

## المحاضرة الأولى

### مقدمة في البرمجة الكائنية

#### Introduction to Object Oriented Programming

#### معنى البرمجة الكائنية OOP :-

عبارة برمجة أسلوب جديد في البرمجة من حيث وحدة بناء البرنامج ومن حيث الخصائص الجديدة التي يسمح بها هذا الأسلوب حيث يعتبر وحدة بناء البرنامج هو الصنف ( Class ) الذي يتكون من البيانات ومعها الدوال (العمليات) التي تعمل على هذه البيانات .

#### **أو بصورة مختصرة :-**

يمكن أن نقول هي عبارة عن طريقه جديدة مستخدمة في البرمجة تمكن من تحليل وتصميم التطبيقات بشكل كائنات تحوي البيانات وتعرف عليها مجموعة من العمليات .

#### **ولها عدة مسميات منها :-**

1. البرمجة غرضيه التوجه

2. البرمجة الموجهة نحو الأشياء

3. البرمجة الكائنية الموجهة

#### المقارنة بين أسلوب البرمجة الهيكلية والكائنية :

كان الأسلوب الشائع في تصميم البرامج القوية هو أسلوب البرمجة الهيكلية الذي يعتمد أساسا على بناء هيكل للبيانات دون الاهتمام بالعمليات التي يمكن تطبيقها على البيانات ، بعد ذلك بدأ المبرمجون في الكشف عن أسلوب جديد للبرمجة يجمع البيانات والعمليات في بنية واحدة وذلك للعلاقة الوثيقة بين البيانات والعمليات وكان هذا الأسلوب هو أسلوب البرمجة الكائنية أو الكائنية .

#### **ويختلف البرنامج المكتوب بأسلوب OOP عن البرنامج المكتوب بالأسلوب الهيكلية فيما يلي :**

#### **1. وحدة بناء البرنامج :**

كان البرنامج المكتوب بالأسلوب الهيكلية عبارة عن داله رئيسيه ومجموعة دوال فرعيه ، ويتم إستدعاء الدوال الفرعية من داخل الدالة الرئيسية حسب تسلسل البرنامج ، كما في لغة C الهيكلية أما البرنامج المكتوب بأسلوب OOP فقد أصبحت وحدة البرنامج فيه هي الصنف Class التي تتكون من البيانات والدوال التي تعمل على هذه البيانات ويتم إستدعاء الدوال كعناصر للأصناف .

#### **2. النظره إلى البيانات :**

في البرنامج الهيكلي كان مجهود المبرمج ينصب على كود البرنامج أي سطور البرنامج التي تتولى سير العمليات بينما كانت نظره إلى البيانات نظره ثانوية ولكن أسلوب برمجة البرمجة الكائنية يعتبر البيانات جزء مهم من البرنامج . وبالتالي أصبحت لدينا مكتبة ليس فقط من الدوال بل مكتبة من الفصائل التي تحتوي على البيانات والدوال التي تتعامل معها ووفرت هذه المكتبة الكثير من المجهود الذي يبذله المبرمج حيث كان يقوم بتعريف البيانات ثم الدوال التي تتعامل معها كل مره . وتظهر مكتبة الأصناف في برمجة النوافذ حيث يوجد صنف للنافذة الرئيسية للبرنامج به جميع البيانات التي تخص النافذة وكذلك الدوال التي تتعامل مع هذه النافذة ويوجد صنف لمربع الحوار وبه بيانات مربع الحوار وكذلك الدوال التي تتعامل معه فأصبحت وحدة بناء البرنامج كبيرة وليس على المبرمج كتابة كل شئ في البداية .

## **بعض المفاهيم الأساسية في البرمجة الكائنية OOP :**

### **1. الكائن : object**

هو عبارة عن وحدة تحوي مجموعة من البيانات تسمى خصائص أو صفات ومعرفه عليها مجموعة عمليات (دوال) مثال:- طالب ، قلم ، حاسب .

### **2. الصنف : Class**

هو عبارة عن نوع يحوي مجموعة من الكائنات التي تشترك في الخصائص والعمليات. مثال :- صنف الحاسبات ، صنف الطلاب .

والصنف يمثل المواصفات العامة للكائنات التي تنتمي لهذا الصنف ، بينما الكائنات تمثل شئ قائم بذاته أو شئ له ذاتية تنتمي لذلك الصنف .

## **الفرق بين الصنف والكائن:**

كل ما في الوجود هو كائن فأنا وأنت وهذه الورقة والقلم كلها كائنات **Objects** ولكل منها خصائص محددة ويستطيع القيام بعمليات محددة. أما الصنف فهو مجموعة من الكائنات المتشابهة فالرجال صنف وزيد كائن منه والنساء صنف وأمل كائن منه. وكمثال آخر الصنف البرمجي: "بطاقة دوام" الذي يحوي الطرق المطلوبة لحساب الأجر و عدد ساعات الدوام أما الكائن "بطاقة دوام محمد" والكائن "بطاقة دوام حسن" فإنهما يستفيدان من الإجراءات الموجودة في الصنف "بطاقة دوام" لحساب أجر محمد أو أجر حسن

## مزايا البرمجة الكائنية Features of OOP

### 1. التجريد Abstraction

وهو عملية تحديد كل الخصائص والعمليات التي تنتمي لصنف أو كائن معين

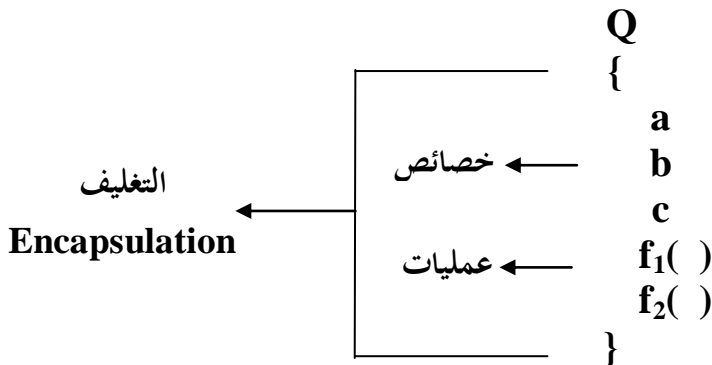
وهي نوعان :-

▪ تجريد البيانات Data Abstraction

▪ تجريد العمليات Methods Abstraction

### 2. الكبسلة (التغليف) Encapsulation

هي عملية تجميع كل الخصائص والعمليات في وحدة واحدة تسمى الصنف (داخل غلاف واحد) ولا يمكن الوصول إلى هذه الخصائص والعمليات إلا عن طريق الكائن



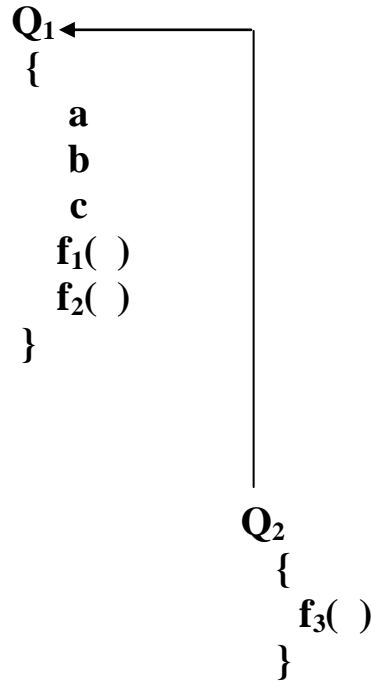
### 3. إخفاء البيانات Data Hiding

وهي ميزة ناتجة عن كبسلة البيانات وتعني إخفاء بعض البيانات وإضافة مستوى حماية معين عليها حتى تمنع الوصول الخطأ إليها

### 4. الوراثة Inheritance

وهي أن يرث صنف معين الخصائص والعمليات المعرفة في صنف آخر مما يساعد على إعادة استخدام الأصناف التي تم أنشاؤها من قبل

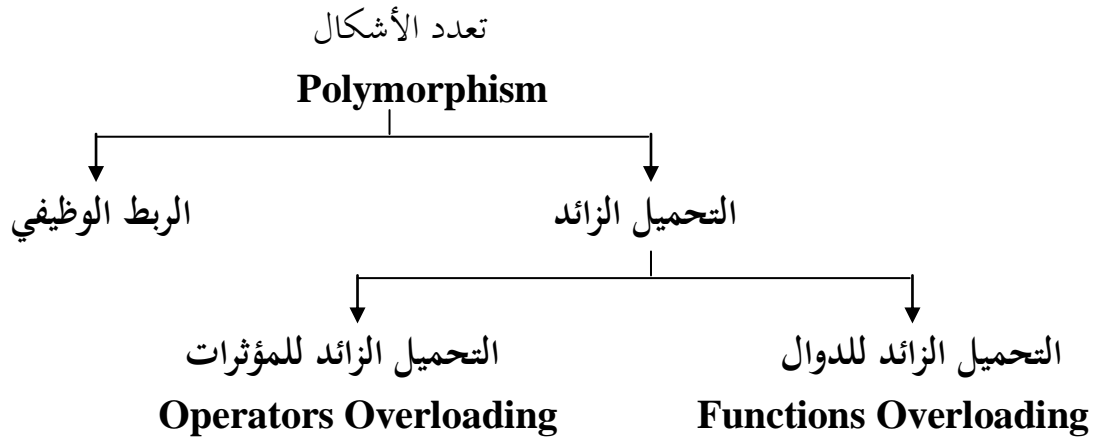
مثلاً إذا كان لدينا صنفان Q1 و Q2



فأن Q2 يرث عناصر Q1 بالإضافة إلى f3( )

## 5. تعدد الأشكال Polymorphism

وهي ميزة تمكن من إنشاء كائنات لها القدرة على القيام بأكثر من وظيفة أو إنشاء دوال لها القدرة على القيام بأكثر من وظيفة



## مستويات الحماية (محددات الوصول داخل الأصناف)

هي عملية تحديد مدى التعامل مع الأعضاء (ولتكن البيانات) هل هذه البيانات خاصة بهذا الصنف أم لأي داله في أي صنف أن تتعامل مع هذه البيانات؟

وهي ثلاثة مستويات :

### 1. مستوى الحماية الخاص Private

يستخدم مستوى الحماية الخاص لتعريف الأعضاء الذي لا يمكن الوصول إليها من خارج الصنف .

### 2. مستوى الحماية المحمي Protected

يشبه مستوى الحماية الخاص غير أنه يمكن توريثه إلى صنف آخر .

### 3. مستوى الحماية العام Public

يستخدم لتعريف الأعضاء التي يمكن الوصول إليها من خارج الصنف ويمكن توريثها إلى صنف آخر

## فائدة تحديد درجة الحماية :

يتم عادة تحديد درجة حماية من نوع Public للأعضاء الدوال ودرجة حماية من نوع private للأعضاء المتغيرات

هنا يتبادر الى الذهن السؤال لماذا ذلك ؟

لأنه من مبادئ برمجة الأهداف الموجهة مبدأ إخفاء البيانات Data Hiding فلا يمكن التعامل مع البيانات مباشرة بل عن طريق الدوال المصممة لذلك .

ومن فوائد إخفاء البيانات تقليل الخطأ في استخدام البيانات حيث تمثل الدوال واجهة الاستخدام الوحيدة مع البيانات .