

Programowanie

Ćwiczenia, semestr letni 2024

Michał Suchorowski

4 kwietnia 2024

Spis treści

1 Ćwiczenia 6	1
1.1 Zadanie 0	1
1.2 Zadanie 1	2
1.3 Zadanie 2	2
1.4 Zadanie 3	2
1.5 Zadanie 4	2
1.6 Dodatkowe zadanie 1	2

1 Ćwiczenia 6

<https://cplusplus.com/reference/vector/vector/>

1.1 Zadanie 0

Przetestuj różne metody iteracji po wektorze. (Plik cpp: [wektory.cpp](#).)

```
#include<iostream>
#include<cmath>
#include<cstdlib>
#include<vector>
#include<algorithm>
using namespace std ;

int main(){

    // zapelnienie wektora
    vector<int> wektor={1,2,3,4};

    cout<<"Metoda_1:" << endl;
    // uzycie iteratorow wskazujacych na poczatek i koniec wektora
    for (vector<int>::iterator it = wektor.begin(); it != wektor.end(); ++it) {
        cout << *it << endl;
        (*it)++;
    }

    cout<<"Metoda_2:" << endl;
    // uzycie opcji auto do okreslenia typu iteratora
    for (auto it = wektor.begin(); it != wektor.end(); ++it) {
        cout << *it << endl;
        (*it)++;
    }

    cout<<"Metoda_3:" << endl;
    //uzycie petli zakresowej bez referencji (mozliwosc wypisu ale nie mozna zmienic)
    for (auto element : wektor) {
        cout << element << endl;
        element++; //komenda nie zmienia wartosci w wektorze
    }

    cout<<"Metoda_4:" << endl;
    //uzycie petli zakresowej z referencja (mozliwosc wypisu i zmiany)
    for (auto & element : wektor) {
        cout << element << endl;
        element++;
    }
```

```

cout<<"Metoda_5:" << endl;
//uzycie petli zakresowej z referencja const (mozliwosc wypisu ale brak mozliwosci zmiany)
for (const auto & element : wektor) {
    cout << element << endl;
    // element++; <- powoduje blad kompilacji
}

cout<<"Metoda_6:" << endl;
for (int it = 0; it<wektor.size(); ++it) {
    cout << wektor[it] << "_" << wektor.at(it) << endl;
}
}

```

1.2 Zadanie 1

Napisz program, który zapełni wektor liczbami losowymi z zakresu od 0 do 1000. Następnie znajdź największy element wektora. Napisz własny algorytm, a następnie przetestuj zaimplementowany w bibliotece STL *max_element*. Zainicjalizuj wektor o rozmiarze 0 i dodawaj do niego elementy metodą *push_back*. Wielkość wektora wczytaj używając *cin*.

1.3 Zadanie 2

Przerób program z zadania 1, by zainicjalizowany wektor miał od początku określony rozmiar. Wypróbuj metodę *at* do odnoszenia się do elementów wektora.

1.4 Zadanie 3

Uzupełnij wektor literami alfabetu. Napisz funkcję, która przesuwa całą tablicę o jedną pozycję.

1.5 Zadanie 4

Napisz program, który narysuje szachownice jedynek i zer. Wymiar szachownicy wczytaj funkcją *cin*. W programie ma powinna być funkcja wczytująca puste wektory i zapełniająca je odpowiednimi liczbami oraz druga funkcja wypisująca te szachownice. Zastanów się jaką różnicę będzie robić przekazanie wektora, referencji do wektora, referencji *const* do wektora jako argument funkcji.

1.6 Dodatkowe zadanie 1

Napisz program, który używa algorytmu sortowania bąbelkowego do posortowania losowo zapełnionego wektora. https://pl.wikipedia.org/wiki/Sortowanie_b%C4%85belkowe.