



**СУ "Св. Климент Охридски",
ФМИ – Софтуерно инженерство
Курсов проект по Обектно-ориентирано
програмиране**

Vector

Кристина Недялкова Дякова, Факултетен № 61765

Съдържание

1. Въведение	2
2. Описание на приложените алгоритми	2
3. Описание на програмния код	2
4. Използвани технологии	3

1. Въведение

Целта на проекта е да бъде реализиран шаблонен клас за динамичен масив - Vector, подобен на vector от STL и да бъде показана примерна употреба на реализирания клас.

2. Описание на приложените алгоритми

Не са използвани алгоритми.

3. Описание на програмния код

Кода може да бъде разгледан на <https://github.com/kriss95/OOP-project-Vector.git>

В отделен header файл използвам шаблон за да създам класа Vector и всичките му методи, така че да наподобява максимално vector от STL. В класа съм имплементирала следните методи:

- Конструктор по подразбиране, за запълване (fill constructor), за копиране
- Vector& operator=(const Vector& v)
- Деструктор- в които изтривам всички показатели
- void push_back(const T& value) - добавя елемент в края на динамичния масив
- void pop_back() - изтрива елемент в края на динамичния масив
- void resize(int newSize) - променя размера на динамичния масив, така че да стане newSize
- T& back() - връща стойността на елемента в края на Vector
- T& front() - връща стойността на елемента в началото на Vector
- void insert(int index, T value) – вмъква елемент със стойност value на позиция index
- void erase(int index) – изтрива елемент на позиция index
- T& operator[](int index) - връща псевдоним към елемент по подаден индекс
- int size() - връща броя на елементите
- int capacity() – връща капацитета на динамичния масив
- bool empty() - проверява дали има елементи
- void clear() - изтрива цялото съдържание на Vector

Освен тези методи в класа Vector съм създавала и подклас iterator в него, който наподобява класа iterator на vector от STL. В него са реализирани следните методи:

- Конструктор по подразбиране, за запълване (fill constructor), за копиране
- iterator& operator=(const iterator& it)
- T& operator*()
- Iterator& operator ++()
- Iterator& operator --()

- `Iterator operator+(const int& value) const`
- `Iterator operator-(const int& value) const`
- `bool operator!=(const iterator& it)`

Използвам съм `iterator`, за да създам следните методи на класа `Vector`:

- `iterator begin()`-връща `iterator` сочещ към първия елемент на `Vector`
- `iterator end()`-`iterator` сочещ към една позиция след края
- `void assign(int n, T value)`-изтрива съдържанието на вектора и запълва `n` негови позиции с `value`
- `void assign(T* p1, T* p2)` изтрива съдържанието на вектора и го запълва с стойностите на масив `T p[]`, където `*p1` и `*p2` са съответно показатели от коя до коя позиция да се копират елементите
- `void assign(iterator it1, iterator it2)` изтрива съдържанието на вектора и го запълва с стойностите на друг вектор, където `it1` и `it2` са съответно итератори от коя до коя позиция да се копират елементите

4. Използвани технологии

Език за програмиране: C++

Среда за разработка: Microsoft Visual Studio Professional 2013.