

Zaawansowane C++

Lista 1: Szablony

Zadanie 1

Stwórz następujące szablony:

- Szablon funkcji `add` przyjmującej dwa argumenty i zwracającej wynik dodawania.
- Szablon funkcji `alias` przyjmującej dwa argumenty oraz operację podaną jako trzeci argument. Szablon funkcji ma zwracać wynik operacji wykonanej na przekazanych argumentach.

W obu przypadkach jawnie zdefiniuj typ zwracanego argumentu używając specyfikatora `decltype`. Przetestuj działanie funkcji dla różnych typów argumentów np. numerycznych i `std::string`, jak i różnych operacji w przypadku szablonu `alias`. Jedną z operacji zdefiniuj za pomocą wyrażenia lambda implementującego np. funkcję `max(a,b)`.

Wskazówka: może się przydać użycie trailing return type.

Zadanie 2

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. operatory do pobrania i modyfikacji poszczególnych składowych wektora. Dodatkowo klasa powinna mieć zdefiniowany typ `value_type`.

c) Dla klasy `cpplab::vector` dopisz operator mnożenia skalarnego tak, aby dało się policzyć iloczyny skalarne wektorów `cpplab::vector` oraz `std::vector` w dowolnej konfiguracji argumentów i typu składowych wektorów.

W przypadku różnej lub zerowej liczby składowych argumentów operatora, powinien zostać zwrócony wyjątek z opisem błędu.

Materiały pomocnicze:

- Podstawy szablonów w C++20:
<https://www.youtube.com/watch?v=HqsEHGOQJXU>

- Namespaces, czyli przestrzenie nazw:
<https://www.youtube.com/watch?v=ts1Eek5w7ZA&t=2s>