

C++ コンストラクタ

[C++][C++ によるオブジェクト指向プログラミング]

コンストラクタ

- ・ コンストラクタを使用して、オブジェクトの初期化を行うことができる
- ・ 戻り値の型を持たない

randum クラス

- ・ 乱数を生成する
- ・ コンストラクタの引数として初期値を渡す

```
class randum
{
public:
    // コンストラクタ
    randum(unsigned int i) {
        r = i;    // 乱数の初期値を初期化
    }

    unsigned int number() {
        r = (r * 109 + 1021) % 32768;
        return r;
    }
private:
    unsigned int r;
};
```

宣言の例

- ・ 引数を 1 つ持つコンストラクタが 1 つあるだけなので、引数を渡さない宣言は無効

```
randum r(13);    // 有効
randum r2;       // 無効
```

デフォルトコンストラクタ

- ・ 引数を要求しないコンストラクタをデフォルトコンストラクタと呼ぶ
- ・ 引数が空のコンストラクタ、もしくはすべての引数がデフォルト値を持つコンストラクタ

引数のないデフォルトコンストラクタ

```
// デフォルトコンストラクタ
randum() { r = 13; }

// コンストラクタ
randum(unsigned int i) { r = i; }
```

デフォルト値を持つコンストラクタ

```
randum(unsigned int i = 13){}
```

宣言の例

- ・初期値を渡さない宣言が可能
- ・配列として宣言が可能

```
random r;           // 有効
random r(15);       // 有効
random r2[5];       // 有効
```

コンストラクタ初期化子

- ・オブジェクトの下位要素をコンストラクタで初期化する構文
- ・個々のメンバーは、次のように初期化子、カンマで区切って並べる

```
member_name(expression_list)
```

- ・コンストラクタ本体の内部では、メンバに値を代入できない場合がある
 - ・非静的メンバが `const` 型、または参照型である場合は初期化子リストが必要になる

初期化子を持つコンストラクタ

```
random(unsigned int i = 13) : r(i) {}
```

変換関数としてのコンストラクタ

- ・パラメータを 1 つだけ持つコンストラクタは、キーワード `explicit` で宣言されていない限り、自動的に変換関数となる

例

```
unsigned int i = 23;
random r;
r = i;
```