11 | TypeScript中的同名处理策略

周爱民（Aimingoo）

目录

declare scope

名字空间 类型 值

Declaration Type	Namespace	Type	Value
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

总结

1. 总的来说有两条规则
   - 只要在不同的域下面，允许同名（namespace / type / value），并且它们可以被合并为一个名字导入导出
   - 在同一个域下面，有三种同名策略：合并（merging）、重载（overload）、覆盖（override）

2. 区分“有实体的”名字空间或对象
   - 名字空间（或模块）是否有实体，取决于它的具体声明
   - 具名函数、类和枚举总是有实体的

名词/概念

名词、术语

名字空间（命名空间）：Namespace
声明：Declarations / declare
定义：Definitions / define
实体：Entities
模块：Modules

概念

- 声明用于在域中创建实体（*a declaration creates entities*）。
  @see: https://www.typescriptlang.org/docs/handbook/declaration-merging.html
- 定义（Define）可以理解为多个声明合并后的声明（declare）统而一之的概念，包含所有声明的全部性质（*merging separate declarations … into a single definition, and the merged definition has the features of both of the original declarations*）。
  @see: https://www.typescriptlang.org/docs/handbook/declaration-merging.html
- 在 Namespace 和 Type 中的实体不会最终出现在编译后的 JavaScript 代码中。
- 早期TypeScript中的模块是名字空间的一个不同的称谓（和实现方式），但由于ES2005标准化了模块机制，所以现在TypeScript中的“名字空间”将是一个与模块——表现上看起来——没有关系的概念：名字空间是一种特定于TypeScript的组织代码的方式（*a TypeScript-specific way to organize code*）。
  @see: https://www.typescriptlang.org/docs/handbook/namespaces-and-modules.html

极客时间

Q&A

Q: 有什么官方文档来解释 TypeScript “将声明的名字分成三个域”吗?

A: 没有。在这里所说的“三个域”，在官方文档中只是称为“三个组（*three groups*）”。本课程引入了“域”这个概念，是想突出这三个域之间是相互隔离的，类似于执行引擎或代码解析中的“作用域（Scope）”。需要说明的是，这也潜在地说明在第二篇中的“运算”可以发生的范围，这一范围被“严格地”限制在“类型（Type）”这个域中。

Q: 三个域中的“声明”是不同的吗?

A: 在官方文档中，在这三个域中创建一个名字的方法都称为“声明（declarations）”，分别为：创建名字空间的声明、创建类型的声明和创建值的声明（*Namespace-creating/Type-creating/value-creating declarations*）。

Q: 在本课程的讲述中，从名字空间中取 type/value 的名字时，是否必须通过别名X?

A: 不需要。在课程中使用“X”这个名字空间并创建“Streams”别名，只是为了展示时结构清晰。而事实上可以更简单地示例如下：

```
import Streams from "node:Stream";
type Stream = Streams;
let StreamClass = Streams;   // NOTE: Streams.Stream === Streams
```

Q: TS 与 ES 中的名字空间与模块有什么不同?

A: 在现在的TS和ES中，模块都是指 export/import 语句所在的文件。但对于名字空间来说，在 TS 中是指用“`namespace xxx { … }`”语法声明的名字 xxx，而 ES 中是指用“`import * as xxx …`”得到的名字 xxx。另外，仅对于 TS 来说，考虑到对早期TS的兼容，还会将动态添加了属性的具名函数名用作模块名，并且总是可以将这种模块名“作为同名的名字空间”进行声明合并。


极客时间
THANKS