

Wprowadzenie

W 1987 próbowano projektować szablony z odpowiednimi interfejsami. Chciano by szablony:

- były w pełni ogólne i wyraziste
- by nie wykorzystywały większych zasobów w porównaniu do kodowania ręcznego
- by miały dobrze określone interfejsy

Długo nie dało się osiągnąć tych trzech rzeczy, ale za to osiągnięto:

- *kompletność Turinga*¹
- lepszą wydajność (w porównaniu do kodu pisanego ręcznie)
- kiepskie interfejsy (praktycznie *typowanie kaczkowe czasu kompilacji*)²

Brak dobrze określonych interfejsów prowadzi do spektakularnie złych wiadomości błędów. Dwie pozostałe właściwości uczyniły z szablonów sukces.

Rozwiązanie problemu specyfikacji interfejsu zostało, przez Alexa Stepanova nazwane **konceptami**. **Koncept** to zbiór wymagań argumentów szablonu. Można też go nazwać systemem typów dla szablonów, który obiecuje znacząco ulepszyć diagnostyki błędów i zwiększyć siłę ekspresji, taką jak przeciążanie oparte na konceptach oraz częściowa specjalizacja szablonu funkcji.

Koncepty (*The Concepts TS*³) zostały opublikowane i zaimplementowane w wersji 6.1 kompilatora GCC w kwietniu 2016 roku. Fundamentalnie to predykaty czasu kompilacji typów i wartości. Mogą być łączone zwykłymi operatorami logicznymi (&&, ||, !)

¹(ang. Turing Completeness) umiejętność do rozwiązania każdego zadania, czyli udzielenie odpowiedzi na każde zadanie. Program, który jest kompletny według Turinga może być wykorzystany do symulacji jakiegokolwiek 1-taśmowej maszyny Turinga

²(ang. duck typing) rozpoznanie typu obiektu, nie na podstawie deklaracji, ale przez badanie metod udostępnionych przez obiekt

³(ang. The Concepts Technical Specification) Specyfikacja techniczna konceptów