

Séance 2

Classe

Déclaration d'une classe, notion d'objet

Parties publique et privée

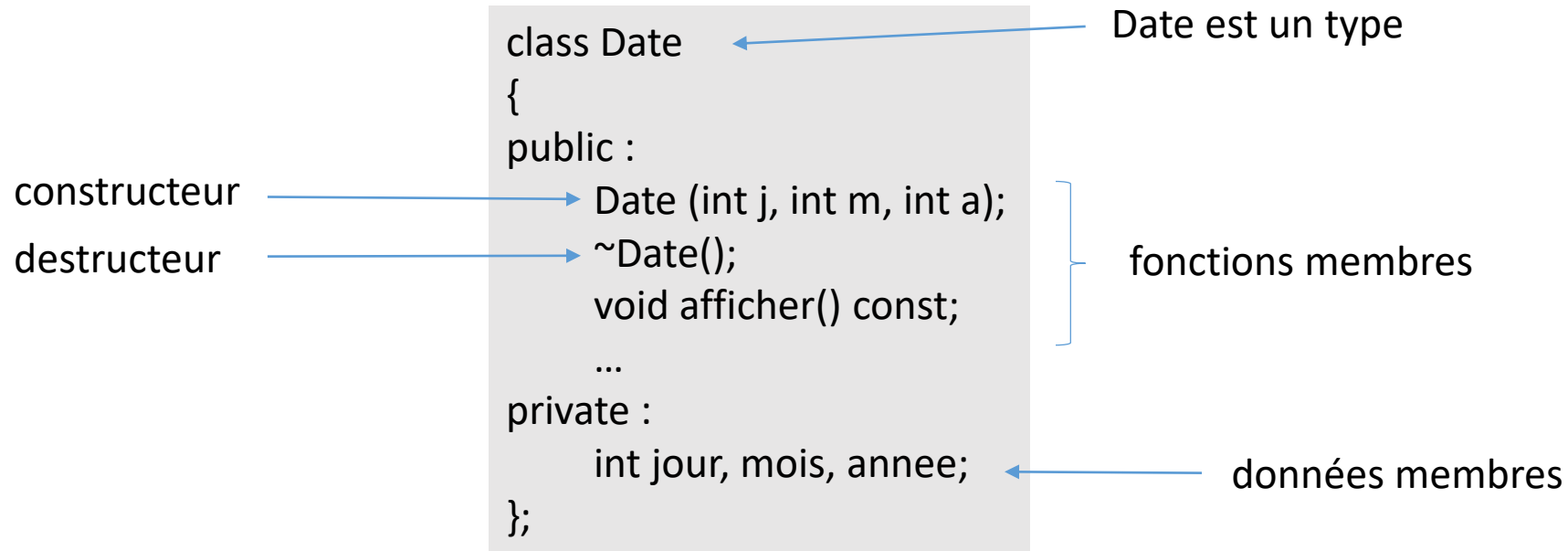
Fonction membre : déclaration, écriture, appel

Fonction membre : objet courant

Constructeur

Destructeur

Déclaration d'une classe, notion d'objet



Un **objet** : une variable dont le type est une classe

`Date d1...` `d1` est un objet

Parties publique et privée

public : accessible à toutes les fonctions du programme

private : accès limité aux fonctions membres

```
class Date
{
public :
    Date (int j, int m, int a);
    void afficher() const;
    ...
private :
    int jour, mois, annee;
};
```

```
int main() {
    Date d1...;
    if (d1.annee == ...)    ERR COMPILATION
}
```

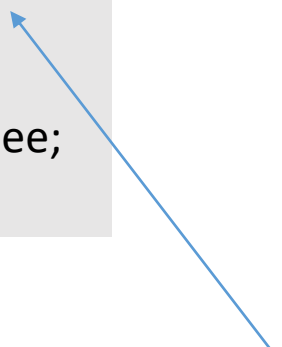
Il est conseillé de : cacher les données en les mettant en privé.

Fonction membre : déclaration, écriture, appel

```
class Date
{
public :
    ...
    void afficher() const;
    ...
private :
    int jour, mois, annee;
};
```

```
void Date::afficher() const {
    cout << jour << ...
}
```

```
Appel :
Date d1...;
d1.afficher();
```



la fonction ne modifie pas l'objet sur lequel elle s'applique

Fonction membre : objet courant

Objet courant : à l'intérieur d'une fonction membre, c'est l'objet sur lequel s'applique la fonction.

```
class Date
{
public :
    ...
    bool comparer(const Date& dat2) const;
    void afficher() const;
    ...
private :
    int jour, mois, annee;
};
```

```
bool Date::comparer(const Date& dat2) const {
    if (jour == dat2.jour && ...)
        ...
    afficher();
    dat2.afficher();
    ...
}
```

appartient à l'objet courant

s'applique sur l'objet courant

```
Date d1..., d2...
if (d1.comparer(d2) ...)
```

Constructeur

- Fonction membre : **même nom que la classe**, appelée automatiquement à chaque **création d'objet**.
- Rôle : initialiser l'objet.

```
class Date
{
public :
    Date (int j, int m, int a); (1)
    Date (int j, int m);      (2)
    Date ();                  (3)

    ...
private :
    int jour, mois, annee;
};
```

```
Date::Date(int j, int m, int a) { (1)
    jour = j;
    mois = m;
    annee = a;
}
```

```
Date::Date(int j, int m) { (2)
    jour = j;
    mois = m;
    annee = 2000;
}
```

```
Date::Date() { (3)
    jour = 1;
    mois = 1;
    annee = 2000;
}
```

Déclaration d'objet : passer les **paramètres** attendus par le constructeur.

```
Date d1(1, 9, 2023);    constructeur 1
Date d2(12, 4);         constructeur 2
Date d3;                constructeur 3, sans paramètre    attention : ne pas mettre les parenthèses
Date d4(15);            NON pas de constructeur correspondant
```

Destructeur

- Fonction membre : `~nom de la classe`, appelée automatiquement à chaque `destruction d'objet`.
- Rôle : libérer des ressources allouées par le constructeur.

```
class Date
{
public :
    ...
    ~Date ();
    ...
private :
    int jour, mois, annee;
};
```

```
Date::~~Date() {    pas d'utilité dans le cas de Date
}
```

Un seul destructeur, pas de paramètre