

第5章 Overload と Template

5.1 関数のオーバーロードとは

C 言語で、二つの値を比較して大きいほうを返却する関数を書いたとします。

```
1 int max(int x, int y)
2 {
3     if( x > y )
4         return x;
5     else
6         return y;
7 }
```

関数を呼び出す時は、

```
int i = max( 1 , 2 );
```

などとすればよいでしょう。

さて、ここで問題となるのが、引数に `double` 型を渡したい場合です。max 関数は引数に `int` を取っているので、

```
int i = max( 1.5 , 2.0 );
```

等とすることはできません。こういった場合、C 言語では `max` の名前を変更するしかありませんでした。

```
1 int max_int(int x, int y)
2 {
3     if( x > y )
4         return x;
5     else
6         return y;
7 }
```

```
8 double max_double(double x, double y)
9 {
10     if( x > y )
11         return x;
12     else
13         return y;
14 }
```

この問題を回避する為に C++ 言語では同じ関数名でも引数が違えば違う関数として扱うことができるようになりました。このことを関数のオーバーロード (Overload) といいます。

```
1 int max(int x, int y)
2 {
3     if( x > y )
4         return x;
5     else
6         return y;
7 }
8 double max(double x, double y)
9 {
10     if( x > y )
11         return x;
12     else
13         return y;
14 }
```

呼び出す時は、

```
int i = max( 1,2 ); //max(int,int) が呼ばれる
double j = max( 1.5,2.2 ); //max(double,double) が呼ばれる
```

とします。

5.2 Template とは

5.1 節の max 関数は引数が違うだけで変数の中身は全く同じでした。この場合、わざわざ引数だけを変えた関数をたくさん作るのは面倒です。そこで登場するのが Template です。次のコードを見てください。

```
1  template<typename T>
2  T my_max(T x,T y)
3  {
4      if( x > y )
5          return x;
6      else
7          return y;
8  }
```

T 型という新しい型を作り出しています。この型は呼び出し時にコンパイラが自動的に型を判断して処理を行ってくれます。

関数の呼び出しは

```
int i = my_max( 1,2 );
double j = my_max( 1.5,2.2 );
```

となります。引数に 1 を渡した場合、T 型は int 型に置き換えられます。同様に 1.5 を渡した場合、T 型は double 型に置き換えられます。

template の使い方

```
template<typename 新しく定義する型の名前>
```


参考文献

- [1] ハーバート・シルト著, トップスタジオ訳, 独習 C++改訂版.