

第4回演習

演習 1

- ▶ 以下のコードを実行して結果を確認せよ。

```
#include <iostream> // prac04-01.cpp
#include <vector>
template<typename T>
void print(const T& a) {
    std::cout << a.size() << ", " << a.capacity() << ": ";
    for (size_t i=0; i<a.size(); i++)
        std::cout << a[i] << " ";
    std::cout << "\n";
}

int main() {
    std::vector<int> a {1,2,3};
    print(a);           // 3, 3: 1 2 3
    a.pop_back(); print(a); // 2, 3: 1 2
    a.pop_back(); print(a); // 1, 3: 1
    a.clear();   print(a); // 0, 3:
}
```

演習 2

- ▶ 以下のコードを実行して結果を確認せよ。

```
#include <iostream> // prac04-02.cpp
#include <vector>
using std::cout;
int main() {
    std::vector a {3,2,3,4,8};
    a.front() = 1;
    a.back() = 5;
    cout << a.front() <<" " << a.back() <<"\n"; // 1 5
    for (auto e : a)
        cout << e <<" ";
    cout <<"\n"; // 1 2 3 4 5
}
```

演習 3

- ▶ 以下のコードを実行して `it` がポインタのように使用できること確認せよ。

```
#include <iostream> // prac04-03.cpp
#include <vector>
using std::cout;
int main() {
    std::vector a {1,2,3,4};
    auto it { a.begin() }; // イテレータ
    cout << *it << " " << *(it + 1) << "\n"; // 1 2
    ++it;
    cout << *it << "\n"; // 2
    cout << a.end() - a.begin() << "\n"; // 4
}
```

演習 4

- ▶ 以下のコードを実行して結果を確認せよ。

```
#include <iostream> // prac04-04.cpp
```

```
#include <vector>
```

```
// 演習 1 と同じ print 関数
```

```
int main() {  
    std::vector a {8,6,5,3,2,1};  
    a.insert(a.begin(), 9);    print(a);  
    a.insert(a.begin()+2, 7);  print(a);  
    a.erase( a.begin()+3 );    print(a);  
}
```

7, 12: 9 8 6 5 3 2 1

8, 12: 9 8 7 6 5 3 2 1

7, 12: 9 8 7 5 3 2 1

演習 5

- ▶ 以下のコードを実行して結果を確認せよ。

```
#include <iostream>    // prac04-05.cpp
#include <vector>
int main() {
    std::vector a {1,2,3,4,5};
    for (auto it=a.begin(); it!=a.end(); ++it) {
        auto e { *it };
        std::cout << e <<" ";
    }
    std::cout <<"\n";    // 1 2 3 4 5

    for (auto e : a) std::cout << e <<" ";
    std::cout <<"\n";    // 1 2 3 4 5
}
```

これ以降の演習

- ▶ これまでと違って講義資料のソースコードの確認が必要です。
- ▶ `myvec.hpp` の中に資料に沿ってメンバ関数を加えていきます。

演習 6

- ▶ 資料 p.18 に `push_back()` と `pop_back()` を加えた `myvec.hpp` を作成し以下の動作を確認せよ。

```
#include <iostream>    // prac04-06.cpp
#include "myvec.hpp"

// 演習 1 と同じ print 関数

int main() {
    MyVec<int> a;
    for (size_t i=0; i<5; i++)
        a.push_back(i+1);
    print(a);           // 5, 8: 1 2 3 4 5
    a.pop_back(); print(a); // 4, 8: 1 2 3 4
    a.pop_back(); print(a); // 3, 8: 1 2 3
    a.clear();          print(a); // 0, 8:

}
```


演習 7

- ▶ 前の演習の myvec.hpp に insert() と begin() を追加して以下を試せ。

```
#include <iostream>    // prac04-07.cpp
#include "myvec.hpp"

// 演習 1 と同じ print 関数

int main() {
    MyVec<int> a;
    a.push_back(2);
    a.push_back(3);
    a.push_back(5);
    a.insert(a.begin(), 1); print(a); // 4, 4: 1 2 3 5
    a.insert(a.begin()+3,4);print(a); // 5, 8: 1 2 3 4 5
}
```

演習 8

- ▶ 前の演習の myvec.hpp に erase() と end() を追加して以下を試せ。

```
#include <iostream>    // proc04-08.cpp
#include "myvec.hpp"

// 演習 1 と同じ print 関数

int main() {
    MyVec<int> a;
    for (size_t i=0; i<6; i++)
        a.push_back(i);
    a.erase(a.begin()+2); print(a); // 5, 8: 0 1 3 4 5
    a.erase(a.end()-3);   print(a); // 4, 8: 0 1 4 5
}
```

演習 9

- ▶ 以下を試してみよ。

```
#include <iostream>    // proc04-09.cpp
#include "myvec.hpp"
// 演習 1 と同じ print 関数
template<typename T>
void test(T& a) {
    for (size_t i=0; i<6; i++) a.push_back(i);
    print(a);
    a.insert(a.begin()+4, 8); print(a);
    a.erase(a.begin()+2);    print(a);
    a.erase(a.end()-3);      print(a);
    a.clear();               print(a);
}
int main(){
    std::vector<int> x; test(x);
    MyVec<int> y;      test(y);
}
```

実行結果：

```
6, 8: 0 1 2 3 4 5
7, 8: 0 1 2 3 8 4 5
6, 8: 0 1 3 8 4 5
5, 8: 0 1 3 4 5
0, 8:
6, 8: 0 1 2 3 4 5
7, 8: 0 1 2 3 8 4 5
6, 8: 0 1 3 8 4 5
5, 8: 0 1 3 4 5
0, 8:
```