

# Zaawansowane C++

## Lista 3: Koncepty, wskaźniki i smart wskaźniki

### Zadanie 1

To zadanie wymaga flagi kompilatora `-std=c++20` i użycia `concept` oraz `requires`.

Operator mnożenia skalarnego z zadania 2c z listy 1 powinien być napisany jako *jeden* szablon. Stwórz koncept, który sprawdzi czy przekazany do szablonu operator `*` typ spełnia minimalne wymagania, aby iloczyn skalarny dał się wykonać.

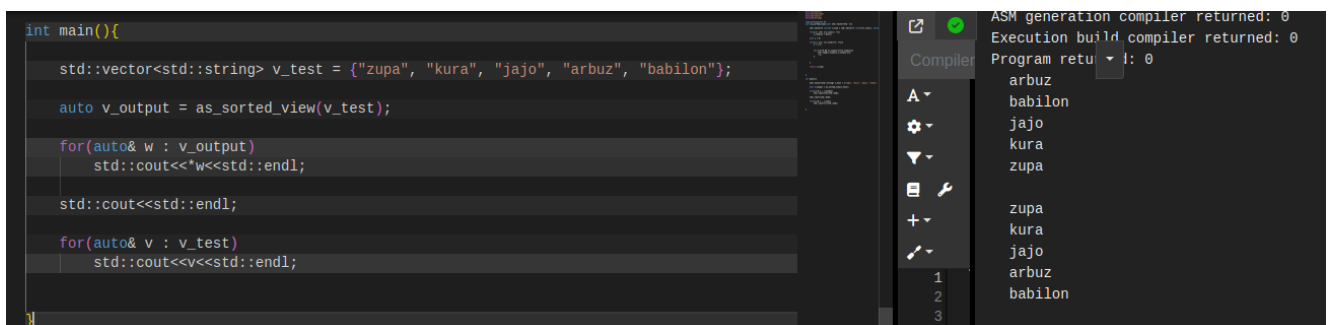
Przetestuj działanie dla kilku różnych typów m.in. `std::string` oraz dla typu, dla którego operator `*` nie zadziała.

### Zadanie 2

Napisz szablon funkcji `as_sorted_view`, przyjmujący jako argument `std::vector<T>` elementów dowolnego typu `T`, który wyświetli jego elementy w posortowanej kolejności nie zmieniając równocześnie elementów oryginalnego wektora.

### Zadanie 3

Używając smart pointera `std::unique_ptr` napisz szablon funkcji `fordawrd_list` implementujący listę pojedynczo związaną. Dopisz metodę `reverse()` odwracającą listę pojedynczo związaną. Przetestuj swoją klasą.



```
int main(){
    std::vector<std::string> v_test = {"zupa", "kura", "jajo", "arbuz", "babilon"};
    auto v_output = as_sorted_view(v_test);
    for(auto& w : v_output)
        std::cout<<"w<<std::endl;
    std::cout<<std::endl;
    for(auto& v : v_test)
        std::cout<<v<<std::endl;
}
```

Compiler output:

```
arbuz
babilon
jajo
kura
zupa

zupa
kura
jajo
arbuz
babilon
```

Rysunek 1: Przykład użycia funkcji `as_sorted_view` (zad. 2).

### **Materiały pomocnicze:**

- Back to basics cppcon21 o konceptach:  
[https://www.youtube.com/watch?v=\\_FoXWnrGuNU&t=1582s](https://www.youtube.com/watch?v=_FoXWnrGuNU&t=1582s)
- Od SFINAE do concepts:  
<https://www.youtube.com/watch?v=dR64GQb4AGo>
- Smart pointery (będzie więcej w przyszłości)  
<https://www.youtube.com/watch?v=UOB7-B2MfwA&t=422s>