**Подготовил материал: Безмен Николай**

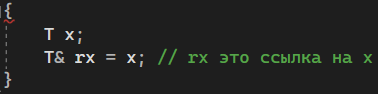
**Источники**:

1. Лекции по С++ Казанцевой О.Г., ссылка на документ: <https://ru.cppreference.com/w/cpp/language/reference>
2. Страуструп, Б. Программирование: принципы и практика с использованием С++, 2-е изд.: Пер. с англ. - М . : ООО "И . Д. Вильяме", 2016. Глава 6
3. <https://metanit.com/cpp/tutorial/2.14.php>

**Вопрос 25:**

***Ссылки:***

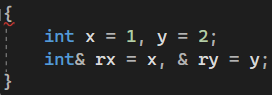
*В простейшем случае ссылка определяется так: если T некоторый тип и есть переменная типа T, то переменная типа T со спецификатором & будет ссылкой на эту переменную, если она инициализирована этой переменной.*



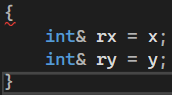
После этого rx можно использовать в любом контексте вместо x, то есть rx становится псевдонимом x.

Инициализация ссылки обязательна, нулевые ссылки (ссылки на «ничто») не поддерживаются. Изменить переменную, на которую ссылается ссылка, невозможно — связь между ссылкой и переменной «до гробовой доски». Таким образом, ссылка является константной сущностью, хотя формально ссылочный тип не является константным.

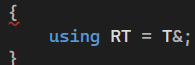
В одной инструкции можно определить несколько ссылок, спецификатор & должен быть у каждой из них.



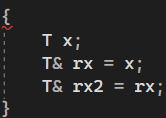
Последняя инструкция эквивалентна следующим двум инструкциям:



Имя типа со спецификатором & будет называться ссылочным типом. Можно объявить псевдоним для ссылочного типа.

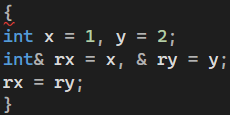


5.1. Можно определить копию ссылки.



После этого на переменную x будут ссылаться две ссылки. Других собственных операций ссылка не поддерживает, все операторы, примененные к ссылке, на самом деле применяются к переменной, на которую она ссылается. Это касается и таких операторов, как = (присваивание), & (получение адреса), sizeof, typeid. Но вот спецификатор decltype, если его применить к ссылке, дает ссылочный тип.

Остановимся подробнее на присваивании. Присваивание ссылок означает присваивание переменных, на которые ссылки ссылаются. Естественно, что тип этих переменных должен поддерживать присваивание.



Последняя инструкция эквивалентна следующей:

