Наследование классов – механизм, позволяющий создавать классы (говорят подклассы) на основе других классов (называемых базовыми или суперклассами). Подклассы в таком случае "наследуют" структуру базовых классов, то есть получают возможность использовать все, что определено в базовом классе.

В 1987 году Барбара Лисков сформулировала принцип подстановки (Liskov Substitution Principle – LSP), следование которому позволяет правильно строить иерархии типов:

Пусть q(x) является свойством, верным относительно объектов x некоторого типа T. Тогда q(y) также должно быть верным для объектов y типа S, где S является подтипом типа T.

Звучит математично. Многие разработчики пытались переформулировать это правило так, чтобы оно было интуитивно понятным. Самая простая формулировка звучит так:

Функции, которые используют базовый тип, должны иметь возможность использовать подтипы базового типа, не зная об этом.

Существует несколько правил, которые надо учитывать при работе с типами:

* Предусловия не могут быть усилены в подклассе
* Постусловия не могут быть ослаблены в подклассе
* Исторические ограничения

Предусловия – это ограничения на входные данные, а постусловия – на выходные. Причем в силу ограничений систем типов, многие из таких условий невозможно описать на уровне сигнатуры. Их либо придется описывать просто текстом, либо добавлять проверки в код ([проектирование по контракту](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%82%D1%80%D0%B0%D0%BA%D1%82%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)).

И последнее, исторические ограничения. Подтипы (в случае JS — классы-наследники) не могут добавлять новые методы для изменения (мутации) данных базового типа (в случае JS-класса). Способы изменения свойств, определенных в базовом типе, определяются этим типом.