

1. Zadanie

Zaimplementować i przetestować szablon klasy `Table` przechowujący tablicę dynamiczną oraz stowarzyszoną z nią klasę `Iterator` pozwalającą na dostęp do elementów tablicy przechowywanej w klasie `Table`.

Dla klasy `Table` zaimplementować konstruktory, destruktor oraz metody `begin()` i `end()` zwracające odpowiednio iterator do pierwszego elementu i iterator do pierwszego elementu za końcem tablicy.

Iterator powinien zapewniać zestaw operatorów `++` i `--`, operatory `*` i `->` zapewniające interfejs wskaźnikowy oraz operator `!=`.

Jako zadanie dodatkowe - zrobić podobne ćwiczenie dla szablonu klasy `List` przechowującego listę dwukierunkową.

Implementację przetestować następującym kodem:

```
int tab[] = {1, 3, 4, 8};

Table<int> vec(tab, tab + 4);
for(Table<int>::Iterator iter = vec.begin(); iter != vec.end(); ++iter)
    cout << *iter << endl;

Table<int> vec2(vec);
for(Table<int>::Iterator iter = vec2.begin(); iter != vec2.end(); iter++)
{
    *iter = 1;
    cout << *iter << endl;
}

for(Table<int>::Iterator iter = vec.begin(); iter != vec.end(); ++iter)
    cout << *iter << endl;

cout << "Kod dla zadanie dodatkowego" << endl;
List<int> lst(tab, tab + 4);
for(List<int>::Iterator iter = lst.begin(); iter != lst.end(); ++iter)
    cout << *iter << endl;
```