C++ 009 静的メンバ

【静的メンバ】

インスタンス化せずに使用できるクラスのメンバを「静的メンバ」と称します。

＜静的メンバの宣言＞

```
static データ型 変数名:                          //静的メンバ変数
static 戻り値のデータ型 関数名 ( 引数 );     //静的メンバ関数
```

宣言時に「static」が付きます。

＜静的メンバの使用＞

```
クラス名::メンバ名
```

＜例文＞

```
class sample {
public:
    static int var;        //静的メンバ変数の宣言
    static void func();    //静的メンバ関数の宣言
};

int sample::var = 0;      //var の実装及び初期化
void sample::func() {     //func の実装
    //処理を記述
}

int main() {
    sample::var = 10;      //静的メンバ変数を使用
    sample::func();        //静的メンバ関数を使用
```

静的メンバ変数はグローバル変数と類似しています。

値はスタティック領域に確保され、いつでも代入、参照ができ、初期化しない場合、基本データ型は 0、ポインタ変数は NULL で初期化されます。

静的メンバ変数はコンストラクタでは初期化できません。

通常はインスタンス化してメンバを使用しますが

```
sample instance();    //インスタンス化
instance.var = 10;    //メンバ変数を使用
instance.func();      //メンバ関数を使用
```

静的メンバはインスタンス化せず「クラス名::メンバ名」で使用できます。

```
sample::var = 10;    //静的メンバ変数を使用
sample::func();      //静的メンバ関数を使用
```

＜注意＞

静的メンバは通常メンバ（静的メンバではないメンバ）を使用できません。

```
class sample {
public:
    int var;                //通常メンバ
    static void func();     //静的メンバ
};

void sample::func() {
    var = 10;               //通常メンバは使用できないのでエラーになる
}
```

【const 付き静的メンバ変数】

```
class sample {
public:
    static const int var = 100;
```

const が付いている静的メンバ変数は初期化した値を変更できません。

```
static const std::string var = "abc";
```

string は const 付き静的メンバ変数できないので注意。

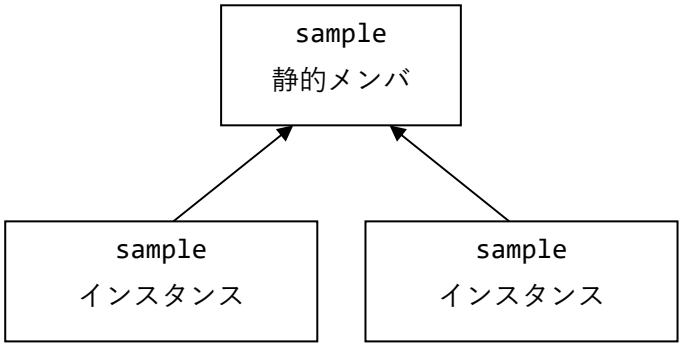
【静的メンバは共有される】

```
class sample {
public:
    static int var;        //静的メンバ変数
    static void func();    //静的メンバ関数
};

int sample::var = 0;
void sample::func(){
    std::cout << sample::var << std::endl;    //var の値を出力
}

int main(){
    sample instance1;      //インスタンス化
    instance1.var = 10;    //var に 10 を代入する
    sample instance1;      //別のインスタンス
    instance1.func();      //「10」と出力される
```

静的メンバはインスタンスからも使用できます。
その場合、静的メンバは複数のインスタンスで共有化されます。



ローカル変数名とメンバ変数名が重複した場合、通常メンバは「this->メンバ名」で区別しますが、静的メンバは「クラス名::メンバ名」と記述します。もちろん名前が重複しなければメンバ名のみでも可能です。