

## C++でのenum(列挙型)の使い方とは? class指定方法や文字列変換方法を紹介!

プログラミング言語

公開日時: 2021-03-17 更新日時: 2022-06-06



この記事でわかること ▾

基本情報技術者試験の試験対策はこちら&gt;&gt;



enumの使い方が知りたいです。

システム  
エンジニアプロジェクト  
マネージャー

enumの基本的な使い方やclass指定などを詳しく見ていきましょう。

### C++でのenum(列挙型)の使い方とは?



今日は、C++でのenum(列挙型)の使い方について説明します。

ここでは、以下について紹介します。

- ・基本的な使い方
- ・class指定
- ・文字列変換
- ・繰り返し処理

C++でのenum(列挙型)の使い方に興味のある方はぜひご覧ください。

#### 基本的な使い方

C++でのenumの基本的な使い方を紹介します。

実際のソースコードを見てみましょう。

リスクリングの実態調査について  
アンケート実施中

あと11問

リスクリングを  
実施している人  
へ  
「リスクリング  
の実態調査」ア  
ンケートのご協  
力をお願いして  
おります。  
(所要時間2  
分)

性別を教えてく  
ださい ※必須

選択してください... ▾

年齢を教えてく  
ださい ※必須

選択してください... ▾

現在の年収を教  
えてください ※必  
須



```

3 // enumの定義
4 enum color {
5   Red,    // 0
6   Blue,   // 1
7   Green,  // 2
8   Yellow, // 3
9   Purple, // 4
10  White,  // 5
11  Black   // 6
12 };
13 };
14
15 int main(void){
16   // enumの宣言
17   enum color c0, c1, c2;
18
19   // enumのメンバを代入
20   c0 = Red;
21   c1 = Green;
22   c2 = Purple;
23
24   // 結果出力
25   printf("Red: %d\n", c0);
26   printf("Green: %d\n", c1);
27   printf("Purple: %d\n", c2);
28
29   return 0;
30 }

```

実行結果は以下のようになります。数値を指定しなければ、0からインクリメントされて採番されます。

```

1 Red: 0
2 Green: 2
3 Purple: 4

```

値を指定したい場合は、以下のように記述することもできます。

```

1 enum color {
2   Red = 1, // 1
3   Blue,    // 2
4   Green,   // 3
5   Yellow,  // 4
6   Purple,  // 5
7   White,   // 6
8   Black    // 7
9 };

```

この場合は、1からインクリメントされて採番されます。実行結果は以下のようになります。

```

1 Red: 1
2 Green: 3
3 Purple: 5

```

少し特殊ですが、以下のように指定することもできます。この場合、指定した数値からインクリメントされていきます。

```

1 enum color {
2   Red = 1, // 1
3   Blue,    // 2
4   Green = 4, // 4 指定した数値になる
5   Yellow,  // 5 そのあとからインクリメントされる
6   Purple,  // 6
7   White,   // 7
8   Black    // 8
9 };

```

実行結果は以下のようになります。指定した数値からインクリメントされていることが分かります。

```

1 Red: 1
2 Green: 4
3 Purple: 6

```

C++のenumはこのように使います。

## class指定

C++でのenum classの使い方を紹介します。実際のソースコードを見てみましょう。

numはintとして扱われるため、以下のように代入できてしまい、型安全ではなく危険性がありました。

```

1 #include <iostream>
2 using namespace std;
3
4 // enumの定義
5 enum color {
6   Red,    // 0
7   Blue,   // 1
8   Green,  // 2
9   Yellow, // 3
10  Purple, // 4
11  White,  // 5
12  Black   // 6
13 };
14
15 int main(void){
16
17   // intへ代入(代入できてしまい、型安全ではなく危険性がある)
18   int val = Blue;
19   printf("val: %d\n", val);
20
21   return 0;
22 }

```

実行結果は以下のようになります。代入できてしまっていることが分かります。



```

1 enum class color {
2     Red,    // 0
3     Blue,   // 1
4     Green,  // 2
5     Yellow, // 3
6     Purple, // 4
7     White,  // 5
8     Black   // 6
9 };

```

この場合、以下のようなコンパイルエラーが発生します。

```

1 error: use of undeclared identifier 'Blue'; did you mean 'true'?
2     int val = Blue;
3             ^~~~
4             true
5 1 error generated.

```

C++でのenum classの使い方を紹介しました。

## 文字列変換

C++でのenumの列挙定数を文字列に変換する方法を紹介します。実際のソースコードを見てみましょう。マクロの引数を文字列リテラルに置換するマクロ関数を使用します。

```

1 #include <iostream>
2 #include <cstdio>
3 #define STR(var) #var // マクロの引数を文字列リテラルに置換するマクロ関数
4 using namespace std;
5
6 // enumの定義
7 enum color {
8     Red,    // 0
9     Blue,   // 1
10    Green,  // 2
11    Yellow, // 3
12    Purple, // 4
13    White,  // 5
14    Black   // 6
15 };
16
17 int main(void){
18     // 結果出力
19     printf("%s: %d\n", STR(Red), Red);
20     printf("%s: %d\n", STR(Green), Green);
21     printf("%s: %d\n", STR(Purple), Purple);
22
23     return 0;
24 }

```

実行結果は以下のようになります。列挙定数が文字列に変換されていることが分かります。

```

1 Red: 0
2 Green: 2
3 Purple: 4

```

このように、C++ではenumの列挙定数を文字列に変換できます。

## 繰り返し処理

C++でのenum要素の繰り返し方法を紹介します。実際のソースコードを見てみましょう。operator ++を実装する必要があります。

```

1 #include <iostream>
2 #include <cstdio>
3 using namespace std;
4
5 // enumの定義
6 enum color {
7     Begin,
8     Red = Begin, // 0
9     Blue,        // 1
10    Green,       // 2
11    Yellow,      // 3
12    Purple,      // 4
13    White,       // 5
14    Black,       // 6
15    End
16 };
17
18 void some_function(color c)
19 {
20     // switchによる分岐処理
21     switch( c )
22     {
23         case Red :
24             printf("Red %d\n", c);
25             break;
26         case Blue :
27             printf("Blue %d\n", c);
28             break;
29         case Green :
30             printf("Green %d\n", c);
31             break;
32         case Yellow :
33             printf("Yellow %d\n", c);
34             break;
35         case Purple :
36             printf("Purple %d\n", c);
37             break;
38         case White :
39             printf("White %d\n", c);
40             break;
41         case Black :
42             printf("Black %d\n", c);
43             break;

```



```

47     }
48 }
49
50 // operator ++
51 color& operator ++ (color& c)
52 {
53     if (c == color::End) {
54         throw std::out_of_range("for color& operator ++ (color&)");
55     }
56     c = color(static_cast<std::underlying_type<color>::type>(c) + 1);
57     return c;
58 }
59
60 int main(void){
61
62     // enumの要素数分繰り返し
63     for (color c = color::Begin; c != color::End; ++c) {
64         some_function(c);
65     }
66
67     return 0;
68 }

```

実行結果は以下のようになります。この場合、連続する数値の場合のみ対応可能です。

```

1 Red 0
2 Blue 1
3 Green 2
4 Yellow 3
5 Purple 4
6 White 5
7 Black 6

```

enumの要素数分繰り返し処理が行えていることが分かります。



enumは他にも、文字列変換や繰り返し処理もできるのですね。

システム  
エンジニア



プロジェクト  
マネージャー

実際に自分でコードを書いてenumを覚えてみましょう。

## まとめ

いかがでしたでしょうか。C++でのenumの使い方について説明しました。

enumの基本的な使い方から、class指定方法や文字列変換方法、繰り返し処理について紹介しました。

ぜひご自身でソースコードを書いて、理解を深めてください。



Fenet.NETナビ・.NETコラムは株式会社オープンアップシステムが運営しています。



株式会社オープンアップシステムはこんな会社です

秋葉原オフィスには株式会社オープンアップシステムをはじめグループのIT企業が集結!  
数多くのエンジニアが集まります。



スマホアプリから業務系システムまで





## 充実した研修制度



毎年、IT技術のトレンドや社員の要望に合わせて、カリキュラムを刷新し展開しています。社内講師の丁寧なサポートを受けながら、自分のペースで学ぶことができます。

## 資格取得を応援



スキルアップしたい社員を応援するために資格取得一時金制度を設けています。受験料(実費)と合わせて資格レベルに合わせた最大10万円の一時金も支給しています。

## 東証プライム上場企業グループ



オープンアップシステムは東証プライム上場「株式会社オープンアップグループ」のグループ企業です。

安定した経営基盤とグループ間のスムーズな連携でコロナ禍でも安定した雇用を実現させています。

## 株式会社オープンアップシステムに興味を持った方へ

株式会社オープンアップシステムでは、開発系エンジニア・プログラマを募集しています。

年収をアップしたい! キャリアアップしたい! 大手の上流案件にチャレンジしたい!  
まずは話だけでも聞いてみたい場合もOK。お気軽にご登録ください。



株式会社オープンアップシステムへ応募する

会員登録不要! カンタン45秒!

## 新着案件 NEW JOB

## 開発エンジニア/東京都品川区/【WEB面談可】/在宅ワーク

¥ 月給29万~30万円

📍 東京都品川区(大崎駅)

Windows

## 遠隔テストサービス機能改修/JavaScript/東京都港区/【WEB面談可】/テレワーク

¥ 月給45万~60万円

📍 東京都港区(六本木駅)

JavaScript

## 病院内システムの不具合対応、保守/東京都豊島区/【WEB面談可】/テレワーク

¥ 月給30万~30万円

📍 東京都豊島区(池袋駅)

Windows

## 開発/JavaScript/東京都豊島区/【WEB面談可】/テレワーク

¥ 月給50万~50万円

📍 東京都豊島区(大塚駅)

JavaScript

## 債権債務システム追加開発/東京都文京区/【WEB面談可】/在宅勤務

¥ 月給62万~67万円





PMO/東京都豊島区/【WEB面談可】/在宅勤務

¥ 月給55万~55万円

📍 東京都豊島区(池袋駅)

Windows

**関連記事 RELATED POSTS**

関連記事はありません。



FEnetはオープンアップグループのグループシナジーを活かして立ち上げられた転職サイトです。  
他のシリーズサイトでも案件・求人情報を掲載中。FEnetの各シリーズこちら。

ITインフラの求人・案件



.NETの求人・案件



Javaの求人・案件



AWSの求人・案件



Salesforceの求人・案件



ITエンジニアを目指す



Copyright © 2023 FEnet All Rights Reserved.

