

不完全なクラス宣言 (C++ のみ)

Last Updated: 2022-01-21

不完全なクラス宣言とは、 クラス・メンバーを定義していないクラス宣言のことです。 宣言が完全なものになるまでは、そのクラス型のオブジェクトを宣言したり、 クラスのメンバーを参照することはできません。 ただし、クラスのサイズが必要でなければ、 不完全な宣言を使用して、クラスの定義の前にそのクラスに特定の参照を行うことはできます。

例えば、以下に示すように、構造体 `second` の定義で、構造体 `first` を指すポインターを定義することができます。 `second` の定義の前に、不完全なクラス宣言として構造体 `first` が宣言されています。 構造体 `second` 内の `oneptr` の定義では、`first` のサイズは必要ありません。

```
struct first;           // incomplete declaration of struct first

struct second          // complete declaration of struct second
{
    first* oneptr;      // pointer to struct first refers to
                       // struct first prior to its complete
                       // declaration

    first one;          // error, you cannot declare an object of
                       // an incompletely declared class type

    int x, y;
};

struct first           // complete declaration of struct first
{
    second two;        // define an object of class type second
    int z;
};
```

しかし、空のメンバー・リストを指定してクラスを宣言すると、それは完全なクラス宣言となります。 次に例を示します。

```
class X;                // incomplete class declaration
class Z {};            // empty member list
class Y
{
public:
    X yobj;             // error, cannot create an object of an
                       // incomplete class type

    Z zobj;             // valid
};
```