Biblioteka STL - wektor i lista

M. Majchrzycki

19 maja 2013

1 Wprowadzenie

Biblioteka STL (ang. *Standard Template Library*) to jedna ze standardowych bibliotek kompilatora C++. Zawiera ona definicje różnych struktur danych, algorytmów oraz iteratorów zaimplementowanych w sposób generyczny. Oznacza to iż elementy biblioteki STL można skonkretyzować dla dowolnego typu danych (o ile spełnia on pewne minimalne wymagania). Biblioteka STL swoją budowę opiera na szablonach klas.

1.1 Szablony klas

Język C++ wprowadził pojęcie szablonu klasy. Jest to struktura programistyczna, która pozwala na uogólnienie budowy klasy dla różnych typów danych. Na przykład: możemy stworzyć szablon klasy `stos` który można skonkretyzować dla różnych typów danych takich jak `int`, `float` czy typ danych użytkownika.

Przykład szablonu klasy przedstawia listing 1.

```
1  template <class T>
2  class MyClass
3  {
4    public:
5        MyClass(T init_arg);
6        T get();
7        void set(T arg);
8      private:
9      T storage;
10     int temp;
11 };
12
13 template <class T>
14 MyClass(T init_arg)
15 {
16   storage = init_arg;
17 }
18
19 template <class T>
20 T MyClass<T>::get()
21 {
22   return storage;
23 }
24
25 template <class T>
26 void MyClass<T>::set(T arg)
27 {
28   storage = arg;
29   temp++;
```

```
30 }
```

Listing 1: Przykład szablonu klasy

Wykorzystanie szablonu klasy w programie do utworzenia obiektu przedstawia listing 2.

```
1  int main()
2  {
3    int value;
4    MyClass<int> a(22);
5
6    a.set(6);
7    value = a.get();
8    cout << value << endl;
9
10   MojaKlasa<float> b(1.2);
11   cout << b.get() << endl;
12
13   return 0;
14 }
```

Listing 2: Przykład wykorzystania szablonu

1.2 std::vector

Kontener `vector` reprezentuje strukturę podobną do tablicy jednowymiarowej, której rozmiar alokowany jest dynamicznie. Oznacza to że tworząc i dodając do wektora kolejne elementy nie musimy dbać o przydział pamięci gdyż klasa ta dokonuje tego w sposób automatyczny. Kontener wektor posiada przeładowany operator [], który pozwala na swobodny dostęp do poszczególnych jego elementów.

Szczegółowa dokumentacja klasy `vector` dostępna jest pod adresem:
http://www.cplusplus.com/reference/vector/vector/

1.3 std::list

Kontener `list` implementuje dwukierunkową listę. Struktura ta opiera się na zastosowaniu węzłów, które zawierają dwa wskaźniki: do kolejnego elementu listy i do poprzedniego elementu. Pamięć dla listy alokowana jest dynamicznie w trakcie dodania lub usunięcia pojedynczego węzła.

Szczegółowa dokumentacja klasy `list` dostępna jest pod adresem:
http://www.cplusplus.com/reference/list/list/

W odróżnieniu od kontenera `vector`, lista nie stosuje tablicy do przechowywania danych. Ponadto lista gwarantuje stały czas wykonania operacji dodania elementu, co w przypadku wektora nie jest możliwe. Natomiast wektor pozwala no swobodny dostęp do dowolnego elementu, podczas gdy lista wymaga dostępu sekwencyjnego.

2 Zadania

1. Napisz program który wczyta dowolny plik tekstowy wskazany przez użytkownika w taki sposób aby każdy wyraz umieszczony był na osobnej pozycji kontenera lista lub wektor. Wczytania dokonaj z pominięciem znaków interpunkcyjnych oraz wyrażeń liczbowych.

2. Napisz funkcję, która przyjmie jako argument listę lub wektor stworzoną w punkcie 1, a następnie zapisze w pliku tekstowym dwa rodzaje statystyk:

   - liczbę wyrazów o określonej długości (wyrazy dłuższe niż 10 literowe traktuj jako tej samej długości)
   - ilość wystąpień poszczególnych wyrazów