

UML

*

لغة النمذجة الموحدة (Unified Modelling Language)، أو (UML) ، هي لغة نمذجة رسومية تقدم لنا صيغة لوصف العناصر الرئيسية للنظم البرمجية. (هذه العناصر تسمى artifacts مشغولات في UML). في هذه الفصول سوف نستكشف النواحي الرئيسية في UML، و نصف كيف يمكن تطبيق UML في مشروعات تطوير البرمجيات.

هي تقدم وسيلة رموزية مبسطة للتعبير عن مختلف نماذج العمل البرمجي يسهل بواسطتها على ذوي العلاقة - من محللين و مصممين و مبرمجين بل و حتى المستخدمين - التخاطب فيما بينهم و تمرير المعلومات في صيغة نمطية موحدة و موجزة، تغنيهم عن الوصف اللغوي المعتاد. إنها مثل مخططات البناء التي يتبادلها المساحون والمعماريون ومهندسو التشييد، أو مخططات الدوائر الكهربائية و الالكترونية التي يمكن لأي كان في هذا المجال أن يفهمها و يتعامل معها.

نظرة تاريخية

لمدة، كان مهندسو البرمجيات يفتقرون لمثل هذه الرموز. بين عامي 1989 و 1994، و هي الفترة التي يشار إليها بـ "حروب المناهج"، كان يوجد ما يزيد عن 50 لغة نمذجة برمجية قيد الاستعمال - كل منها تملك رموزها الخاصة! كل لغة تحتوي على قواعد تميزها، بينما في نفس الوقت، كل لغة لديها عناصر تتشابه مع تلك التي في اللغات الأخرى.

و لمزيد من الفوضى، لا توجد لغة متكاملة، بحيث نادرا ما يجد القائمون على البرمجيات ما يرضي كامل حاجتهم في لغة واحدة!

في منتصف التسعينيات، برزت ثلاث منهجيات لكي تكون الأقوى. بدأت هذه المنهجيات الثلاث في التقارب، كل واحدة منها تحوي على عناصر من الآخرين. كل منهجية تملك نقاط قوة خاصة بها:

- **بوك Booch** كانت ممتازة فيما يخص التصميم و التنفيذ. لقد عمل "قرادي بوك" Grady Booch بكثافة على لغة آدا Ada، و كان له دور رئيسي في تطوير تقنيات المنحى الكائني (object oriented) للغة. وبالرغم من قوة منهجية بوك إلا أن الرموز فيها لم تأخذ القبول الحسن (الكثير من الأشكال السحابية تغزو نماذجه - ليست بالجميلة!)
- **OMT** (تقنية النمذجة الكائنية Object Modelling Technique) كانت الأفضل في التحليل و في أنظمة المعلومات ذات البيانات الكثيفة.
- **OOSE** (Object Oriented Software Engineering) هندسة البرمجيات كائنية المنحى) و تتميز بنموذج يسمى وقائع الاستخدام (Use Cases). تعد وقائع الاستخدام أسلوب قوي من أجل فهم سلوك كامل النظام (و هو المجال الذي كان فيه المنحى الكائني ضعيفا).

في عام 1994، قام جيم رامبخ Jim Rumbaugh، مؤسس OMT، بمفاجأة عالم البرمجيات حين ترك العمل بشركة جنرال الكتريك General Electric و انضم الى قرادي بوك للعمل في شركة راشيونال (Rational Corp). الغرض من المشاركة كانت من أجل دمج أفكارهما و صبّها في منهجية موحدة (و كان بالطبع عنوان العمل لهذه المنهجية هي "المنهجية الموحدة" Unified Method). مع عام 1995، انضم أيضا مبدع OOSE ايفار جاكوبسون Ivar Jacobson، إلى راشيونال، و تم ضم أفكاره (خاصة مفهوم "وقائع الاستخدام" Use Cases) في المنهجية الموحدة - الآن تدعى لغة النمذجة الموحدة (Unified Modelling Language). * وعُرف الفريق الذي يتكون من رامبخ و بوك و جاكوبسون بـ "الأصدقاء الثلاثة" (Three Amigos).

بغض النظر عن بعض الحروب و المشاحنات البسيطة، بدأت المنهجية الجديدة تجد استحبابا لدى أوساط صناعة البرمجيات، فتم تكوين لجنة مشتركة consortium خاصة بـ UML، شاركت فيها عدد من المؤسسات ثقيلة الوزن مثل هيولت-باكارد (Hewlett-Packard) و ميكروسوفت (Microsoft) و أوراكل (Oracle).

إشكالية عام 2000 (ثغرة الألفية) ذات التكلفة العالية
فحتى لو أن حلها يفترض فيه أن يكون بسيطا (تغيير كل حقل سنة من خانتين إلى أربع)
فإن التداعيات المحتملة لهذا التغيير البسيط يجب التحقق منها و فحصها بدقة.

UML diagrams

مخططات لغة النمذجة الموحده

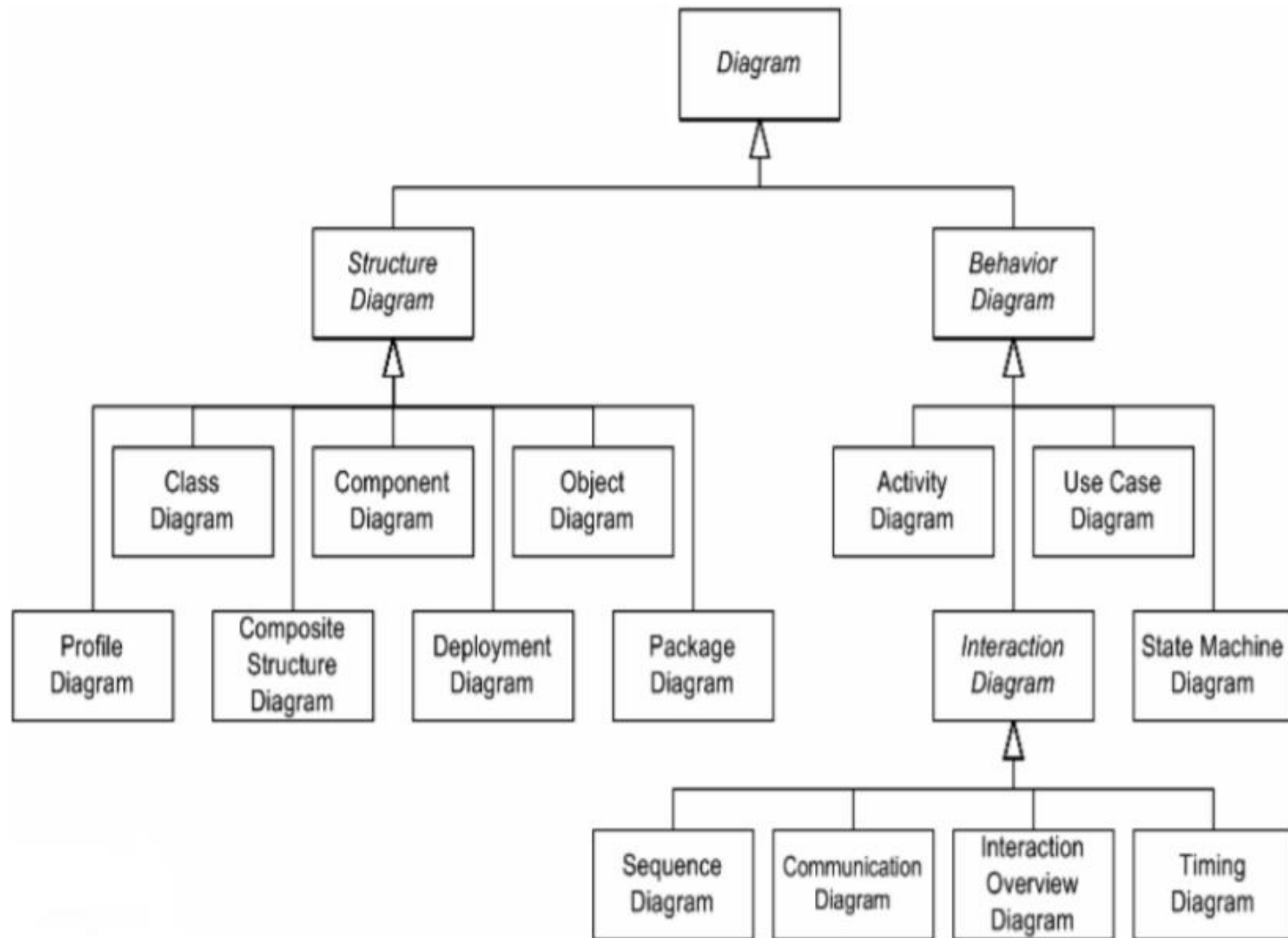
تنقسم إلى قسمين:

مخططات هيكلية/ساكنه Static/structure:

تركز على البنية الثابتة للنظام باستخدام (الكائنات، السمات، العمليات، العلاقات)

مخططات سلوكيه/تفاعليه Dynamic/behavior:

يركز على السلوك الديناميكي للنظام أي تغيراته مع الزمن



في هذا الكورس سوف نركز علي عدد من الامثله في كل قسم
من اقسام لغة النمذجه الموحده:

اولا المخططات الهيكلية:

مخطط الصنف

ثانيا المخططات السلوكيه:

مخطط حالات الاستخدام

مخطط التتابع

مخطط الصنف Class diagram

*هو احد امثلة المخططات الهيكلية.

البيانات data الخاصة بالكائن object تسمى عادة بخصائص أو سمات Attributes الكائن. و تسمى التصرفات المختلفة التي يقوم بها الكائن طرق أو نهجيات Methods الكائن. النهجيات هي مرادف لما يعرف في لغات البرمجة بالوظائف functions أو الإجراءاتيات procedures .

المصطلح الآخر المشهور في هذا السياق هو Class الصنفية. الصنفية أو الصنف هي ببساطة أرضية (template) يقوم عليها الكائن. يتم في الصنفية وصف السمات attributes و النهجيات methods التي ستكون حاضرة لكل تمثيلات أو تجسّدات الصنفية. في منظومة المعهد

.Student

يمكن وضع الأشياء ضمن أصناف Classes ، مهمة Class diagram توضيح هذه الأصناف والعلاقات associations فيما بينها

لكل صنف Class :

- اسم class name
- مميزات attributes
- عمليات operations وأحيانا نسميها Methods

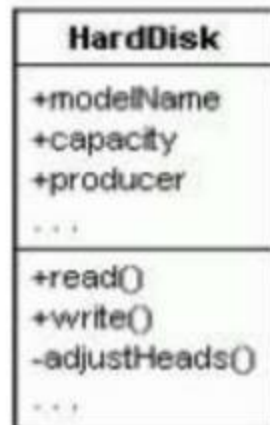
وبالتالي يمكن تمثيل Class بأحد الطرق التالية:

Class Name

Class Name
Att1: type 1
Att 2: type 2 = "default value"
...
...

Class Name
Att1: type 1
Att2: type 2 = "default value"
...
...
Operation1 ()
Operation2 (list of parameters)
Operatipn3: returned value type

*مثال علي مخطط الصنف

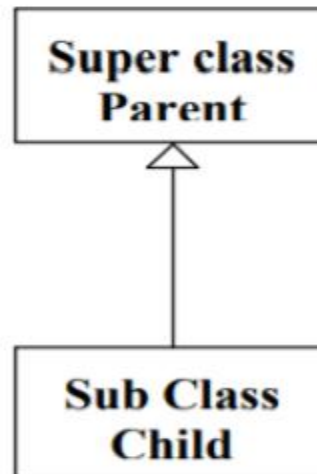


*علاقة الوراثة

الوراثة Inheritance & Generalization

توجد بعض المميزات المشتركة بين الأصناف والتي يمكن نقلها بين الأصناف Classes وبالتالي يفضل في هذه الحالة أن ندخل مفهوم الوراثة.

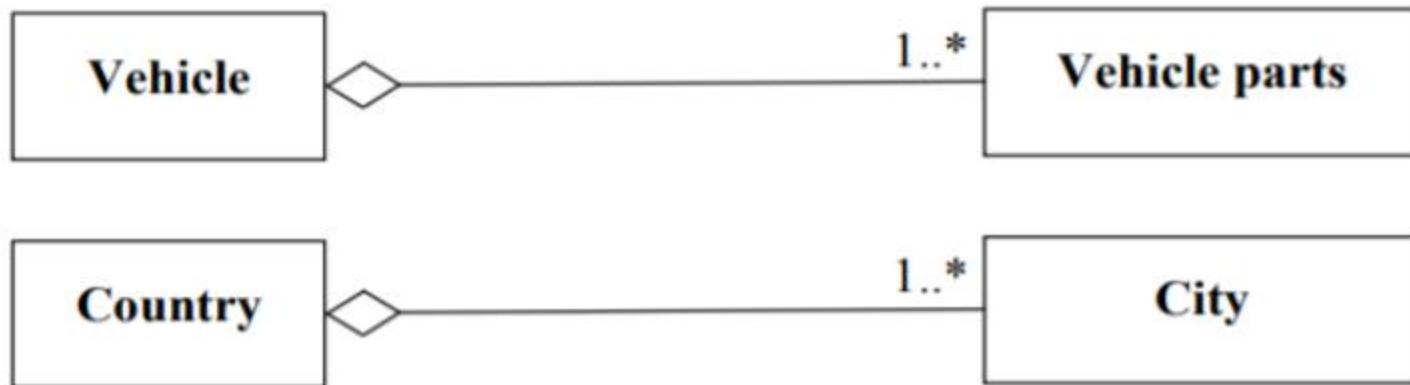
يمكن لأحد الأصناف تسميته Child class (subclass) أن يرث attributes and operations من صنف آخر نسميه Parent class (super class) حيث يمكن أن ينوب الأب عن ابنه ولكن العكس غير صحيح. مثال: موظفين في مؤسسة ما (مدير، مهندس، سكرتيرة، رئيس قسم، عامل بوفية، عامل نظافة، ...)



*علاقة التجميع

علاقة Aggregations

هي علاقة بين الكل والجزء ويتم تمثيلها كما يلي:



*مثال

