية
 - كل منهما يمتلك وسيلة محددة للتصميم وتسلسل البيانات بينما طريقة التصميم الكينوني تمتلك الصف Class
 - كلاهما يطبق فكرة تجريد البيانات ويستخدم مجموعة من العمليات ذات العلاقة

... تابع

- ... ويختلفان بتعريف الكينون في الأولى يعتبر بيانات فقط اما الثانية بيانات ومعالجة.

- طريقة التصميم الكينوني تمتلك مواصفات هامة لا تتوفر بالطريقة الأولى مثل:-

– تصميم جاكسون لتطوير الأنظمة

– التصميم وفق تراكيب البيانات لتطوير الأنظمة

أولاً: تصميم جاكسون لتطوير الأنظمة

Jackson System Development (JSD)

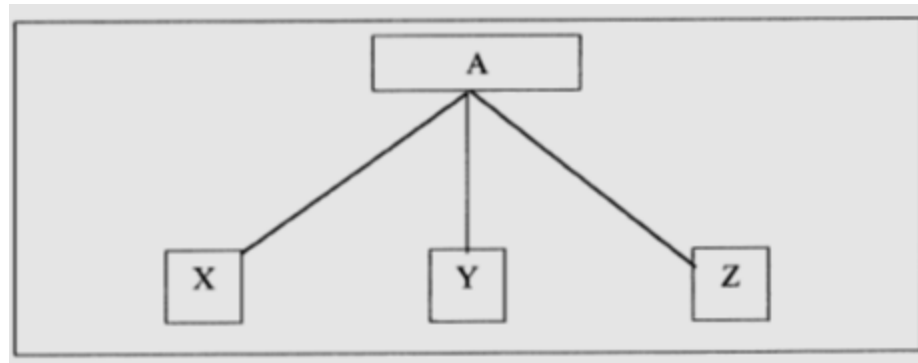
- تعتبر طريقة متكاملة من التحليل فالتصميم الى التنفيذ والتطبيق.
- تم تطويرها في منتصف السبعينيات وتعد نموذجاً لدورة حياة البرمجيات في التصميم والبناء.
- تستند في عملها في تحليل المتطلبات وتصميم البرمجيات الى المفهوم الهندسي العلمي من حيث التركيب والتحديث المستمر والتدرج في التصميم من الهام الى الخاص.
- ينطلق مصممو هذه الطريقة من فكرة ان مواصفات اي برنامج تعتمد على المدخلات والمخرجات فهما اللذان يحددان الشكل العام.

خطوات طريقة جاكسون

1. تخطيط الاشكال التركيبية للبيانات: تبين حركة البيانات من النظام كمخرجات والى النظام كمدخلات.
2. ايجاد شكل تركيبي واحد للبرنامج وذلك من خلال مجموعة الاشكال البيانية التركيبية.
3. تحديد كافة العمليات الاساسية التي يستخدمها البرنامج في تنفيذ المهمة المطلوبة وتكتب حسب مفهوم لغة البرمجة المستخدمة.
4. ربط العمليات الاساسية بمواقعها المناسبة في الشكل التركيبي للبرنامج.
5. تحويل تركيبة البرنامج من الأشكال التخطيطية الى الشكل الانشائي

... تابع

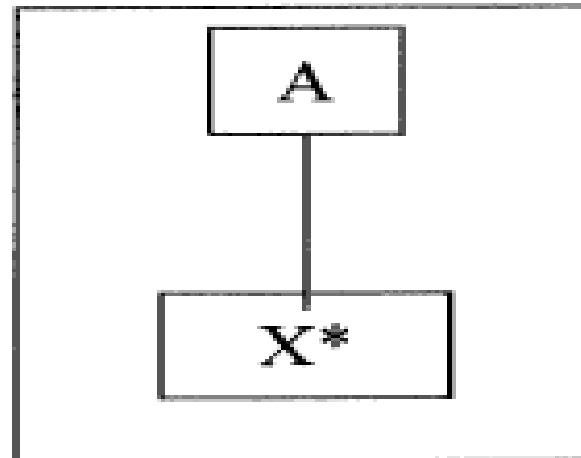
الأشكال التركيبية في طريقة جاكسون تنقسم الى ثلاثة أنواع:-
1. التابع:-



يفيد الشكل أن تركيب البيانات A يتكون من X ويتبعه Y ويليه Z. أما معناه في مفهوم البرمجة فهو أن المقطع البرمجي A يتطلب تنفيذ المقطع X، Y والمقطع Z على الترتيب.

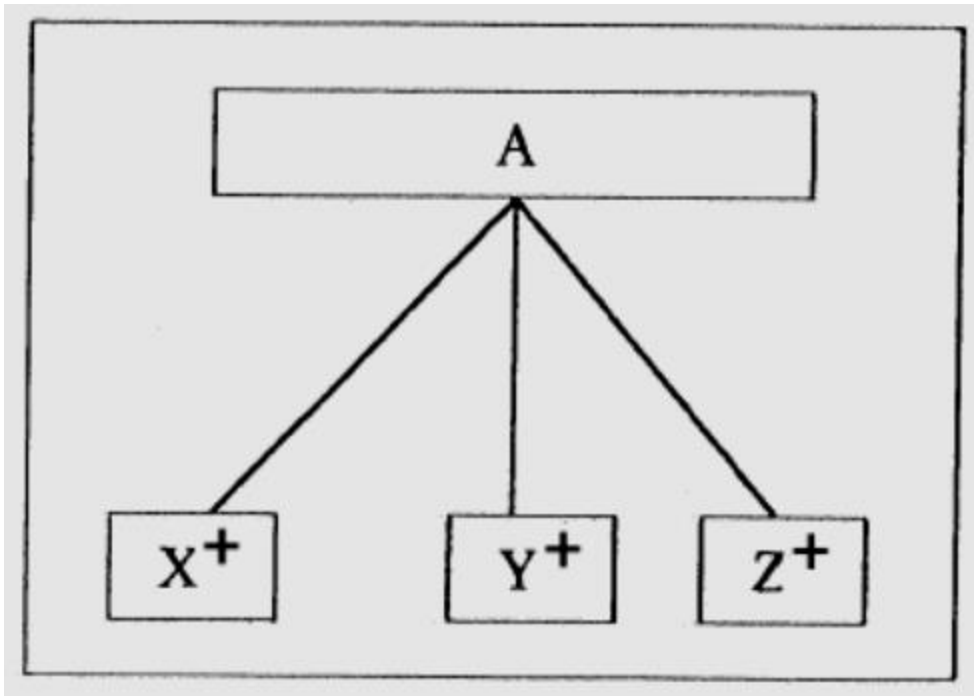
... تابع

- **2- التكرار Iteration:-** حيث يفيد هذا الشكل أن تركيب البيانات A يتكون من تكرار X صفر أو أكثر من المرات. أما معناه في مفهوم البرمجة فهو أن المقطع البرمجي A يتطلب تنفيذ المقطع X صفر مرة أو أكثر.



... تابع

• **3- الاختيار Selection :-** ، حيث يفيد هذا الشكل أن تركيب البيانات A يتكون من عنصر واحد من العناصر X ، Y ، Z . أما معناه في مفهوم البرمجة فهو أن : المقطع البرمجي A يتطلب تنفيذ مقطع واحد من المقاطع X ، Y ، Z . ويمكن توضيح الأشكال الثلاثة في مخطط هرمي واحد



هام جداً: أنظر المثال في صفحة
109 - 110

محاسن ومساوئ طريقة جاكسون

- تكون طريقة جاكسون في التصميم جيدة لتعليم أساليب التعامل مع دورة حياة النظام وطرق تحويل التصميم الى برامج
- طريقة جاكسون لا تكون مناسبة للتعامل مع برامج كبيرة وعملية ولا يمكن استخدامها اذا تطلب الأمر تعديل تلك البرامج أو اذا تغيرت ظروف العمل.

ثانيا : التصميم وفق تراكيب البيانات لتطوير الانظمة

Data Structures Systems Development : DSSA

- تستند على مفاهيم اساسية طورها وورنر عام 1981 بهدف انتاج طريقة شاملة لتحليل البرمجيات وتصميمها.
- تتكون من شقين: الاول يعتني بالمخرجات وواجهة التخاطب والإجراءات البرمجية ويسمى التصميم المنطقي
- الجزء الثاني: يتبع الاول ويسمى التصميم الفيزيائي, ويعتني بجمع وتنسيق جميع اجزاء البرمجية لتحقيق مواصفات الاداء الجيد وإمكانية الصيانة وما شابهها من المواصفات لتحقيق الوضع الافضل للبرمجية.

... تابع

التصميم المنطقي يقسم الى قسمين:-

أ- هيكلية المخرجات المنطقية (Logical Output Structure: LOS)

- في هذا القسم يتم تنظيم وحدات البيانات التي تتضمنها المسألة قيد الدراسة بشكل متسلسل أو هرمي (hierarchically)، ويتم ذلك من خلال الخطوات التالية
- 1- يتم تقييم المسألة أو المعلومات ذات العلاقة بها، ويتم بعد ذلك رصد وتسجيل وحدات البيانات التي لا تقبل التجزئة أكثر.
 - 2- يتم تسجيل عدد مرات تكرار هذه الوحدات.
 - 3- يتم تقييم وحدات البيانات القابلة للتجزئة.
 - 4- يتم استنتاج تمثيل بالأشكال لمرحلة LOS.

... تابع

ب- هيكلية المعالجة المنطقية (Logical Process Structure: LPS)

هذا القسم إجرائي، وتتم فيه عملية معالجة للجزء الناتج من القسم السابق LOS، حيث تتم إضافة تعليمات للمعالجة لكل وحدة بيانات من النوع القابل للتجزئة، ويتم تنفيذ القسم LPS هذا من خلال الخطوات التالية :

1- بعد تحديد وحدات البيانات المعنية، تتم إضافة المحددات البدء (BEGIN) والنهاية (END) لها جميعاً.

2- يتم تعريف جميع تعليمات البدايات والنهايات وعمليات المعالجة اللازمة.

3- يتم تحديد جميع الحسابات أو المعالجة غير العددية.

4- يتم تحديد جميع تعليمات المخرجات وعمليات المعالجة ذات العلاقة.

5- يتم تحديد جميع تعليمات المدخلات وعمليات المعالجة ذات العلاقة.

Design Real Time التصميم الواقعي

- يخضع هذا التصميم لمجموعة قوية من القيوم ويرتبط بشكل قوي بمجال المسألة ويأخذ الاطار الحقيقي والزمني لعمل وتنفيذ البرمجية بعين الاعتبار .
- بسبب خضوع هذا النوع لمجموعة قوية من القيود كان لا بد أن تنتج مثل هذه البرمجيات وفي الذهن انها ستنفذ ضمن بيئة متعددة الجوانب مثل:-
 - معمارية المعدات والبرمجيات معاً –
 - مواصفات نظام التشغيل ومتطلبات المسألة التطبيقية نفسها –
 - لغة البرمجة المختارة ومواصفات التصميم –