

Zaawansowane C++

Lista 2: Szablony

Zadanie 1 Sortowanie naturalne

Do szablonu funkcji `insertion_sort` z zadania 2 z listy 1 dopisz specjalizację dla typu `std::string` sortującą nazwy za pomocą *porządku naturalnego*, a nie leksygraficznego. Jak zwykle w funkcji `main()` przetestuj działanie swojej funkcji.

Zadanie 2 Metaprogramowanie

Wykorzystując częściową specjalizację szablonu klasy napisz program, który oblicza *rekurencyjnie* silnię w czasie kompilacji. Argument silni przekaz jako argument szablonu.

Zadanie 3 variadic templates

Napisz, **bez** użycia fold expression, szablon funkcji `print_all` dodający dowolną, niezerową liczbę argumentów dowolnego typu np.

```
print_all(1, 1.0, 1.0f, "Hello")
print_all(1, 2.0)
```

Wskazówka: zaprzyjaźnij się z operatorem `sizeof...` może się przydać.

Materiały pomocnicze i uzupełniające:

- SFINAE and `std::enable_if`
<https://www.youtube.com/watch?v=dR64GQb4AGo&t=46s>
- Zawędrowaliśmy nawet na stackoverflow <https://stackoverflow.com/questions/70100750/in-what-way-const-is-interfering-with-marking-a-more-specialized-template-when-u/70103825#70103825>
- Szablony są zupełne w sensie Turinga
<https://matt.might.net/articles/c++-template-meta-programming-with-lambda-calculus/>