Séance 7
Héritage (2) – Flots d'Entrée/Sortie

Vocabulaire : polymorphisme

Polymorphisme du C++ : les fonctions virtuelles.

Un même appel (calculPaie sur un pointeur) => l'appel à des fonctions différentes.

Partie protégée

class MaClasse
{
public :
    accessible à toutes les fonctions

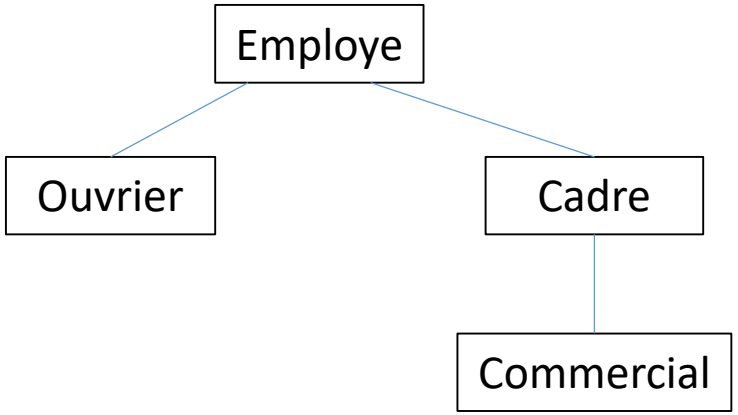
protected :
    accessible aux fonctions des classes filles de MaClasse

private :
    accessible aux fonctions membres de MaClasse
};

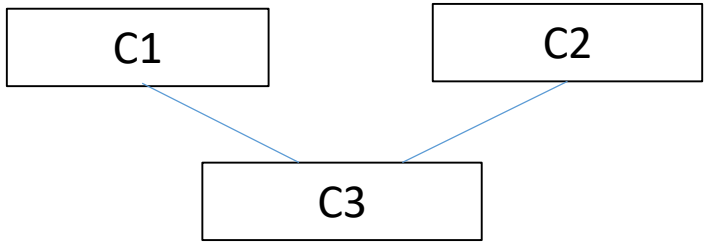
protected : à éviter; c'est préférable de garder seulement public/private.

Héritage multiple : définition

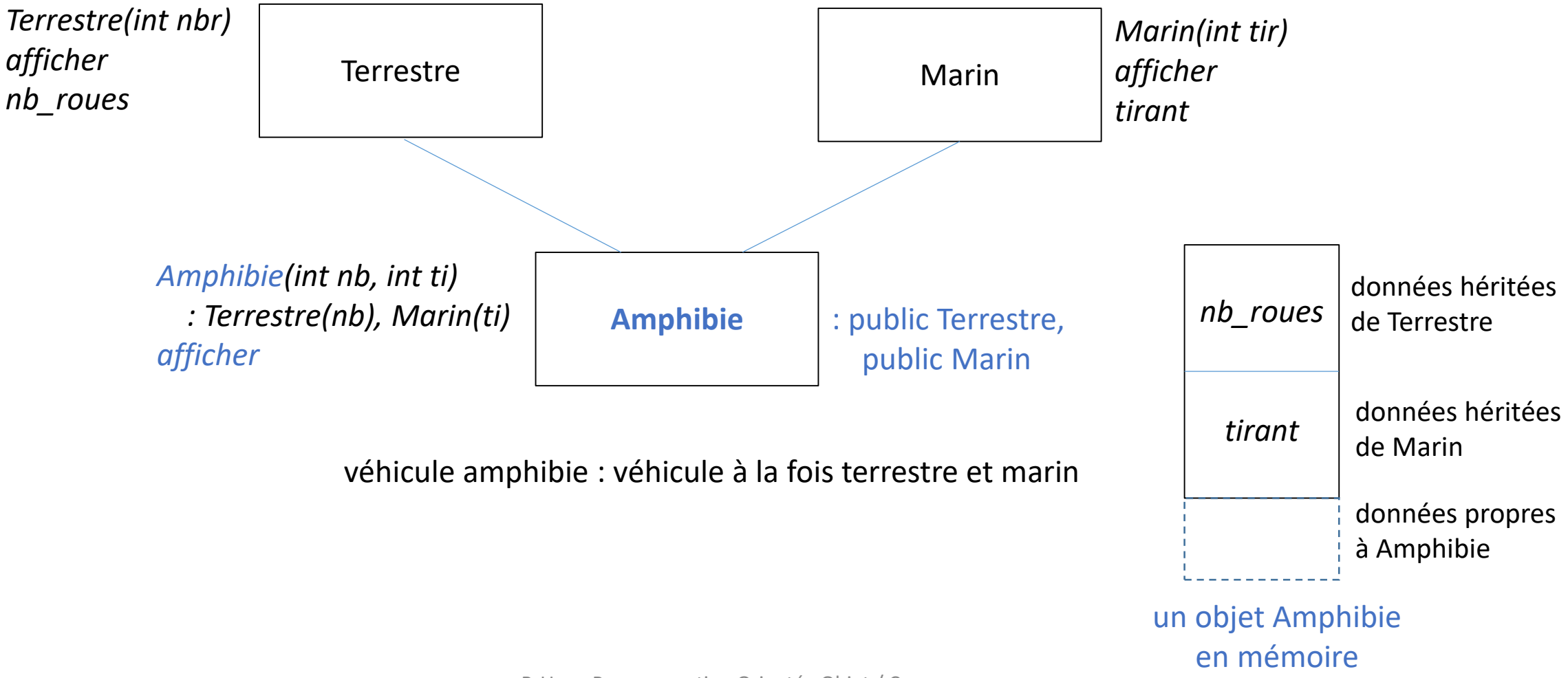
Héritage **simple** : une classe a une seule classe mère directe.



Héritage **multiple** : une classe a plusieurs classes mères directes.



Héritage multiple : un exemple



véhicule amphibie : véhicule à la fois terrestre et marin

Les différents flots d'E/S formatés

	en C	en C++
flots sur terminal	scanf, printf	cin (istream), cout (ostream)
flots sur chaine	sscanf, sprintf	istringstream, ostringstream
flots sur fichier	fscanf, fprintf	ifstream, ofstream

Lecture formatée dans un fichier texte

monfichier.txt :
bonjour 12
hello 56
…

pour lire un couple chaine/entier

```
#include <fstream>

ifstream fic("monfichier.txt");

char chaine[50];
int val;
fic >> chaine >> val;
```

pour lire tout le fichier

```
#include <fstream>

ifstream fic("monfichier.txt");

while (!fic.fail()) {
    char chaine[50];
    int val;
    fic >> chaine >> val;
    if (!fic.fail())
        // données lues ok
}
```

Ecriture formatée dans un fichier texte

```
#include <fstream>

ofstream fic("fic2.txt");          crée le fichier ou le vide s'il existe

int x = 215;
char ch[50] = "salut";
fic << x << " " << ch << endl;     écrit le texte 215 salut et retour à la ligne
```

Changement du format de sortie : les manipulateurs

```
#include <iomanip>
```

Largeur de champ et caractère de remplissage : manipulateurs setw et setfill

```
int x = 215, y = 32;
// affichage des valeurs sur 4 chiffres avec des 0 de remplissage
// 0215  0032
cout << setfill('0') << setw(4) << x << " " << setw(4) << y << endl;
```

setw doit être répété à chaque variable

Nombre de chiffres après la virgule : manipulateurs fixed et setprecision

```
float z = 4.257;
// affichage avec 2 chiffres après la virgule : 4.26
cout << fixed << setprecision(2) << z << endl;
```

Les manipulateurs sont utilisables sur tous les types de flot (terminal, chaine, fichier).