**Пространство имён** ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *namespace*) — некоторое [множество](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%BE), под которым подразумевается модель, [абстрактное](https://ru.wikipedia.org/wiki/%D0%90%D0%B1%D1%81%D1%82%D1%80%D0%B0%D0%BA%D1%86%D0%B8%D1%8F) [хранилище](https://ru.wikipedia.org/wiki/%D0%A5%D1%80%D0%B0%D0%BD%D0%B8%D0%BB%D0%B8%D1%89%D0%B5_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) или окружение, созданное для [логической](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B8%D0%BA%D0%B0) группировки уникальных [идентификаторов](https://ru.wikipedia.org/wiki/%D0%98%D0%B4%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%82%D0%BE%D1%80) (то есть имён).

Идентификатор, определённый в пространстве имён, ассоциируется с этим [пространством](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%BE). Один и тот же идентификатор может быть независимо определён в нескольких пространствах. Таким образом, значение, связанное с идентификатором, определённым в одном пространстве имён, может иметь (или не иметь) такое же значение, как и такой же идентификатор, определённый в другом пространстве. Языки с поддержкой пространств имён определяют [правила](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C), указывающие, к какому пространству имён принадлежит идентификатор (то есть его определение).

Что такое ссылочный тип данных, чем он отличается от указателя

1. Указатель может быть переназначен. Ссылка не может быть повторно привязана и должна быть привязана при инициализации.
2. Переменная указателя имеет свою собственную идентичность: отдельный видимый адрес памяти, который может быть взят с помощью унарного оператора & , и определенный объем пространства, который может быть измерен с помощью оператора sizeof . Использование этих операторов в ссылке возвращает значение, соответствующее тому, к чему привязана ссылка; собственный адрес и размер ссылки невидимы. Поскольку ссылка предполагает идентичность исходной переменной таким образом, удобно думать о ссылке как о другом имени для той же переменной.
3. Вы можете иметь произвольно вложенные указатели на указатели, предлагающие дополнительные уровни косвенности. Ссылки предлагают только один уровень косвенности.
4. Указателю может быть присвоено nullptr , в то время как ссылка должна быть привязана к существующему объекту. Если вы достаточно постараетесь, вы можете привязать ссылку к nullptr , но это не [определено](https://coderoad.ru/2397984/) и не будет вести себя последовательно. Однако вы можете иметь ссылку на указатель, значение которого равно nullptr .
5. Указатели могут перебирать массив; вы можете использовать ++ для перехода к следующему элементу, на который указывает указатель, и + 4 для перехода к 5-му элементу. Это не имеет значения, какого размера объект, на который указывает указатель.
6. Указатель должен быть разыменован с помощью \* , чтобы получить доступ к ячейке памяти, на которую он указывает, в то время как ссылка может использоваться напрямую. Указатель на класс/структуру использует -> для доступа к его членам, в то время как ссылка использует . .
7. Ссылки не могут быть помещены в массив, в то время как указатели могут быть
8. Ссылки Const могут быть привязаны к временным объектам. Указатели не могут (не без некоторой косвенности). Это делает const & более удобным для использования в списках аргументов и так далее.