

MP 20-21 Tema 2

Arrays dinàmics

Sessió 13: Estructures de dades dinàmiques



Volem crear un conjunt de classes que ens permetin guardar els alumnes que s'han matriculat a una determinada titulació de la UAB. Per cada estudiant hem de poder guardar les seves dades bàsiques (NIU i nom) i el nom de totes les assignatures a les que s'ha matriculat. Suposem que hem creat aquestes classes per guardar les dades d'un estudiant i de tots els estudiants matriculats a una titulació.

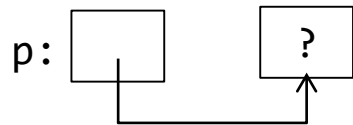
```
class Estudiant
{
private:
    string m_nom;
    string m_NIU;
    string m_assignatures[100];
    int m_nAssignatures;
};
```

```
class Titulacio
{
private:
    string m_nom;
    int m_nMaxAssignatures;
    Estudiant m_estudiants[MAX_ESTUDIANTS];
    int m_nEstudiants;
};
```

- Cada titulació determina un nº màxim d'assignatures que s'han de fer per poder completar la titulació.
- Volem que la longitud de l'array `m_assignatures` a `Estudiant` sigui exactament el nº màxim d'assignatures de la seva titulació.
 - Com haurem de modificar la declaració de l'array perquè pugui ser diferent per cada estudiant, depenent de la titulació?



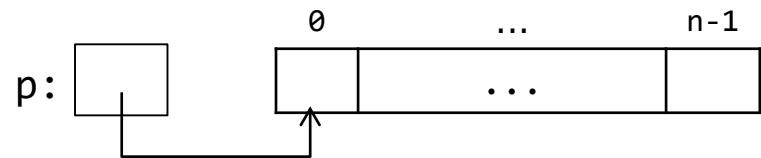
```
int* p;  
  
p = new int;
```



Objecte dinàmic:

- Creació d'un únic objecte dinàmic del tipus especificat (en aquest cas, `int`).

```
int n;  
int* p;  
  
cin >> n;  
p = new int[n];
```

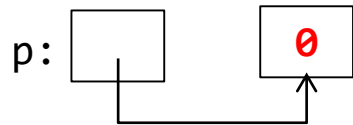


Array dinàmic:

- Creació de **`n` objectes dinàmics** del tipus especificat (en aquest cas, `int`) en **posicions consecutives de memòria**.
- Cada vegada que executem el codi el valor de `n` (la longitud de l'array) pot ser diferent: arrays de longitud variable



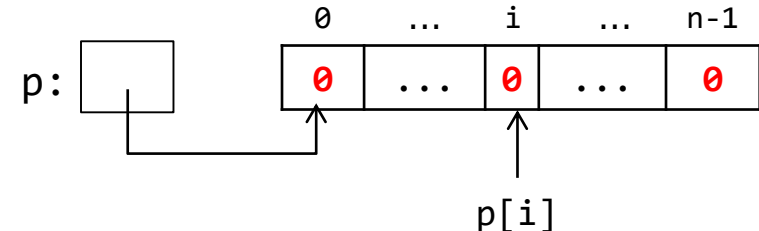
```
int* p;  
  
p = new int;  
  
*p = 0;
```



Objecte dinàmic:

- Accés al valor de l'objecte amb l'operador *.

```
int n;  
int* p;  
  
cin >> n;  
p = new int[n];  
  
for (int i = 0; i < n; i++)  
    p[i] = 0;
```



Array dinàmic:

- Utilitzant l'apuntador podem accedir als valors de l'array dinàmic igual que als valors d'un array estàtic.

$$p[i] \Leftrightarrow *(p + i)$$

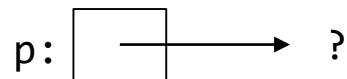


```
int* p;
```

```
p = new int;
```

```
*p = 0;
```

```
delete p;
```



Objecte dinàmic:

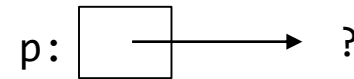
- Es destrueix l'objecte dinàmic i s'allibera la memòria.

```
int n;  
int* p;
```

```
cin >> n;  
p = new int[n];
```

```
for (int i = 0; i < n; i++)  
    p[i] = 0;
```

```
delete[] p;
```



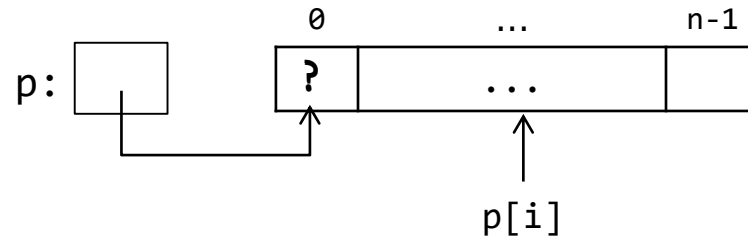
Array dinàmic:

- S'ha d'alliberar tota la memòria utilitzant **delete[] p**.



- **Arrays dinàmics:** arrays de tamany variable
 - El tamany de l'array es fixa en temps d'execució
 - Utilització de l'operador **new**, especificant el tamany de l'array
 - Un cop creat, es pot utilitzar com qualsevol altre array

```
int* p;  
int n;  
  
cin >> n;  
p = new int[n];  
  
...  
  
delete[] p;
```



Creació de l'array dinàmic

Alliberament de l'array dinàmic
És important no oblidar els corxets



Exercici

Volem adaptar el nº d'assignatures que guardem per cada estudiant al nº màxim d'assignatures de la titulació que fa:

1. **Modifiqueu** la **declaració** de la classe **Estudiant** per fer que l'array que guarda les assignatures sigui un array dinàmic.
2. **Afegeu** a la classe **Estudiant** un **constructor** que rebi com a paràmetre el nom, el niu de l'estudiant i el nº d'assignatures i inicialitzi les dades de l'estudiant i l'array dinàmic.
3. **Afegeu** a la classe **Estudiant** un **destructor** que alliberi la memòria ocupada per l'array dinàmic.
4. **Afegeu** a la classe **Estudiant** un **mètode** **afegeixAssignatura** que permeti afegir una nova assignatura a l'array dinàmic, a l'última posició de l'array.

```
class Estudiant
{
private:
    string m_nom;
    string m_NIU;
    string m_assignatures[100];
    int m_nAssignatures;
};
```

```
class Titulacio
{
private:
    string m_nom;
    int m_nMaxAssignatures;
    Estudiant m_estudiants[MAX_ESTUDIANTS];
    int m_nEstudiants;
};
```