Object Oriented – Design

ان الاهداف الرئيسيه التى يحققها التصميم كاننى التوجه تتلخص فى الاتى :

1 – تساعدك على تسريع عملية تطوير وانتاج الانظمه المختلفه

2 – انتاجيه عاليه وانظمه اكثر اعتماديه وكفاءه اعلى

3 – المصادقه الفوريه على تصميم النظام اذا ما تحققت فيه جميع شروط هذه الطريقه

4 – ان عملية تطوير البرمجيات اصبحت اكثر تعقيدا وذات هيكيله ضخمه لذلك يلزم استخدام الطرق الاكثر خبرة واحترافيه لعملية التصميم والبرمجه والتى توفرها لك هذه الطريقه

العناصر الاساسيه التى تبنى عليها النظريه الكائنية التوجه :

Encapsulation

منطق التغليف ومعناه انه يجب تغليف عناصر Module والذى تلكمنا عنه فى الجزء الثانى من دوال او Process – ومتغيرات او قيم تعمل عليها تلك الدوال او Variable – وبعض المتغيرات التى تحمل قيما معقده مثل المصفوفات او التراكيب Matrix or Struct داخل مكون واحد وسيعرف بالفصيل او Class وهذا الفصيل سوف

Information Hiding

عملية اخفاء القيم بحيث لا يستطيع احد التعامل معها خارج نطاق الفصيل الذى يمثلها

وهذه الطريقه تحمى بعض القيم الهامه فى النظام والتى من الخطوره التلاعب فيها

Object Identity

تخليق كائن من فصيل معين وهذا هو الدور المحورى الذى تلعبه الطريقه الكائنية

التوجه وهذا الكائن سوف يمتلك جميع الصفات والخصائص التى يتصف بها هذا

الفصيل بالاضافه انه سوف يحصل على جميع القيم الموجوده به

Classes

ويعرف بالفصيل والذى تحدثنا عنه ويتم صناعة هذا الفصيل اولا ثم اعطائه كافة

البيانات المتعلقه به

Inheritance

الوراثه مبدء مهم من مبادىء التصميم او البرمجه كائنية التوجه وبها يستطيع فصيل

جديد ان يرث جميع ما يتصف به اى فصيل اخر ولكن يجب ان يؤخذ بحذر فقد تجد ان

هناك بعض الفصائل ممنوعه من عملية التوريث بهدف حماية ما بداخلها من معلومات

Polymorphism

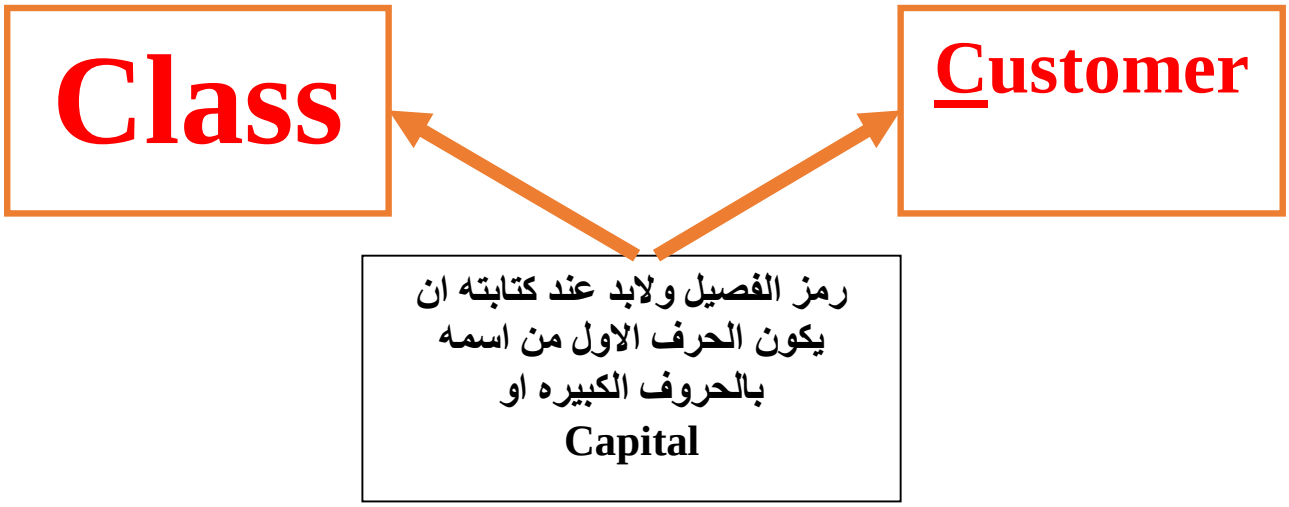
تعطى لك هذه الميزه داخل النظريه كائنية التوجه باعطاء نفس الاسماء لعمليات معينه داخل فصائل مورثه من فصيل واحد وذلك للتشابه بينها فى الوظيفه التى تؤديها واهميتها هى عدم تكرار الكود او اعادة كتابة العمليات حيث انها من صميم فكر التصميم كائنى التوجه

Unified Modeling Language [UML]

وكان لابد من توظيف لغه عالميه تستخدم فى تصميم الانظمه المختلفه وتطبق النظريه الكائنية التوجه وهى لغه رسوميه تمثل تركيب النظام ككل وتصرف كل جزء فيه

UML Symbols

الرموز المستخدمه داخل هذه اللغه

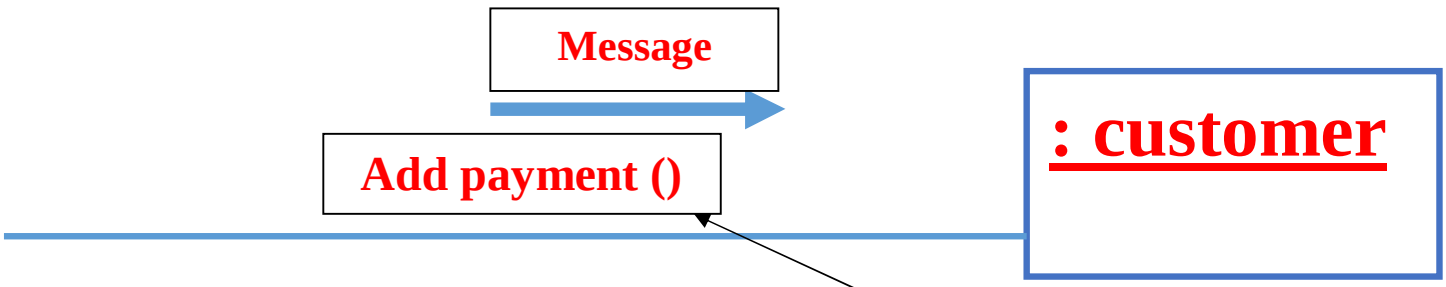


الكائن المخلق من فصيل معين وهو الفصيل السابق ويحمل اسم الفصيل دلاله على انه تابع له ويكتب بحروف صغيره ونقتطان وتحته خط لكن هذا الكائن لم يعطى له اسم محدد بعد

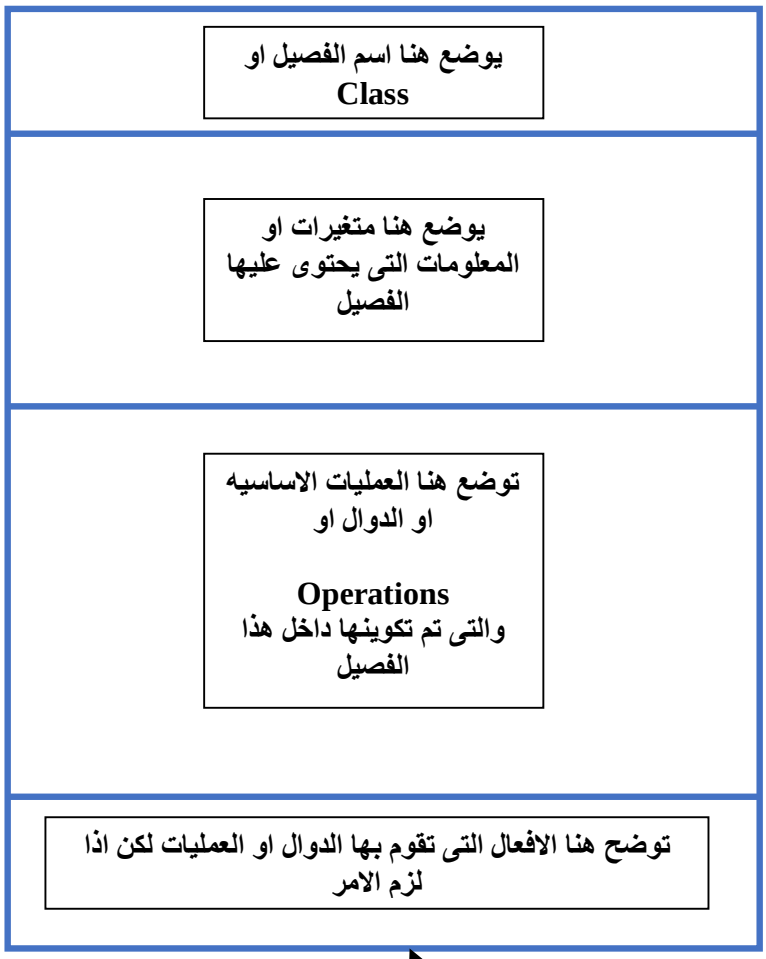
: customer

الكائن السابق لكن بعد اعطائه اسم محدد Cust1
والذى يعطى له بالطريقه التى تراها فى الرمز

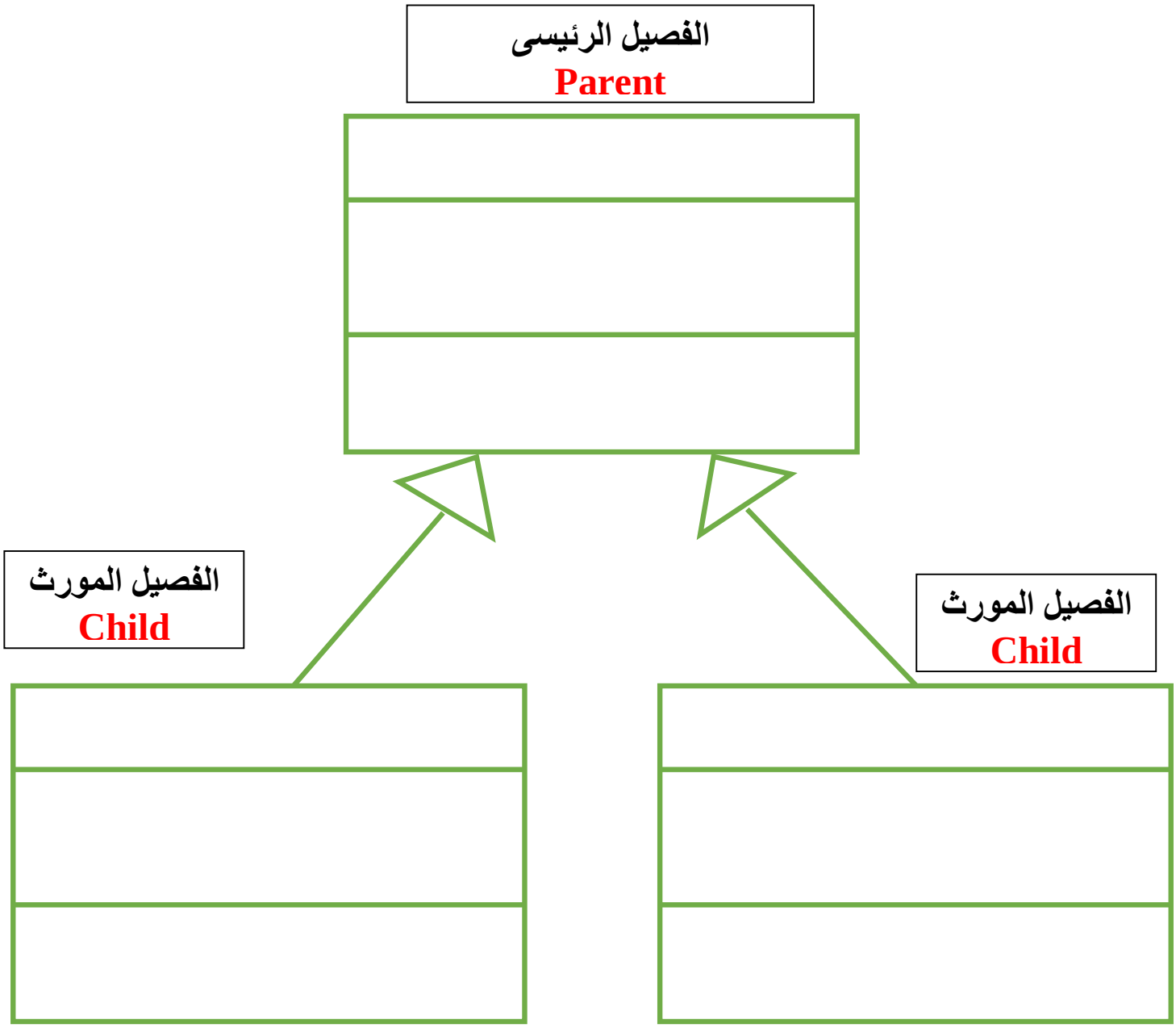
Cust1: customer



هذا الرمز يشير الى كائن تربطه علاقه بكائن اخر والخط المرسوم يوضح هذه العلاقه حيث يكتب فوقه الداله التى ينفذها كائن ما لكائن اخر يقوم بارسالها اليه والسهم المرسوم فوقه يمثل الرساله التى يبعث بها اليه لكى ينفذ له هذا الامر ولابد ان تكون رأس السهم سميكه وملونه ولا تكون فارغه



هذا الرمز يوضح الشكل العام للفصيل لكن هذه المره مع سرد لجميع الاعضاء التى يحتويها هذا الفصيل
من قيم او متغيرات ومن عمليات او دوال تؤدى وظائف معينه



تمثيل الوراثه بلغة
UML

Public and Private Members

ان الاساس فى نظرية كائنية التوجه هى الفصيل مجموعه من الفصائل ويتم تغليف عناصر عديده بداخلها مث المتغيرات – الدوال – الخصائص لكن الذى يتفاعل مع هذه العناصر والتى يحتويها الفصيل هو ليس الفصيل وانما الكائن المخلق منه والذى يحمل مفاتيح الوصول الى جميع عناصر هذا الفصيل

ومعنى ذلك ان لا قيمة لاى فصيل بدون الكائن لكن يجب ان نعرف ان لكل عنص من عناصر الفصيل خصوصيه فى التعامل فقد تستطيع ان تتعامل مع عناصر وعناصر اخرى لا تستطيع التعامل معها والذى يسمح بذلك او لا يسمح هى تاشيرة الاتصال التى ترافق العناصر والتى تعرف بطريقة الاتصال او

Access Modifiers

وهذه الطرق هى التى تستطيع بها التعامل مع عناصر الفصيل من الخارج او لا تستطيع ونوضحها كالتالى :

Public

تستطيع التعامل مع العنصر من خارج الفصيل

Private

لا تستطيع التعامل مع العنصر من خارج الفصيل نهائيا

Protected

تستطيع التعامل مع العنصر من الفصيل ذاته او اى فصيل مشتق منه بمعنى مورث منه

Internal

تستطيع التعامل مع العنصر من خلال المشروع ككل

Protected Internal

تستطيع التعامل مع العنصر من خلال المشروع او من مشروع مشتق منه

Static

يعتبر هذا النوع مهم لانه يمكنك من التعامل معه بواسطة الفصيل مباشرتا دون الحاجه

للتعامل مع الكائن او دون الاتصال به عن طريق الكائن

جميع الانواع السابقه تستخدم على نطاق واسع داخل لغة السى شارب

لكن لا نسنطيع ان نتجاهل نوعين هامين وتستخدم بشكل اساسى مع لغة السى بلس بلس

وهما :

Global

وبها تستطيع ان تتعامل مع العنصر على مستوى المجال او خارجه او على مستوى الفصيل

Local

تستطيع فقط التعامل مع العنصر من خلال المجال الذى يحتويه فقط

****

نظره اشمل على مفهوم الوراثه :

ان مفهوم الاشتقاق او التوريث هو ان يرث الفصيل الابن جميع خصائص الفصيل الاب

بمعنى ان يأخذ منه جميع الصفات – الخصائص – العمليات التى يحتويها ولا مانع من اضافة

ميزات اخرى للفصيل الابن او التعديل على ميزات كان يملكها الفصيل الاب ولاحظ ان اى

كائن مخلق من الفصيل الابن يستطيع استدعاء اى عنصر من عناصر الفصيل الاب لكن

لا يحدث العكس وعلى هذا نستطيع ان نعرف المصطلحات الاتيه

ان الفصيل الاب هو حاله عامه من الفصيل الابن بينما الفصيل الابن هو حاله خاصه من

الفصيل الاب

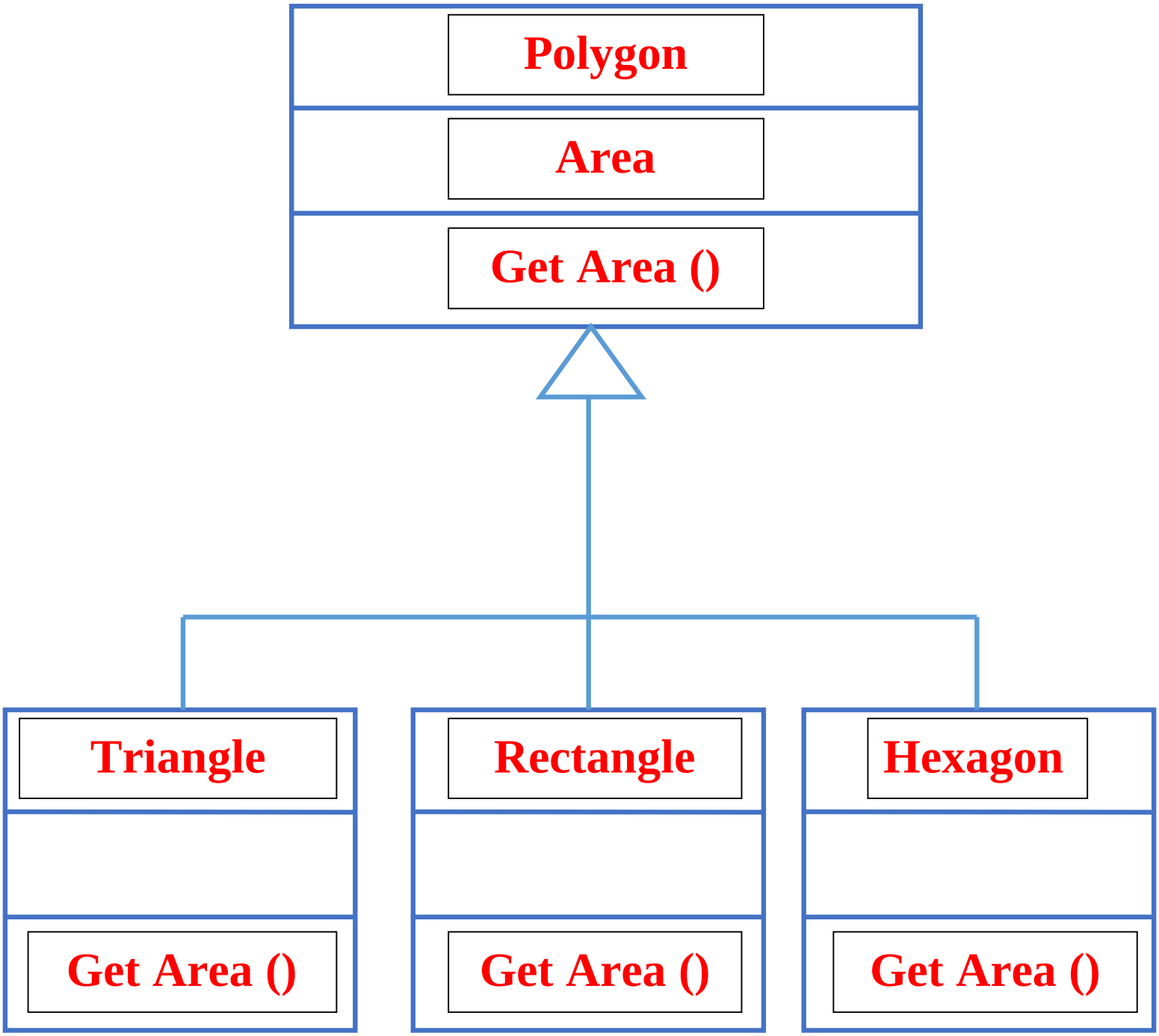
Parent is General State from Child but Child Is Special State from Parent

وتعرف هذه الحاله بمصطلح داخل نظرية الكائنية التوجه

Generalization – Specialization Diagram or (Gen – Spec) Diagram

نظره اشمل على مفهوم التعدديه :

-Polymorphism-



فى الرسمه السابقه اذا كنت تستخدم اسلوب الدوال فى حساب مساحة الاشكال فأن بعض الاكواد لابد ان تتكرر وهذا عكس مفهوم النظريه كائنية التوجه

لكنك اذا اردت ان تطبق النظريه فعليك بناء فصيل اساسى ويسمى المساحات مثلا وتوضع فيه داله رئيسيه لحساب المساحه ويتم اشتقاق اكثر من فصيل كل منهم يسمى بأسم الشكل الذى تريد حساب مساحته ويتم عمل تعديل على الداله حسب نوع الشكل ولاحظ ان جميع فصائل الاشكال هى مورثه من نفس الفصيل وهو فصيل المساحه

لقد تم استخدام جميع اساليب النظريه كائنية التوجه من

Encapsulation – Inheritance – Retention of Data

مما يدعم فكرة

Polymorphism

لكن على اى حال لا يكتمل المفهوم دون ان تعرف انه يدعم عملية الاشتقاق الهرمى او المتدرج وعملية

Interface – Multi Level Hierarchies

ان فائدة لغة التمثيل العالميه التى استخدمت فى التصميم بنظرية الكائنية التوجه

1 – تمثيل جزء من الكود وذلك لفهمه جيدا وارتباطه مع بعضه

2 – تستطيع الرجوع من عملية التكويد الى لغة التصميم مره اخرى مما يعرف بعملية

الهندسه العكسيه

3 – تساعدك على انتاج اصدارات لعملية التصميم ويصادق على الاصدار الاقوى فيهم

بعض الاساليب المتقدمه والاكثر فائده داخل التصميم بأستخدام لغة التصميم :

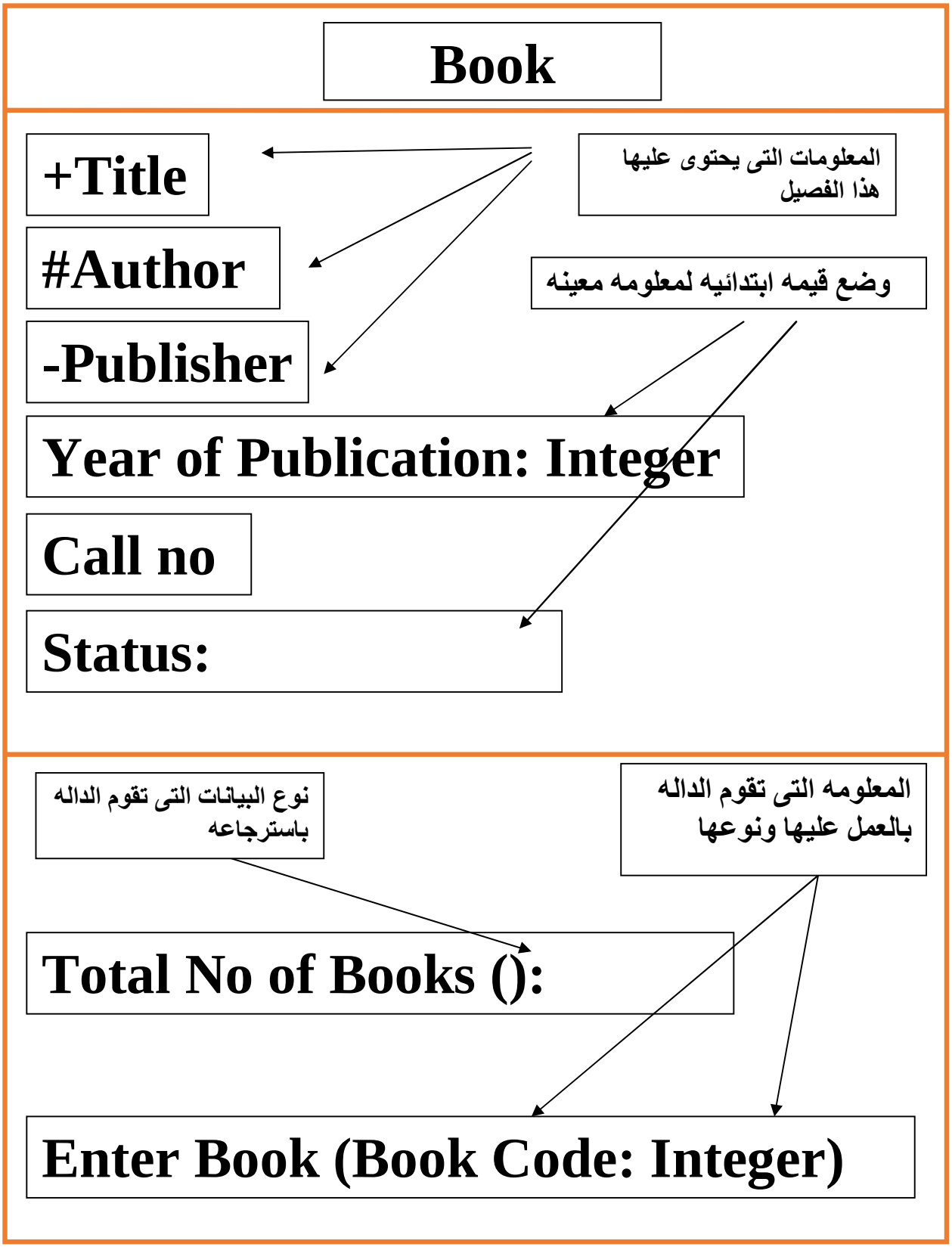
1 – تستطيع ان تضع نوع للبيانات للمعلومات التى يحتويها الفصيل

2 – تستطيع ان تضع قيمه ما ونوعها والتى سوف ترجع بها الدوال الموجوده داخل الفصيل

فيما يعرف ب

Data Type – Return Value

ولنرى ذلك فى مثال للتصميم التالى :


Book
+Title
#Author
-Publisher
Year of Publication: Integer
Call no
Status:
المعلومات التى يحتوى عليها هذا الفصيل
وضع قيمه ابتدائيه لمعلومه معينه
Total No of Books ():
Enter Book (Book Code: Integer)
نوع البيانات التى تقوم الداله باسترجاعه
المعلومه التى تقوم الداله بالعمل عليها ونوعها

نود التعليق على الرسم السابق بوجود بعض الرموز والتى تحدثنا عن مفهومها داخل النظريه الكائنية التوجه لكن بقى ان نربط تلك الرموز بالاشياء التى تدل عليها :

(+Public, -Private, #Protected)

- دراسه حول العلاقات بين Classes وبعضها البعض :

والعلاقات يوجد منها انواع محدده :

1 – Dependency او الاعتماديه

2 – Generalization الحاله العامه

3 – Association الانداد

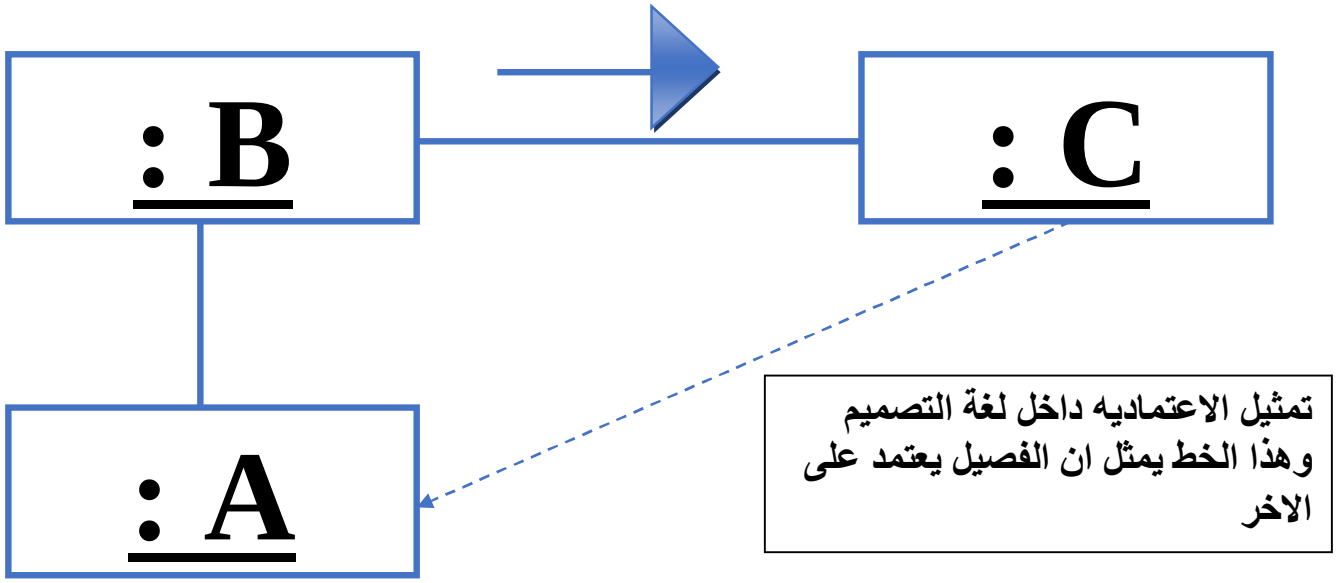
4 – Realization

ناتى لدراسة النوع الاول وهو الاعتماديه :

قد تجد بعض Classes او الفصائل تعتمد على بعضها فى اداء المهام الخاصه بها

بمعنى اخر اذا كان هناك فصيل يعتمد على فصيل اخر فان اى تغيير فى الفصيل الحر يشعر به

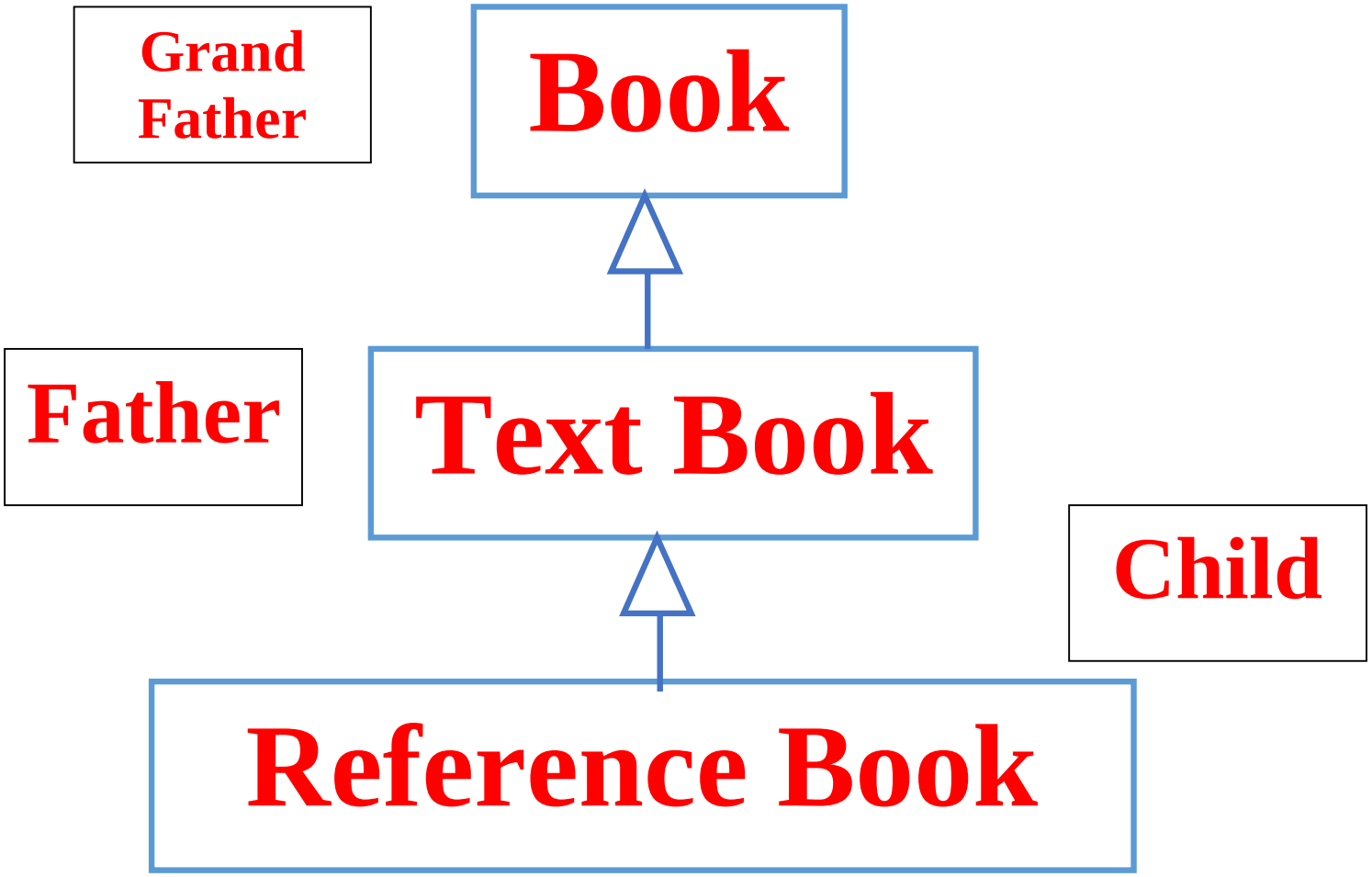
الفصيل الذى يعتمد عليه لكن لا يحدث العكس ولنرى ذلك فى الرسم التالى وكيفية تمثيله



فى الرسم السابق تجد ان Class C يعتمد على Class A وهذا معناه ان اى تغيير يحدث فى الفصيل A سوف يشعر به الفصيل C لكن لا يحدث العكس

النوع الثانى وهو الحاله العامه او Generalization :

وقلنا سابقا انها عباره عن علاقة وراثه متدرجه او تمثل الشكل الهرمى لسلسة التوريث بين الفصائل وبعضها البعض بمعن ان الابن يرث الاب ثم أبن ثانى يرث الفصيل المورث منه وهكذا تماما كعلاقة الاباء بالابناء والاحفاد بالاجداد



كما تستطيع بهذه الميزه عمل شجره كامله تكون اشبه بشجره العائله لكن المهم ان تعرف ان الابن يرث جميع صفات وخصائص الاب بل انك تستطيع ان تضيف عليه اشياء كما يشعر بأى تغيير يحدث فى الفصيل الاب لكن لا يحدث العكس وبالتالى فان الفصيل الاب هو حاله عامه من الفصيل الابن لكن الابن هو حاله خاصه

دراسة النوع الثالث وهو علاقة الانداد او Association :

ان معنى علاقة الانداد انه لا يوجد وراثه على الاطلاق فهى علاقة الند للند ولها بعض الخصائص :

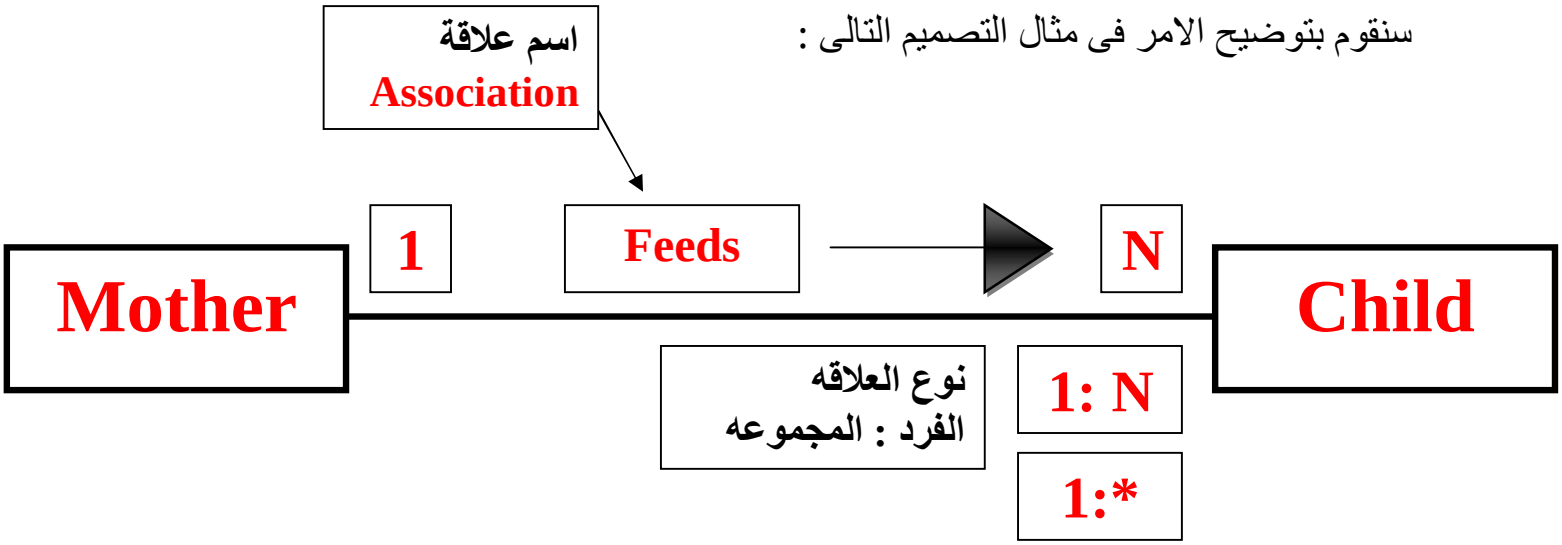
1 – الاسم او Name

2 – القواعد او Role

3 – التعدديه او Multiplicity

4 – التجميع او Aggregation

سنقوم بتوضيح الامر فى مثال التصميم التالى :



ان العلاقه فى الرسمه السابقه ليست علاقة وراثه بل علاقة انداد حيث ان الام تغذى الاطفال وهى علاقة الفرد للمجموعه حيث ان الام لديها اكثر من طفل لكن الطفل ليس له الا ام واحده

ملحوظه : من الطبيعى هنا ان يرث الاطفال صفات الام لكن حسبت فى هذا المثال انه ليس هناك وراثه بل علاقة انداد او ارتباط من نوع معين وهذا ما يعرف بالتعدديه

Object Interaction

تفاعلات الكائنات مع بعضها : اذا كان هناك علاقة ترابط بين كائنين فى النظام فلابد من وجود رابط بينهما وهو يمثل بالخط الواصل بين هذين الكائنين

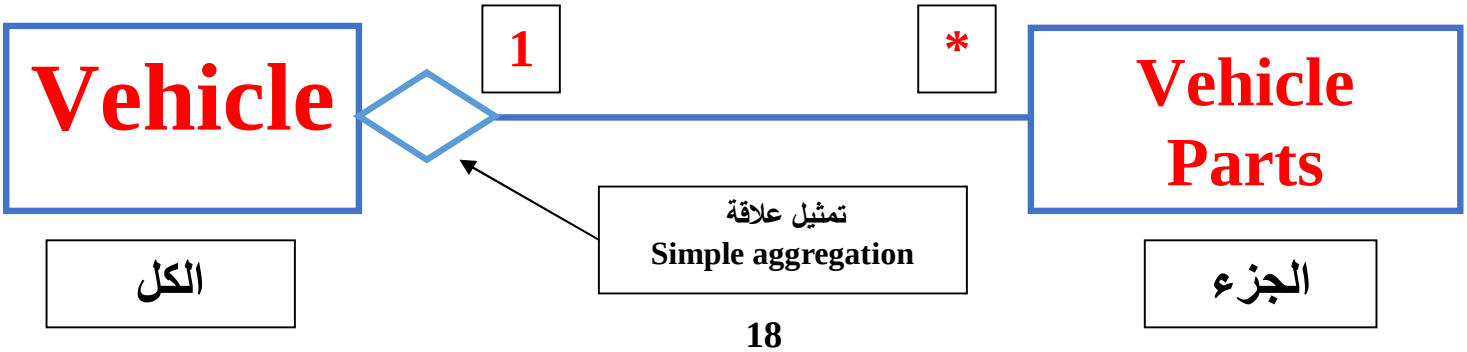
Events

الاحداث لحظة ارسال رساله من كائن لكائن اخر ويبنى عليها فعل او Action ينفذ على اساسه جزء من العمليه او Executable

Aggregation

ولها نوعان وهما Simple Aggregation – Composite Aggregation

Simple Aggregation



معنى Simple Aggregation

انه اذا انتهى او مات الكل فان الجزء باقى ولا ينتهى بمعنى انه اذا تم ازالة السياره فأن قطع

الغيار متوفره كجزء منفرد

Composite Aggregation

اذا مات الاصل او الكل مات او انتهى الفرع او الجزء وتمثل كالتالى

بالطبع هنا اذا ما انهار المبنى كاملا لا اظن انك سوف تجد فيه غرفه سليمه

الى هنا انتهى كتاب هندسة البرمجيات وقد تم تلخيصه فى ثلاثة اجزاء موجوده كاملا على

موقع الحاسب العربى واتعمد فى اسلوبى البساطه والمرونه وجعل المعلومه فى ابسط شكل

ممكن

Sofyany
Memorycode_84@yahoo.com