Introduction aux classes (cpp)

Retour sur les connaissances de base:

Types principaux:

```
void, unsigned int, int, float, double, char
```

Types de bases des "int":

```
uint8_t  [0, 255]
uint16_t [0, 65535]
uint32_t [0, 4294967295]
```

```
int8_t  [-128, 127]
int16_t [-32768, 32767]
int32_t [-2147483648, 214748367]
```

Retour sur les connaissances de base:

Variables

```
int counter = 1;
float angle = 45.92f
int tableau[3] = {0, 1, 2};
```

Functions

```
int add(int a, int b);

[...]

int add(int a, int b)
{
    return a + b;
}
```

Retour sur les connaissances de base:

Variables

```
int counter = 1;
float angle = 45.92f
int tableau[3] = {0, 1, 2};
```

Functions

```
int add(int a, int b);

[...]

int add(int a, int b)
{
    return a + b;
}
```

Retour sur les connaissances de base:

Le scope

```c
int main(void)
{
    char a = 'a';
    {
        char b = 'b';
        {
            char c = 'c';
            printf("%c, %c, %c \n", a, b, c);
        }
        printf("%c, %c, %c \n", a, b, c); // c doesn't exist in this scope
    }
    printf("%c, %c, %c \n", a, b, c); // b and c doesn't exist in this scope
}
```

Définition d'une classe:

Un **type** personnalisé constitué de plusieurs variables et fonctions dans le but de simplifié le code pour l'humain

Éléments d'une classe:

- Membres (variables)
- Méthodes (functions)

Exemple:

```
class Moteur
{
public:
    Moteur()
    {

    }

    ~Moteur()
    {

    }
};
```

Exemple:

```
class Moteur
{
public:
    Moteur(){}
    ~Moteur(){}
};
```

Exemple:

```
class Moteur
{
public:
    // Membres
    int mode = 1;
    float position = 0.0f;
    float angle = 0.0f;

    // Methodes
    Moteur(){}
    ~Moteur(){}
};
```

```
int main(void)
{
    Moteur m1;
    printf("%i \n", m1.mode);
    m1.mode = 25;
    printf("%i \n", m1.mode);
}
```

```
Output:
    1
    25
```

Exemple:

```
class Moteur
{
public:
    // Membres
    float position;
    float lastPosition;
    unsigned long lastTime;

    // Methodes
    Moteur(){}
    ~Moteur(){}
    float getSpeed();
};
```

```
int main(void)
{
    Moteur m1;
    printf("%f \n", m1.getSpeed());
}
```

```
Output: 10.0
```

private:

- Les éléments sont seulements disponibles dans les méthodes de la classe
- Les éléments sont toujours privés si non-spécifié
- Les membres sont généralement tous privés par convention

protected:

- Les éléments sont accessible seulement dans la class ou par ses enfants*

public:

- Les éléments sont disponible en dehors de la classe

```
class Moteur
{
private:
    // Membres
    float position;
    float lastPosition;
    unsigned long lastTime;

public:
    // Methodes
    Moteur();
    ~Moteur();
    float getSpeed();
};

float Moteur::getSpeed()
{
    unsigned long dt = getTime() - lastTime;
    float speed = (lastPosition - position)/dt;
    return speed;
}
```

```
int main(void)
{
    Moteur m1;
    printf("%f \n", m1.position);
}
```

```
Erreur de compilation:
"m1.position is inaccessible"
```

```
class Moteur
{
private:
    // Membres
    float pos;
    float lastPos;
    unsigned long lastTime;

public:
    // Methodes
    Moteur();
    ~Moteur();
    float getSpeed();
};

float Moteur::getSpeed()
{
    unsigned long dt = getTime() - lastTime;
    float speed = (lastPosition - position)/dt;
    return speed;
}
```

```
int main(void)
{
    Moteur m1;
    printf("%f \n", m1.getSpeed());
}
```

```
Output: 10.00
```

Opérateur "::" (Namespace)

Ex:

```
float Moteur::getSpeed()
```

Namespaces

```
namespace LibRovus
{
    int add(int a, int b)
    {
        return a + b;
    }
}

namespace LibInternet
{
    int add(int a, int b)
    {
        printf("arg1: %i, arg2 %i \n", a, b);
        return a + b;
    }
}
```

```
int main(void)
{
    printf("%i \n\n", LibRovus::add(1, 2))
    printf("%i \n", LibInternet::add(1, 2))
}
```

Namespaces

```
namespace LibRovus
{
    int add(int a, int b)
    {
        return a + b;
    }
}

namespace LibInternet
{
    int add(int a, int b)
    {
        printf("arg1: %i, arg2 %i \n", a, b);
        return a + b;
    }
}
```

```
int main(void)
{
    printf("%i \n\n", LibRovus::add(1, 2))
    printf("%i \n", LibInternet::add(1, 2))
}
```

```
Output:
    3

    arg1: 1, arg2: 2
    3
```

```
class Moteur
{
private:
    // Membres
    float pos;
    float lastPos;
    unsigned long lastTime;

public:
    // Methodes
    Moteur();
    ~Moteur();
    float getSpeed();
};

float Moteur::getSpeed()
{
    unsigned long dt = getTime() - lastTime;
    float speed = (lastPosition - position)/dt;
    return speed;
}
```

```
int main(void)
{
    Moteur m1;
    printf("%f \n", m1.getSpeed());
}
```

```
Output: 10.00
```

Une classe dans une classe

```
class Vache
{
public:
    Vache() {}
    ~Vache() {}

    void meumeu()
    {
        printf("La vache fait meu");
    }
};

class Ferme
{
public:
    Ferme() {}
    ~Ferme() {}

    Vache monica;
}
```

```
int main(void)
{
    Ferme ferme;
    ferme.monica.meumeu();
}
```

```
Output:
    La vache fait meu
```