混入，兼顾值和类型的合并操作

更新时间：2019-07-15 09:58:47

> 耐心是一切聪明才智的基础。
>
> ——柏拉图

混入即把两个对象或者类的内容，混合起来，从而实现一些功能的复用。如果你使用过 Vue，你应该知道 Vue 的 mixins 这个 api，它可以允许你将一些抽离到对象的属性、方法混入到一些组件。接下来我们先来看看个在 JavaScript 中实现的简单混入：

```
class A {
  constructor() {}
  funcA() {
    console.log("here");
  }
}
class B {
  constructor() {}
  funcB() {}
}
const mixin = (target, from) => { // 这里定义一个函数来将一个类混入到目标类
  Object.getOwnPropertyNames(from).forEach(key => { // 通过Object.getOwnPropertyNames可以获取一个对象自身定义的而非继承来的属性名组成的数组
    target[key] = from[key]; // 将源类原型对象上的属性拿来添加到目标类的原型对象上
  });
};
mixin(B.prototype, A.prototype); // 传入两个类的原型对象
const b = new B();
b.funcA(); // here
```

我们通过 Object.getOwnPropertyNames 方法获取一个对象自身的属性，这里自身指除去继承的属性，获取到属性后将属性赋值给目标对象。

这是 JavaScript 中的简单混入，在 TypeScript 中我们知道，除了值还有类型的概念，所以简单地将属性赋值到目标元素是不行的，还要处理类型定义，我们来看下 TypeScript 中混入的例子：

```
class ClassAa {
  isA: boolean;
  funcA() {}
}
class ClassBb {
  isB: boolean;
  funcB() {}
}
// 定义一个类类型接口AB，我们在讲类的时候补充过类和接口之间的继承，也讲过类类型接口
// 这里是让类AB继承ClassAa和ClassBb的类型，所以使用implements关键字，而不是用extends
class AB implements ClassAa, ClassBb {
  constructor() {}
  isA: boolean = false; // 定义两个实例属性
  isB: boolean = false;
  funcA: () => void; // 定义两个方法，并指定类型
  funcB: () => void;
}
function mixins(base: any, from: any[]) { // 这里我们改造一下，直接传入类，而非其原型对象，base是我们最后要汇总而成的类，from是个数组，是我们要混入的源类组成的数组
  from.forEach(fromItem => {
    Object.getOwnPropertyNames(fromItem.prototype).forEach(key => {
      base.prototype[key] = fromItem.prototype[key];
    });
  });
}
mixins(AB, [ClassAa, ClassBb]);
const ab = new AB();
console.log(ab);
/*
{
    isA: false,
    isB: false,
    __proto__: {
        funcA: f ()
        funcB: f ()
        constructor: f
    }
}
*/
```

来看下这个例子。这个例子中我们定义了两个类 A 和 B，它们分别有自己的方法和实力属性。如果我们想使用它们的所有属性和方法来创建实例，就需要将它们做一个混合。因为包含类型定义，所以我们首先要定义一个接口，来包含着两个类中元素类型的定义。所以我们定义一个类类型接口，然后让这个类类型接口 AB 通过 implements 继承 A 和 B 这两个类，这样类 AB 就会同时拥有类 A 和 B 的类型定义，还有自身定义的一些类型和值。所以此时类 AB 相当于：

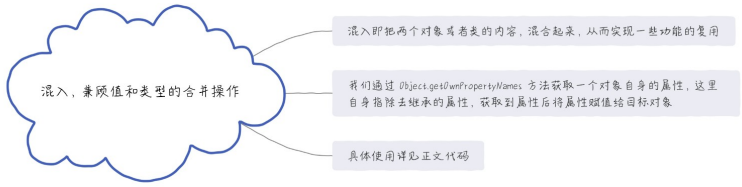
```
class AB {
  isA: boolean = false;
  isB: boolean = false;
  funcA: () => void;
  funcB: () => void;
}
```

然后我们通过 mixins 函数将类 A 和类 B 的原型对象的属性方法赋值给类 AB，因为类 AB 有 funcA 和 funcB 的类型定义，所以可以把 funcA 和 funcB 函数实体赋值给类 AB。

本节小结

本小节我们学习了在TypeScript中如何实现混入，来复用现有逻辑。我们还同时复习了类类型接口和接口继承类的知识，可以去2.10小节复习一下，大家可以多看下本小节的例子，来深入理解类和接口的混合使用。

下个小节我们将学习Promise以及它的语法糖async/await语法，通过它们我们可以实现同步操作，过去我们需要保证代码执行顺序的逻辑，一般都是通过回调函数来实现，现在我们可以使用Promise及其语法糖来更便捷地实现了。



对声明合并的爱与恨

Promise及其语法糖async和await

精选留言 0

欢迎在这里发表留言，作者筛选后可公开显示

目前暂无任何讨论