

C++

<C++>

Méthodes et Accès

Dorian.H Mekni

19 | APRIL | 2020

# Encapsulation

/\*

**Des méthodes qui permettent l'accès des données au sein d'une classe ->**

**SET() | GET()**

**Sans l'utilisation de ces méthodes, l'accès est interdit comme c'est le cas dans des propriétés stockées dans la partie private de la classe .**

**Les méthodes d'accès sont généralement d'ordre public.**

**Ce qui veut que même si les propriétés dans private sont interdites d'accès au public, des méthodes dites public permettant l'accès aux propriétés privées seront néanmoins possibles.**

\*/

# Les accesseurs |

**get()**



**Donne accès à l'attribut**

**set()**



**Permet la mutation d'un attribut**

# get() |

Méthode `get()` qui permet l'accès à la propriété `name` logée dans la partie privée de la classe `Human` instanciée dans notre fonction `main()` par `h`. `get` nous donne un accès mais ne nous permet pas de modifier l'attribut `name`.

```
5 class Human
6 {
7
8     public:
9         Human();
10        Human(std::string name);
11        Human(std::string name, int age);
12        Human(const Human &other);
13
14        std::string getName() const; // const sera le mot clé pour signaler que cette méthode ne
15                                     // pourra modifier l'objet qu'elle récupère.
16
17        void meeting(); // Fonction qui ne retourne rien
18
19    private:
20        std::string a_name;
21        int a_age;
22    };
23
```

```
std::string Human::getName() const
{
    return a_name;
}
```

# set() | initialisation

// paramètres const pour éviter les modifications de la désignation du paramètre -> dans le cas contraire, enlever const  
-| Utilisation de notre méthode set() pour test de compilation avec objectif de changer name

```
1 #ifndef class_hpp
2 #define class_hpp
3 #include<string>
4
5 class Human
6 {
7
8     public:
9         Human();
10        Human(std::string name);
11        Human(std::string name, int age);
12        Human(const Human &other);
13
14        std::string getName() const;
15        void setName(const std::string name); // Option Mutation
16
17        void meeting();
18
19     private:
20        std::string a_name;
21        int a_age;
22
23 };
24
25
26 #endif /* class_hpp */
```

```
void Human::setName(const std::string name)
{
    a_name = name; // Affectation de l'attribut de notre instance
}
```

# set() | Compilation

```
1  #include <iostream>
2  #include "class.hpp"
3  #include<string>
4
5  int main()
6  {
7
8      Human h{"Dorian", 36};
9
10     std::cout << h.getName() << std::endl;
11
12     h.setName("Marc");
13
14     std::cout << h.getName() << std::endl;
15
16     return 0;
17
18 }
```



```
Dorian
Marc
Program ended with exit code: 0
```

C++

->