

第14回演習

準備

- ▶ 以下のヘッダファイルを演習で用います。

```
// ex14.hpp
#include <iostream>
class T {
    int a;
    void log(std::string m) {
        std::cout << m << " " << &a << "\n";
    }
public:
    T(int b)      :a{b}    { log("ctor(int):      "); }
    T()           :a{1}    { log("ctor(default):"); }
    T(const T& x):a{x.a} { log("ctor(copy):   "); }
    ~T()          { log("dtor:           "); }
    int get()      const { return a; }
    void set(int x) { a = x; }
};
```

演習 1 ex14-1[ab].cpp

- 以下を実行して、ctor と dtor の呼び出し回数をそれぞれ数えよ。
- sum 関数の引数と関数の中の範囲 for 文をそれぞれリファレンスに変更して、ctor と dtor の呼び出し回数をそれぞれ数えよ¹。

```
#include <vector> // ex14-1a.cpp
#include "ex14.hpp"
using std::vector;

int sum(vector<T> x) { // あえて値渡し
    int s {0};
    for (auto e:x) s += e.get(); // コピー
    return s;
}

int main(){
    vector<T> a(10);
    std::cout <<"sum = "<< sum(a) <<"\n";
}
```

¹2 種類のプログラムと実行例を提出。以降の演習も ab ならば同様。

演習 2 ex14-2[ab].cpp

- 以下を実行して、`new T{1}`と `new T[5]{3,2}`がそれぞれどの種類の `ctor` を呼び出しているかを答えよ。
- メモリの解放を指定して、各 `ctor` に対応する `dtor` の呼び出しを確認せよ。

```
#include "ex14.hpp"
using std::cout;
int main() {
    T* p { new T{1} };
    cout << p->get() << "\n";

    T* a { new T[5]{3,2} }; // 初期値指定
    for (int i = 0; i<5; i++)
        cout << a[i].get() << " ";
    cout << "\n";
    // どちらも解放を忘れている。
}
```

演習 3 ex14-3.cpp

- ▶ 前の演習の p と a を `std::unique_ptr<>` に置き換えよ。
- ▶ 配列の初期値指定はできないので、配列を割り当てた後に値を `set()` で設定せよ。

```
ctor(int):      0xba3eb0
1
ctor(default): 0xba42e8
ctor(default): 0xba42ec
ctor(default): 0xba42f0
ctor(default): 0xba42f4
ctor(default): 0xba42f8
3 2 1 1 1
dtor:          0xba42f8
dtor:          0xba42f4
dtor:          0xba42f0
dtor:          0xba42ec
dtor:          0xba42e8
dtor:          0xba3eb0
```

演習 4 ex14-4[ab].cpp

- 以下を実行して、ctor と dtor の呼び出し回数をそれぞれ数えよ。
- 変数 `a` の宣言を `std::vector<T> a(5);` に変更して、`set()` で値を設定する様にせよ。ctor と dtor の呼び出し回数をそれぞれ数えよ。

```
#include <vector>
#include "ex14.hpp"

int main() {
    std::vector<T> a;
    for (int i = 0; i < 5; i++)
        a.push_back(T(i));
}
```

演習 5 myvec-u.hpp, ex14-5.cpp

- ▶ 4 回目の授業資料の myvec.hpp を myvec-u.hpp という名前でコピーし以下を変更せよ。
 - ▶ 3 番目 (capsize 関連) は複数箇所にある。

myvec.hpp	myvec-u.hpp
#include <vector>	#include <memory>
std::vector<T> ar;	std::unique_ptr<T[]> ar;
std::vector<T> n(capsize);	auto n{std::make_unique<T[]>(capsize)};

- ▶ 以下のプログラムを試してみよ。

```
#include <iostream>
#include "myvec-u.hpp"
int main() {
    MyVec<int> myvec;
    for (size_t i = 0; i < 5; i++)
        myvec.push_back(i);
    myvec.erase(3);
    for (size_t i = 0; i < myvec.size(); i++)
        std::cout << myvec[i] << " ";
    std::cout << "\n"; // 0 1 2 4
}
```

演習 6 ex14-6.cpp

unique_ptr を使ったスタックの実装を試してみよ。

```
#include <iostream>
#include <memory>
using Ptr = std::unique_ptr<struct Node>;

struct Node { /* ... */ };
class Stack { /* ... */ };

int main(){
    Stack s;
    s.push(1);
    s.push(2);
    s.push(3);
    std::cout << s.top() << "\n";
    s.pop();
    std::cout << s.top() << "\n";
}
```