

C++ワンポイントレッスン

知っていると便利なC++のテクニック

ソフトウェア研究会

戻り値による関数のオーバーライド

- C++は、異なった引数による同一名の関数を利用できます。

```
void hoge(int a);  
void hoge(char* b);  
void hoge(float c);
```

- C++には戻り値の違いによる同一関数は定義できません。

```
int getValue();  
char* getValue(); //エラー  
float getValue(); //エラー
```

ところが、以下のようなプログラムをエラーなしにコーディングできます。

```
Hoge hoge;  
int a = hoge.getValue();  
char* b = hoge.getValue();  
float c = hoge.getValue();
```

戻り値の型によるオーバーライド

種明かしをすると、任意の型に暗黙のうちに変換が可能なクラスを用意し、そのクラスを戻り値として間接的に型変換を行わせるというテクニックです。

下の例では、Xというクラスオブジェクトを戻り値とし、Xはそれぞれint, char*, float型への暗黙の型変換が可能なため、見掛け上はgetValueが戻り値によってオーバーライドされている状態と同じ動作になります。

```
struct Hoge {  
    struct X {  
        operator int () { ins.getInt(); }  
        operator char* () { ins.getChar(); }  
        operator float() {ins.getFloat(); }  
        X(Hoge& h) : ins(h) {}  
        Hoge& ins;  
    };  
  
    X getValue() { return X(*this); }  
  
    int getInt();  
    char* getChar();  
    float getFloat();  
}
```

```
ColorRGB a = ColorRGB(0, 0, 0);  
vector<ColorRGB> a = {ColorRGB(0, 0, 0), ColorRGB(1, 1, 1) };
```