

クラスとは

- クラス(class)
インスタンス(実体)のもととなる設計図

例えば、「人間」という構造の生物は、Aさん、Bさんみたいな多種多様な「個人」に派生する。
ここでいう「人間」がクラスで、「個人」がインスタンス(実体)

C言語の構造体と似ているが、構造体と違うところはメンバ変数だけでなくメンバ関数を持てること

クラスの例

- 教科書P96~97 Sample301
- C++用フォルダにSample301フォルダを作成
`mkdir Sample301`
`cd Sample301`
- car.h、car.cpp、main.cppを新規作成
`copy nul car.h`
`copy nul car.cpp`
`copy nul main.cpp`

クラスの例

- car.h

```
#ifndef _CAR_H_
#define _CAR_H_

class Car {
public:
    double speed; //メンバ変数
    void drive(double hour); //メンバ関数
};
#endif // _CAR_H_
```

クラスの例

- car.h

```
#ifndef _CAR_H_
```

```
#define _CAR_H_
```

```
class Car {
```

```
public:
```

```
    double speed; //メンバ変数
```

```
    void drive(double hour); //メンバ関数
```

```
};
```

```
#endif // _CAR_H_
```

クラスの例

- car.h

```
#pragma once
```

```
class Car {  
public:  
    double speed;           //メンバ変数  
    void drive(double hour); //メンバ関数  
};
```

#ifndef, #define, #endif は使わず
【#pragma once】を使ってください

クラスの例

- car.cpp

```
#include "car.h"
#include <iostream>
using namespace std;

void Car::drive(double hour){
    cout << "時速" << speed << "kmで" <<
        hour << "時間走行" << endl;
    cout << speed*hour << "km移動"<< endl;
}
```

クラスの例

- main.cpp

```
#include "car.h"
using namespace std;

int main() {
    Car kuruma; //テキストではkurumaでなくcar
    kuruma.speed = 40;
    kuruma.drive(1.5);
    return 0;
}
```

クラスの例

- コンパイルの仕方

コマンドプロンプトで次のコマンドを入力する

```
cl _/EHsc _main.cpp _car.cpp
```

重要なのはcppファイルをすべて書くこと

成功したら、main.exeを実行して結果を確認

クラスの定義

- car.h

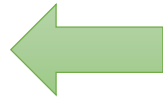
```
#pragma once
```

```
class Car {  
public:
```

```
    double speed;
```

```
    void drive(double hour);
```

```
};
```



二重インクルードの禁止

```
#ifndef _CAR_H_  
#define _CAR_H_  
#endif//_CAR_H_
```

インクルードガード処理という
2重インクルードを防ぐ仕組み
#pragma once で代用可能
(P.104~105)

//メンバ変数

//メンバ関数

クラスの定義

- car.h

```
#pragma once
```

```
class Car {  
public:  
    double speed;           //メンバ変数  
    void drive(double hour); //メンバ関数  
};
```

← クラス名(先頭は大文字にするのが一般的)

クラスの定義

• car.h

```
#pragma once
```

```
class Car {
```

```
public: ← アクセス指定子
```

```
    double speed; //メンバ変数
```

```
    void drive(double hour); //メンバ関数
```

```
};
```

アクセス指定子(P.113)

- ① **public**
クラスやクラス外からアクセス可
- ② **private**
クラス内からしかアクセス不可
- ③ **protected**
クラスとサブクラスからアクセス可

クラスの本体

- `car.cpp`

`car.h`に記述した内容によって
メンバ関数を定義する必要があるので
`Car`クラスの設計図をここで読み込んで
おく

```
#include "car.h"
#include <iostream>
using namespace std;

void Car::drive(double hour){
    cout << "時速" << speed << "kmで" <<
        hour << "時間走行" << endl;
    cout << speed*hour << "km移動" << endl;
}
```

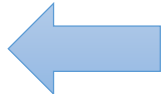
クラスの本体

- car.cpp

```
#include "car.h"
#include <iostream>
using namespace std;

void Car::drive(double hour){
    cout << "時速" << speed << "kmで" <<
        hour << "時間走行" << endl;
    cout << speed*hour << "km移動" << endl;
}
```

これがないと
std::cout , std::endl
と書く必要がある



クラスの本体

- car.cpp

```
#include "car.h"
```

```
#include <iostream>
using namespace std;
```

【Car::】でCarクラスで定義されていることを表す

car.hで定義したメンバ関数の実際の処理を記述

```
void Car::drive(double hour){
    cout << "時速" << speed << "kmで" <<
        hour << "時間走行" << endl;
    cout << speed*hour << "km移動" << endl;
}
```

クラスの例

- main.cpp

```
#include "car.h"  
#include <iostream>  
using namespace std;
```

```
int main() {  
    Car kuruma;  
    kuruma.speed = 40;  
    kuruma.drive(1.5);  
    return 0;  
}
```

Carクラスのインスタンスとして
kurumaという実体を生成



クラスの例

- main.cpp

```
#include "car.h"  
#include <iostream>  
using namespace std;
```

car.hに記述したメンバ変数／関数を使用するために読み込んでおく必要がある

メンバ変数やメンバ関数を使用する際は構造体のときと同様に「.」を使う

```
car kuruma;  
kuruma.speed = 40;  
kuruma.drive(1.5);  
return 0;
```

kurumaの

メンバ変数へ値を代入

メンバ関数を実行

```
}
```


クラスの例

- main.cpp

(略)

```
int main() {  
    Car nbox, tanto; // 複数のインスタンス生成  
    nbox.speed = 40;  
    nbox.drive(1.5);  
    tanto.speed = 50;  
    tanto.drive(1.0);  
    return 0;  
}
```

nboxおよびtantoはCarクラスから生成されたインスタンスなので、同じメンバ構成をしている
ただし、個々は独立して存在している

クラスの例(Vector2D)

- 教科書P106~109 Example301
- C++用フォルダにExample301フォルダを作成
`mkdir Example301`
`cd Example301`
- car.h、car.cpp、main.cppを新規作成
`copy nul vector2D.h`
`copy nul vector2D.cpp`
`copy nul main.cpp`

クラスの例(Vector2D)

- vector2D.h

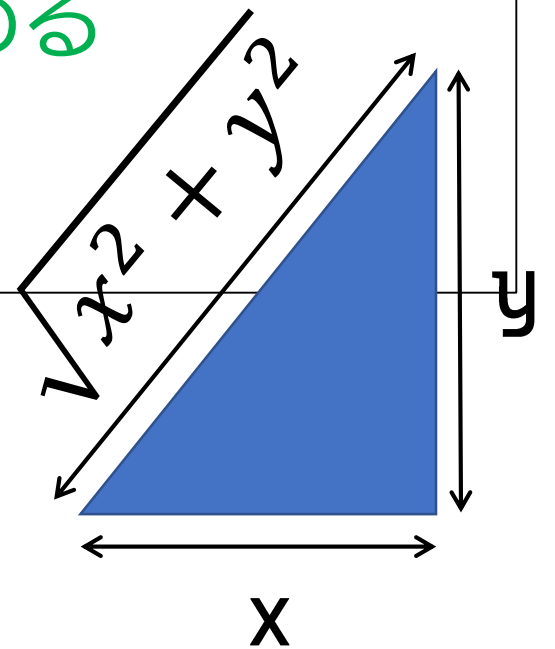
```
#pragma once
```

```
class Vector2D {  
public:  
    double x,y;           //メンバ変数  
    double length();      //メンバ関数  
};
```

クラスの例(Vector2D)

- vector2D.cpp

```
#include "vector2D.h"  
#include <math.h>  
double Vector2D::length() {  
    //三平方の定理を使って斜辺の長さを求める  
    return sqrt(x*x + y*y);  
};
```



クラスの例(Vector2D)

- main.cpp その①

```
#include "vector2D.h"
#include <iostream>
using namespace std;
int main(){
    Vector2D v; //Vector2Dクラスのインスタンスv
    cout << "v.x=" << endl;
    cin >> v.x;
    cout << "v.y=" << endl;
    cin >> v.y;
```

クラスの例(Vector2D)

- main.cpp その②

```
cout << "v=( " << v.x << " , " << v.y  
      << " )" << endl;  
cout << "vの長さ:" << v.length() << endl;  
return 0;  
}
```