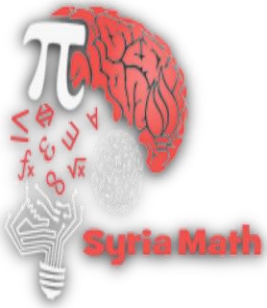

◄ دكتور المادة: سمير جعفر

◄المحاضرة التاسعة عنوان المحاضرة : الصفوف

المستوى العلمي : أهلاً بكم أصدقائي في المحاضرة التاسعة من مقررنا البرمجة والخوارزميات سنتناول في هذه المحاضرة مفهوم الهادم ومثال شامل عن الصف واستخدامه والتعامل مع العمليات عليه.

تعرفنا سابقا على الباني :وهو احدى الطرق موجود داخل الصف له نفس اسم صف وليس له نوع ارجاع ولا حتى (void) يقوم ببناء أغراض (objects) أي إعطاء قيم ابتدائية للمعطيات الاعطاء وحجز مكان في الذاكرة للغرض.

لا يمكن تعريف متحولات من الصف الا بوجود هذا الباني لذلك ان لم نكتب ضمن كود البرنامج باني تقوم الدالة بتوليد باني هو الباني الافتراضي .

سؤال : ما هو المصطلح المستخدم للتعبير عن دالة معرفة داخل صف؟

A- عضو بياني , B-طريقة, C-دالة صف, D-دالة عضو الجواب الصح هو طريقة واي إجابة غير الطريقة هي خاطئة.

مثلما يوجد باني يقوم ببناء الأغراض هناك هادم يهدم الأغراض من صف معين :

الهادم (deconstructer):

هو طريقة (دالة) خاصة لها اسم الصف نفسه مسبوقا بإشارة (~) ليس له وسطاء ويقوم بتحرير الذاكرة المخصصة للغرض المحدد(object)

اذا اردنا انشاء هادم للصف complex نكتب ~complex( ) { } كدالة (طريقة)داخل الصف .

بشكل افتراضي عند انتهاء البرنامج يتم استدعاء جميع الهوادم للصفوف .

مثال: اكتب صف يمثل (يعرف) نقطة في الفراغ R^3 بالاحداثيات الديكارتية ثم استخدم هذا الصف في برنامج يقوم :

1. ادخال نقطتين من الفراغ R^3 .

2. حساب وطباعة البعد بين النقطتين .
3. طباعة النقطتين المدخلتين .

الحل :

```
include < iostream.h >
class point{puplic:
double x,y,z;
point (double a,double y,double c){
x = a; y = b; z = c;}
~point( ){cout << point is deleted;}
double dist(double x1,double y1,double z1){
double r = ((x − x1) * (x − x1) + (y − y1) * (y − y1) + (z − z1)
        * (z − z1))
r = Math.sqrt(r);
return r;}
void print ( ){cout << x << , << y << , << z << endl;}};
int main( ){point p1,p2;
double x1,y1,z1,x2,y2,z2;
cin >> x1 >> y1 >> z1 >> x2 >> y2 >> z2;
p1 = new point(x1,y1,z1);
p2 = new point(x2,y2,z2);
cout << p1.dist(p2.x,p2.y,p2.z);
p1.print;p2.print;return 0;}
```

- محدد الوصول عام أي يمكن الوصول الى الأعضاء العامة بعد كلمة public من قبل أي جزء اخر من البرنامج الذي يحتوي الصف .
- المعطيات الأعضاء داخل اي الصف هو مايلزم لتعريف هذا الشيء ،تعريف نقطة يلزم ثلاث احداثيات (هم x,y,z من النوع الحقيقي).

- باني الصف له اسم الصف يستقبل ثلاث وسطاء ويعطي قيمة الوسيط الأول ل(x) وقيمة الوسيط الثاني ل(y) وقيمة الوسيط الثالث ل(z)

هادم الصف له اسم الصف مسبوقا ب(~) كتبنا تعليمةالطباعة ان النقطة حذفت للتاكد ان الهادم يعمل

- دالة لحساب المسافة بين النقطتين من النوع double اسميناها dist تستقبل ثلاث وسطاء
- وعرفنا متغير r من النوع الحقيقي اسندنا له مجموع مربعات فرق احداثيات النقطتين
- Math هي مكتبة الرياضيات كتبناها من اجل دالة الجذر

دالة الطباعة

- صرحنا عن الأغراض
- صرحنا عن المتغيرات وادخلنا احداثيات النقطتين
- استدعينا الباني من اجل بناء النقطة الأولى والثانية
- استدعينا دالة حساب البعد
- ثم استدعينا دالة الطباعة

يمكننا كتابة دالة البعد بين نقطتين بطريقة ثانيه ف بدل ان نمرر لها الاحداثيات كل من على حدا نمررها كنقطة واحده كما يلي:

$$double\ dist(point\ p)\{$$

$$double\ r = ((x - p.x) * (x - p.x) + (y - p.y) * (y - p.y) + (z - p.z)$$

$$* (z - p.z))$$

$$r = Math\ .sqrt(r);$$

$$return\ r;\}$$

اذن لاستدعاء هذه الدالة نكتب :$cout \ll p1.dist(p2.x, p2.y);$

كما يمكننا كتابة دالة الطباعة بطريقة ثانية بان نمرر لها وسيط :

$$void\ print(point\ p)\{cout \ll p.x \ll , \ll p.y \lll\ll p.z \ll endl;\}$$

لاستدعاء هذه الدالة نكتب مثلا:

$$p1.print(\ p2);$$

$$p2.print\ (p1);$$

بالنسبة للمثال السابق على فرض اردنا بناء محور x فقط وكتبنا :

$$point(double\ a)\{x = a; y = 0; z = 0;\}$$

واردنا بناء محور y فقط وكتبنا :

$$point(double\ b)\{x = 0; y = b; z = 0;\}$$

واردنا بناء محور z فقط وكتبنا :

$$point\ (double\ c)\{x = 0; y = 0; z = c;\}$$

هل ماكتبناه سابقا صحيح ؟؟؟؟؟

كل ماكتبناه خاطئا فنحن لانستطيع الكتابة بهذا الشكل لان الباني يأخذ اسم الصف وبذلك تعريف اكثر من باني يلزم مفهوم التحميل الزائد للدوال (نكتب مثلا اكثر من باني على ان يكون الاختلاف بالترويسة)

لذلك نصحح ماسبق ونكتب :

$$point\ (double\ a, char\ c)\{if(c == `x`)\{x = a; y = 0; z = 0;\}$$

$$else\ if\ (c == `y`)\{x = 0; y = a; z = 0;\}else\{x = 0; y = 0; z = a;\}$$

تعلمنا سابقا في المؤشرات ان تعليمة newلحجز مكان جديد للمتغيرات في الذاكرة وتعليمة delete لتحرير مكان في الذاكرة وأيضا تعلمنا بالبرمجة الغرضية التوجه ان الباني كتعليمة new فهو يحجز أماكن للاغراض في الذاكرة والهادم يقوم بتحرير أماكن التي حجزت للأغراض في الذاكرة .

اذا لم نكتب محدد وصول فان محدد الوصول الافتراضي هو المحدد الخاص (private) ف كل ماهو موجود في الصف خاص ولايستطيع احد من خارج الصف الوصول لداخله.

اذا اردنا للدالة الرئيسية main رؤية كل ماهو داخل الصف نضع محدد الوصول public.

بالعودة الى باني الصف اذا كتبنا:

$$point\ (double\ x, double\ y, double\ z)\{x = x; y = y; z = z;\}$$

x الأولى هي من المعطيات الأعضاء و x الثانية هي من وسطاء

نحن نستطيع التمييز بينها لكن الكومبايلر لايمكنه ذلك ف اذا اردنا تمييز محتويات الصف نعطي اسم للصف خاص به نستطيع استخدامه داخل الصف فقط

أي الصف له اسم ثاني هو $this$

الكلمة المحجوزة this هي اسم اخر للصف داخل الصف نفسه ويمكن كذلك القول انها مؤشر تؤشر على الصف نفسه الموجودة فيه .

للتصحيح نكتب :

$$point\ (double\ x, double\ y, double\ z)\{$$

$$this \rightarrow x = x; this \rightarrow y = y; this \rightarrow z = z;\}$$

انتهت المحاضرة

إعداد: عبير خزنة كاتبي &سندس درويش&وفاء شيخ سالم