

0.1 Definicja konceptu

Rozróżniamy dwa rodzaje konceptów:

Zmienna konceptowa - jest typem czasu kompilacji i nie niesie za sobą żadnych kosztów czasu wykonania.

Najprostsza forma zmiennej konceptowej:

```
template<template T>
concept bool zmienna_konceptowa = true;
```

Taka zmienna nie może być zadeklarowana z jakimkolwiek innym typem niż `bool` oraz bez inicjalizatora. Błąd pojawi się też, gdy inicjalizatorem nie będzie ograniczone wyrażenie.

Przykład użycia:

```
template<template T>
requires zmienna_konceptowa<T>
void f(T t){
    std::cout << t << "\n";
}
```

Funkcja konceptowa - wygląda i zachowuje się jak zwykła funkcja.

```
template<template T>
concept bool funkcja_konceptowa(){
    return true;
}
```

Funkcja konceptowa nie może:

- być zadeklarowana z żadnym specyfikatorem funkcji w deklaracji
- zwracać żadnego innego typu niż `bool`
- mieć żadnych elementów w liście parametrów
- mieć innego ciała niż `{ return E; }`, gdzie `E` to wyrażenie ograniczone

Przykład użycia:

```
template<template T>
requires funkcja_konceptowa<T>()
void f(T t){
    std::cout << t << "\n";
}
```