

Специализация и перегрузка шаблонов

Специализация шаблона функции позволяет генерировать другой код для некоторых типов. В C++ есть два вида специализации шаблона: явная или полная, и частичная.

Явная конкретизация создает конкретный экземпляр обобщенного кода, в котором вместо обобщенных типов поставлены конкретные.

Явная специализация, в отличие от явной конкретизации, создает экземпляр функции с тем же именем, но под именем реализации.

```
template <typename T>
T do_smthg(T x) {
    return(++x);
}
template <>
double do_smthg<double>(double x) {return(x/2);}
```


do $\text{Strg}(3)$: // 4
do $\text{Strg}(3.0)$: // 5

Перегрузка шаблонов функций

Если существуют шаблоны ф-й, которая почти
идеально соответствует функциям аргументам
то она и выбирается. Если такой ф-и нет -
компилятор пытается конкретизировать
все шаблоны ф-й с теми же типами, что и
у вызывающей

Понимая шаблоны ф-й, можно быть
определено несколько шаблонов ф-й
с теми же типами и количеством аргументов.
Все они создадут свои кандидаты.