

INTERFACES

Интерфейс - это еще один тип в TS, который позволяет синтаксически записать шаблон того **объекта**, который будет создан

Интерфейс создается при помощи ключевого слова **interface**, имени с большой буквы и раскрытия фигурных скобок:

```
1 interface Config {
2     protocol: "http" | "https";
3     port: 3000 | 3001;
4 }
```

В свойствах объекта может быть что угодно: литералы, другие типы и тд. При замене type на interface в коде ничего не сломается. Они действительно очень похожи. Отличия в логике использования будут рассмотрены в следующем уроке.

Обычно имена интерфейсов пишут в виде венгерской нотации, начиная с буквы I и дальше camelCase. Это необязательное правило, зависит от проекта и компании (IConfig, IMainSlider, IAdminUser ...)

Префикс	Сокращение от	Смысл	Пример
s	string	строка	sClientName
n, i	int	целочисленная переменная	nSize, iSize
b	boolean	булева переменная	bIsEmpty
a	array	массив	aDimensions
p	pointer	указатель	pBox
r	reference	ссылка	rBoxes
C	class	класс	CString
T	type	тип	TObject
I	interface	интерфейс	IDispatch
v	void	отсутствие типа	vReserved

ОБЪЕДИНЕНИЕ ИНТЕРФЕЙСОВ

Для комбинации нескольких интерфейсов и получения нового используется ключевое слово `extends`:

```
1 interface Config {
2     protocol: "http" | "https";
3     port: 3000 | 3001;
4 }
5
6 interface Role {
7     role: string;
8 }
9
10 interface ConfigWithRole extends Config, Role {
11     // Тут могут быть новые свойства или пустые скобки
12 }
13
14 const serverConfig: ConfigWithRole = {
15     protocol: "https",
16     port: 3001,
17     role: "admin"
18 };
```

Полученный интерфейс будет иметь все свойства указанных + можно добавлять новые внутри фигурных скобок

INDEX SIGNATURES

Если вы не знаете сколько свойств будет в объекте, но знаете, в каком виде они все будут, то можно использовать специальный синтаксис:

```
1 interface Styles {
2     [key: string]: string;
3 }
4
5 const styles: Styles = {
6     position: "absolute",
7     top: "20px",
8     left: "50px",
9 };
```

В документации он называется Index Signatures,
в реальной жизни слышал только “индексная сигнатура”

Теперь все объекты, аннотированные этим интерфейсом, должны соблюдать структуру: свойство только строка, его значение - тоже только строка