유니온의 인터페이스보다는 인터페이스의 유니온을 사용하기

이펙티브 타입스크립트

@Leemainsw

유니온의 인터페이스

유니온의 인터페이스보다는 인터페이스의 유니온 사용하기

- 하나의 인터페이스 안에서 값의 타입을 유니온으로 사용하는 것
- 속성 간의 관계가 분명하지 않다.

```
interface Layer {
  layout: FillLayout | LineLayout | PointLayout;
  paint: FillPaint | LinePaint | PointPaint;
}
```

인터페이스의 유니온

유니온의 인터페이스보다는 인터페이스의 유니온 사용하기

- 태그된 유니온을 사용하여 만든 것
- 각각 타입의 계층을 분리시키는 것
- 속성 간의 관계가 분명하다.

```
interface FillLayer {
  type: 'fill';
  layout: FillLayout;
  paint: FillPaint;
}
interface LineLayer {
  type: 'line';
  layout: LineLayout;
  paint: LinePaint;
}
interface PointLayer {
  type: 'paint';
  layout: PointLayout;
  paint: PointPaint;
}
type Layer = FillLayer | LineLayer | PointLayer;
```

태그된 유니온

유니온의 인터페이스보다는 인터페이스의 유니온 사용하기

- 태그된 유니온이란
- 상황에 따라 인터페이스를 분리한 후 해당 인터페이스들을 type을 이용하여 유니온으로 사용한 것
- 태그는 type에 주어진 개별 속성이며 런타임 시 타입의 범위를 줄일 수 있도록 도와준다.

태그된 유니온 - (1)

유니온의 인터페이스보다는 인터페이스의 유니온 사용하기

- 분기별 처리가 필요할 경우
- 태그를 사용하여 분기처리를 할 수 있다.
- 동일한 코드가 반복되어 어수선해 보인다는 단점이 있다.

```
function drawLayer(layer: Layer) {
  if (layer.type === 'fill') {
    const {paint} = layer;   // 타입이 FillPaint
    const {layout} = layer;  // 타입이 FillLayout
  } else if (layer.type === 'line') {
    const {paint} = layer;   // 타입이 LinePaint
    const {layout} = layer;  // 타입이 LineLayout
  } else {
    const {paint} = layer;   // 타입이 PointPaint
    const {layout} = layer;  // 타입이 PointLayout
  }
}
```

태그된 유니온 - (2)

유니온의 인터페이스보다는 인터페이스의 유니온 사용하기

- 여러개의 선택적 필드가 동시에 값이 있거나 동시에 undefined인 경우
- 두 개의 속성을 하나의 객체로 모으는 것이 더 나은 설계이다.

```
interface Person {
  name: string;
  birth?: {
    place: string;
    date: Date;
  }
}
```

```
interface Person {
  name: string;
  // 다음은 둘 다 동시에 있거나 동시에 없습니다.
  placeOfBirth?: string;
  dateOfBirth?: Date;
}
```

인터페이스의 유니온

유니온의 인터페이스보다는 인터페이스의 유니온 사용하기

- 타입의 구조에 손댈 수 없는 상황일 경우
- 인터페이스의 유니온을 사용하여 속성 사이의 관계를 모델링할 수 있다.

```
interface Name {
  name: string;
}

interface PersonWithBirth extends Name {
  placeOfBirth: string;
  dateOfBirth: Date;
}

type Person = Name | PersonWithBirth;
```

결론

유니온의 인터페이스보다는 인터페이스의 유니온 사용하기

- 유니온의 인터페이스에서는 속성 간의 관계가 분명하지 않기 때문에 실수가 자주 발생한다.
- 유니온의 인터페이스보다는 인터페이스의 유니온이 더 정확하고 타입스크립트가 이해하기에도 좋다.
- 타입스크립트가 제어 흐름을 분석할 수 있도록 타입에 태그를 넣는 것을 고려해야 한다.
- 태그된 유니온은 타입스크립트와 매우 잘 맞는다.