

31 | 函数与类的泛型声明

周爱民 (Aimingoo)

目录

1 函数类型的泛型声明（简单复习）

2 实体：泛型函数与泛型类

3 类型：带泛型参数的函数类型

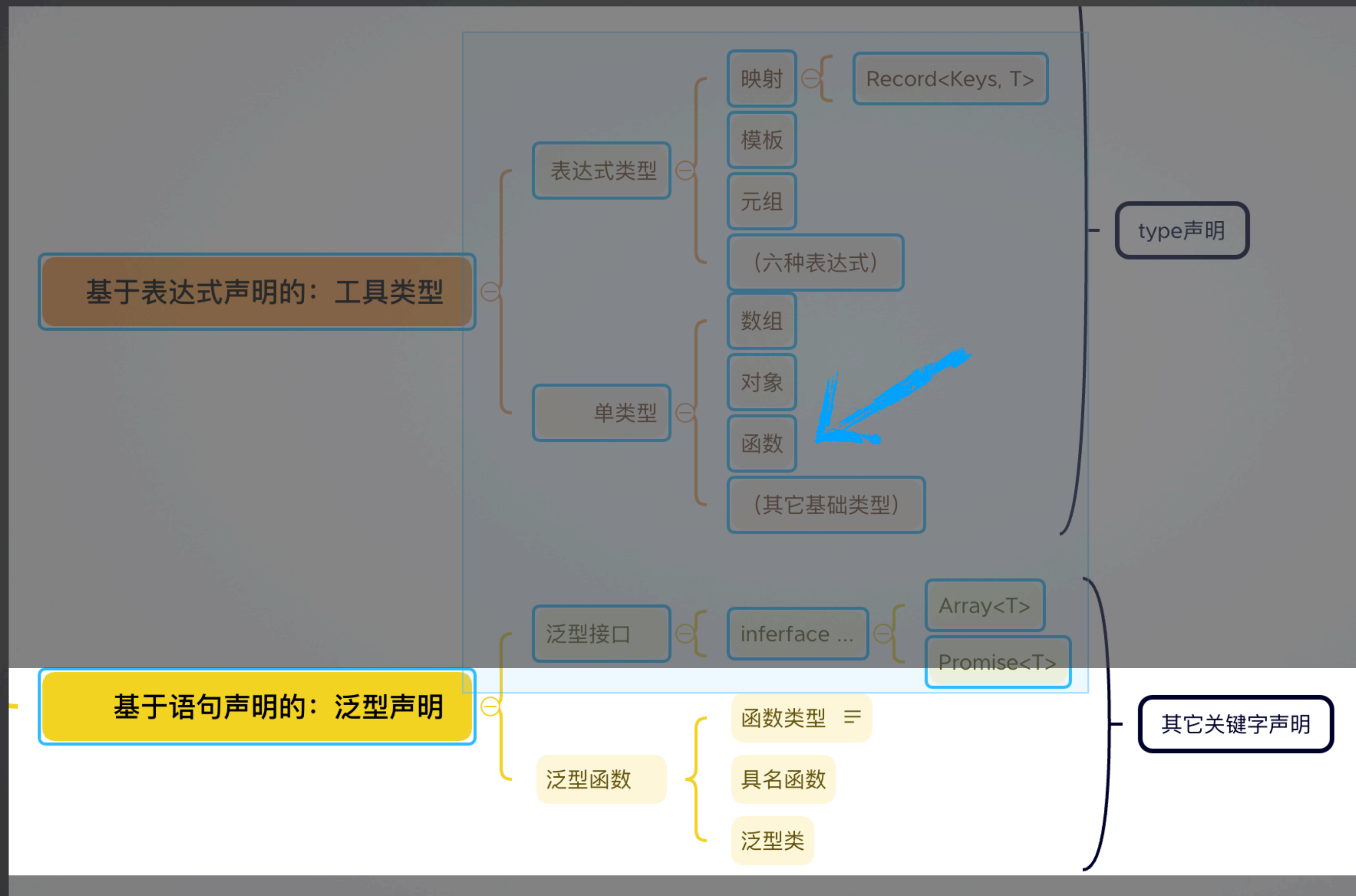
4 有两种实例化的方法

5 总结

与函数相关的泛型声明



与函数相关的泛型声明



Declaration Type	Namespace <small>名字空间</small>	Type <small>类型</small>	Value <small>值</small>
Namespace	X		X ²
Class		X	X
Enum		X	X
Interface		X	
Type Alias		X	
Function	X ¹		X
Variable			X

<https://typescriptlang.org/docs/handbook/declaration-merging.html>

JS中的泛型化（在JS中支持泛型编程）

泛型函数与泛型类

1. 具名的 (V1)

- ▶ `function foo<T> (a: T): T { return a }`
- ▶ `class MyClass<T> { a: T }`

2. 匿名的 (V2)

- ▶ `let foo = <T> (a: T): T { return a }`
- ▶ `let MyClass = class <T> { a: T }`

3. 带泛型参数的函数类型 (`T = typeof V`)

- ▶ `type Foo = typeof foo;`
- ▶ `type MyClassConstructor = typeof MyClass;`
- ▶ 也可以如下直接声明:
 - `type Foo2 = <T> (a: T) => T; // 或者使用调用符名`
 - `type MyClassConstructor = <T> { new <T>(): MyClassPrototype }; // 使用构造签名 (没有类声明语法)`

总结

1. 函数类型的“泛型声明”：`type Foo<T> = (a: T): T;`

- ▶ 跟所有“泛型声明”一样，`Foo<T>`返回的类型跟它声明的类型一致，是后者的一个实例。这个过程称为实例化。

2. 泛型函数与泛型类（Generic Functions/Classes）

- ▶ 直接在js中声明成“带泛型参数的函数或类”。注意它的声明结果是“函数/类”这种实体，而不是类型。
- ▶ 泛型函数或类（例如 `foo`）在声明时，同时会有对应的类型，可以使用 `Foo2 = typeof foo` 来获取。

3. 带泛型参数的函数类型：`type Foo = <T>(a: T): T;`

- ▶ 对于泛型函数（Generic Functions/Generic Classes）来说，上述 `Foo2` 就是“带泛型参数的函数或类”
- ▶ 带泛型参数的函数或类是通过构造、调用或显式声明来实例化的。

作业

>1. 请解释如下声明：

```
type SuperFoo<T> = <X>(x: T) => X;
```

>2. 阅读理解课件中的 t3.ts

- ▶ 注意分析其中的 IsTransporter() 的三个不同实现。
- ▶ 注意理解 IsTransporter3() 的设计与使用。

THANKS