

Fold Expressions

Omówienie i przykłady użycia

Autor
Łukasz Jaremek
2023.05

Czym jest Fold Expressions

Fold Expressions to funkcjonalność dodana do standardu w wersji C++17, która pozwala na łączenie wyrażeń za pomocą operatorów:

- dodawanie +
- odejmowanie -
- mnożenie *
- dzielenie /
- równość ==
- albo ||
- oraz &&
- ...

Jest ona najczęściej wykorzystywana w szablonach, gdy mamy do czynienia z dużą liczbą argumentów – dzięki Fold Expressions nie musimy obsługiwać argumentów oddzielnie, tylko możemy to zrobić za jednym razem co pozwala zredukować kod do jednej czytelnej linii.

Fundamentalne przykłady użycia Fold Expressions

Wykorzystanie +

Funkcja sumująca wszystkie liczby, które do niej prześlemy.
Nie ma limitu przekazywanych liczb.

```
template<typename... Args>
auto sum_numbers(Args... args) {
    return (args + ...);
}
```

Wykorzystanie *

Funkcja zwracająca iloczyn wszystkich podanych liczb.

```
template<typename... Args>
auto product_numbers(Args... args) {
    return (args * ...);
}
```

Wykorzystanie ||

Funkcja zwraca *true*, gdy choć jeden argument ma wartość *true*.

```
template<typename... Args>
bool any(Args... args) {
    return (args || ...);
}
```

Wykorzystanie &&

Funkcja zwraca *false*, gdy choć jedna wartość wynosi *false*.

```
template<typename... Args>
bool all(Args... args) {
    return (args && ...);
}
```

Przykłady użycia Fold Expressions

Liczenie średniej

```
template <typename... Args>
double avg(Args... args) {
    return static_cast<double>((args + ...)) / sizeof...(args);
}
```

Liczenie średniej kwadratowej

```
template <typename... Args>
double mean_square(Args... args) {
    return std::sqrt(avg(args * args...));
}
```

Sprawdzanie równości argumentów

```
template<typename... Args>
bool equal(Args... args) {
    return ((args == args) && ...);
}
```

Liczenie sumy liczb nieparzystych

```
template<typename... Args>
int sum_even(Args... args) {
    return (0 + ... + (args % 2 == 0 ? args : 0));
}
```

Wyświetlanie zestawu argumentów

```
template<typename... Args>
void print(Args... args) {
    ((std::cout << args << " "), ...);
    std::cout << std::endl;
}
```

To, co się rzuca w oczy w każdej funkcji to jej liczba linii, która w naszych przypadkach zawsze wynosi 1. Dzięki Fold Expression, jesteśmy w stanie skrócić często wielolinijkowe, złożone funkcje do jednej linii.