

# Boostライフライリー周の旅

**ver.1.48.0**

高橋晶 (Akira Takahashi)  
[id:faith\\_and\\_brave](#)  
[@cpp\\_akira](#)

前回は、Boost 1.44.0までのライブラリを紹介しました。

今回は1.45.0から1.48.0までに追加されたライブラリを紹介していきます。

1. Interval Container
2. Chrono
3. Geometry
4. Phoenix
5. Container
6. Move

区間演算のコンテナを提供するライブラリ。

```
typedef std::set<string> guests;  
interval_map<ptime, guests> party;  
party += make_pair(interval<ptime>::right_open(  
    time_from_string("20:00"),  
    time_from_string("22:00")),  
    make_guests("Mary"));  
  
party += make_pair(interval<ptime>::right_open(  
    time_from_string("21:00"),  
    time_from_string("23:00")),  
    make_guests("Harry"));
```

[20:00, 21:00)->{"Mary"}

[21:00, 22:00)->{"Harry", "Mary"} //時間帯が重なっていたら集約される

[22:00, 23:00)->{"Harry"}

時間計算のためのライブラリ。

C++11標準ライブラリに導入されたものと、その拡張。

```
// 500ナノ秒遅延する
namespace chrono = boost::chrono;

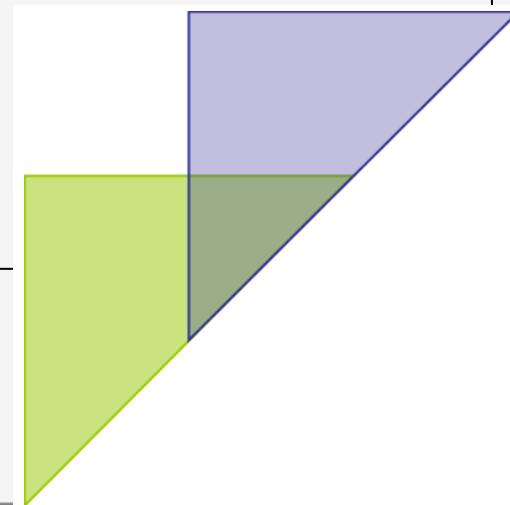
auto go = chrono::steady_clock::now() + chrono::nanoseconds(500);
while (chrono::steady_clock::now() < go)
    ;
```

様々な時間の単位と、いくつかの特性をもった時計クラスが提供される。CPU時間を扱う拡張もある。

計算幾何のライブラリ。

N次元の点、線、三角形、四角形などのモデルと、それらに対するアルゴリズムが提供される。

```
polygon a, b;  
geometry::exterior_ring(a) =  
    assign::list_of<point>(0, 0)(3, 3)(0, 3)(0, 0);  
  
geometry::exterior_ring(b) =  
    assign::list_of<point>(1.5, 1.5)(4.5, 4.5)(1.5, 4.5)(1.5, 1.5);  
  
// 2つのポリゴンが交わっているか  
const bool result = geometry::intersects(a, b);  
BOOST_ASSERT(result);
```



新たなラムダ式のライブラリ。

通常の関数を部分適用可能な形式にアダプトしたりできる。

関数オブジェクトを返すSTL風アルゴリズムも提供される。

```
namespace ns {  
    int plus(int a, int b) { return a + b; }  
}  
BOOST_PHOENIX_ADAPT_FUNCTION(int, plus, ns::plus, 2)  
  
using namespace boost::phoenix::arg_names;  
  
int result = plus(arg1, 2)(3); // plus関数を部分適用  
std::cout << result << std::endl;
```

5

標準コンテナのBoost実装。

placement insertやmoveなどの最新の仕様が提供される。

```
struct Person {  
    int id;  
    std::string name;  
    Person() {}  
    Person(int id, const std::string& name) : id(id), name(name) {}  
};
```

```
boost::container::vector<Person> v;
```

```
// これまで通りのpush_backだが、一時オブジェクトならmoveされる  
v.push_back({1, "Alice"});
```

```
// 関数内部でコンストラクタを呼び出すplacement insert  
v.emplace_back(2, "Bob");
```



ムーブセマンティクスのC++03実装。  
一時オブジェクトのコストを軽減する。

```
template <class T>
void swap(T& a, T& b)
{
    T tmp(boost::move(a));
    a = boost::move(b);
    b = boost::move(tmp);
}
```

- まとめはとくにありません。
- ここでは差分のみを紹介しましたが、1.48.0までのライブラリをまとめたスライドも別途用意しています。  
全体を知りたい方はそちらを参照してください。

Boostライブラリー一周の旅 1.48.0(all)

<http://www.slideshare.net/faithandbrave/boost-tour-1480-all>