

Kontenery

dr inż. Ireneusz Szczęśniak

jesień 2017

1 Zadanie 1

Napisać szablonowy kontener `range`, który będzie generował żądany ciąg liczb całkowitych przy użyciu pętli `for`. To jest przykładowy plik `range.cc`:

```
#include <iostream>
#include "range.hpp"

using namespace std;

int main()
{
    for(auto i: range<int>(1, 11))
        cout << i << endl;
}
```

2 Rozwiązanie zadania

Zawartość pliku `range.hpp`:

```
template <typename T>
struct range_iter
{
    T m_i;

    range_iter(T i): m_i(i)
    {
    }

    range_iter &
    operator++()
    {
        ++m_i;
        return *this;
    }

    T
    operator *() const
    {
        return m_i;
    }

    bool
    operator!=(const range_iter &i)
    {
```

```

        return m_i != *i;
    }
};

template <typename T>
struct range
{
    T m_a;
    T m_b;

    range(T a, T b): m_a(a), m_b(b)
    {
    }

    range_iter<T> begin()
    {
        return range_iter<T>(m_a);
    }

    range_iter<T> end()
    {
        return range_iter<T>(m_b);
    }
};

```

3 Zadanie 2

Napisać klasę **A** tylko do przenoszenia. Stworzyć dwa zbiory obiektów tej klasy. Dodać jeden element do pierwszego zbioru, a potem ten element bez kopiowania do drugiego zbioru.

4 Rozwiązanie zadania

```

#include <iostream>
#include <set>

using namespace std;

struct A
{
    int m_id;

    A(int id): m_id(id)
    {
    }

    A(const A &a) = delete;
    A& operator = (const A &a) = delete;

    A(A &&a): m_id(a.m_id)
    {
        cout << "move-ctor\n";
    }

    A &
    operator = (A &&a)

```

```
{
    m_id = a.m_id;
    cout << "move-assignment\n";
}

bool
operator < (const A &a) const
{
    return m_id < a.m_id;
}
};

int main()
{
    set<A> a, b;
    a.insert(A(1));
    cout << "TEST\n";
    b.insert(a.extract(a.begin()));
    return 0;
}
```