

## - Что такое специализация шаблона функции?

Специальная версия шаблона функции, со своим телом функции

## - Для чего нужна специализация шаблона?

Для разной обработки разных типов.

## - Чем конкретизация шаблона отличается от специализации, приведите пример

Специализация создает свое особое поведение шаблона для определенных типов. В то время как конкретизация лишь заменяет параметр шаблона конкретным типом.

```
template <class T>
T do_something(T x){
    return (+x);
}

template<>
double do_something<double>(double x){
    return (x/2);
}

void main(){
    auto i1 = do_something(3); // 4, конкретизация
    auto i2 = do_something(3.0); // 1.5, специализация
}
```

## - Что такое перегрузка нешаблонных функций

Процесс замены шаблона конкретным экземпляром функции, у которой известны типы.

## - Как определяется множество перегруженных функций с учетом шаблонных?

Если существует нешаблонная функция, которая почти идеально соответствует фактическим аргументам, то она и выбирается. Если такой функции нет – компилятор пытается конкретизировать все шаблоны функций с тем же именем, что у вызываемой, стремясь получить почти идеальное соответствие, для чего применяет выведение аргументов шаблона. Если был конкретизирован ровно один шаблон, то вызывается функция, получающаяся в результате этой конкретизации.

## - Что означает понятие «более специфичный шаблон», как это работает при выборе перегрузки шаблонной функции?

Более специфичный означает идеально точно соответствующий фактическому аргументу. Если более специфичный шаблон можно использовать для порождения функции, являющейся допустимым перегруженным вариантом, то он и будет выбран. В противном случае мы вынуждены обратиться к более общему шаблону.

## - В чем отличие разрешения перегруженных шаблонов от нешаблонных функций?

В отличие нешаблонных функций в перегруженных шаблонах предпочтение отдаётся варианту без преобразования. Т.е. в процессе разрешения может быть выбран перегруженный шаблон а не шаблонная функция.