

第十四课 类与接口

学习目录

- 类与接口的关系
- implements 实现接口

一. 类与接口的关系

ts 中接口的作用可以声明成员的类型，定义一个类型约定，但是不会做具体的实现。ts 中类既声明方法，又做具体实现。

ts 中的类可以构建出类型，因此我们可以在允许使用接口的地方来继承使用类类型，这是他们的第一种关系。

```
class Work{  
  
    a: string;  
  
    b: string;  
  
}  
  
interface GoodWork extends Work{  
  
    c: string;  
  
}  
  
let gw1: GoodWork = {a: '双休', b: '收入多', c: '工作轻松'};
```

ts 还可以强制的让某个类符合某个接口的类型约定，这种实现类型约定的方法，我们称之为 implements 实现。

简单来说，一个类可以继承另一个类，有时候不同类之间会有一些相同的特性，比如相同的方法，只不过每个类在实现这个方法的时候会有不同的方式。因此我们可以把这种相同的特性抽离成接口，在类的定义中用 implements 关键字来实现接口中的类型约定。

二. implements 实现接口

我们可以用播放视频方法为例，我们可以用手机播放视频，可以用笔记本播放视频，因此播放视频就是他们相同的方法特性。

```
interface Video{

    play():void;

}

class Phone implements Video{

    play() {

        console.log('手机播放视频');

    }

}

class Notebook implements Video{

    play() {

        console.log('笔记本播放视频');

    }

}

let p1 = new Phone();

let n1 = new Notebook();
```

```
p1.play();
```

```
n1.play();
```

这里我们把播放视频的方法抽离出来封装成一个接口类型约定，需要这个功能的类可以用 implements 来实现这个方法。

另外一个子类还可以继承另外一个父类之后在实现子类的接口类型约定。

```
class HuaweiPhone extends Phone implements Video{
```

```
    play() {
```

```
        console.log('华为手机播放视频');
```

```
    }
```

```
}
```

```
let h1 = new HuaweiPhone();
```

```
h1.play();
```

另外假如要实现多个接口的类型约定，我们可以用逗号来做到。

```
interface Video{
```

```
    play():void;
```

```
}
```

```
interface Music{
```

```
    listen():void;
```

```
}
```



学习前端，最快的进步是持续！

```
class Phone implements Video,Music{

    play() {

        console.log('手机播放视频');

    },

    listen() {

        console.log('手机播放音乐');

    }

}
```

类与接口的互动使用在 ts 做大型应用的时候比较常用，接口抽离封装类型约定，类实现具体方法，因此接口等于定义了类型约定方法，类可以声明实现具体的方法。

谢谢观看！如果觉得课程还不错的话，记得给个好评！

我是星星课堂老师：周小周