

Zaawansowane C++

Lista 1: Szablony

Zadanie 1

Zdefiniuj szablon funkcji `my_max` (`T a`, `T b`, `C comp`) parametryzowany przez dwa argumenty szablonu: `T` zawierający informację o typie danych porównywanych elementów oraz `C` opisujący typ komperatora, czyli dwuargumentowej funkcji zwracającej większy z elementów. Przekaż komperator jako wyrażenie lambda porównujące liczby przy użyciu standardowego operatora `<`. Przetestuj dla kilku różnych typów danych.

Zadanie 2

Zdefiniuj szablon funkcji `insertion_sort` przyjmującej jako argument wektor `std::vector<T>` elementów i sortującej elementy tego wektora za pomocą algorytmu *insertion sort*. Jaką złożoność ma ten algorytm?

Zadanie 3

a) Zdefiniuj przestrzeń nazw `cpplab`.

b) W powyższej przestrzeni nazw stwórz szablon klasy `cpplab::vector` z typem danych podanym jako argument szablonu. Klasa powinna wspierać dynamiczne dodawanie i usuwanie elementów oraz zmianę długości wektora. Domyślny konstruktor nie powinien alokować żadnej pamięci. Zaimplementuj podstawowe funkcjonalności tj. operator `[]` do pobrania i modyfikacji poszczególnych składowych wektora. Dodatkowo klasa powinna mieć zdefiniowany typ `value_type`, przechowujący informację o typie elementów wektora.

c) Dla klasy `cpplab::vector` dopisz operator mnożenia skalarnego tak, aby dało się policzyć iloczyn skalarny wektorów `cpplab::vector` oraz `std::vector` w dowolnej konfiguracji argumentów i typu składowych wektorów. Operator mnożenia skalarnego powinien być zdefiniowany **jednym** szablonem.

Materialy pomocnicze:

- Podstawy szablonów w C++20:
<https://www.youtube.com/watch?v=HqsEHG0QJXU>
- Namespaces, czyli przestrzenie nazw:
<https://www.youtube.com/watch?v=ts1Eek5w7ZA&t=2s>
- Algorytm *insertion sort*:
<https://www.youtube.com/watch?v=eTvQIbB-AuE>