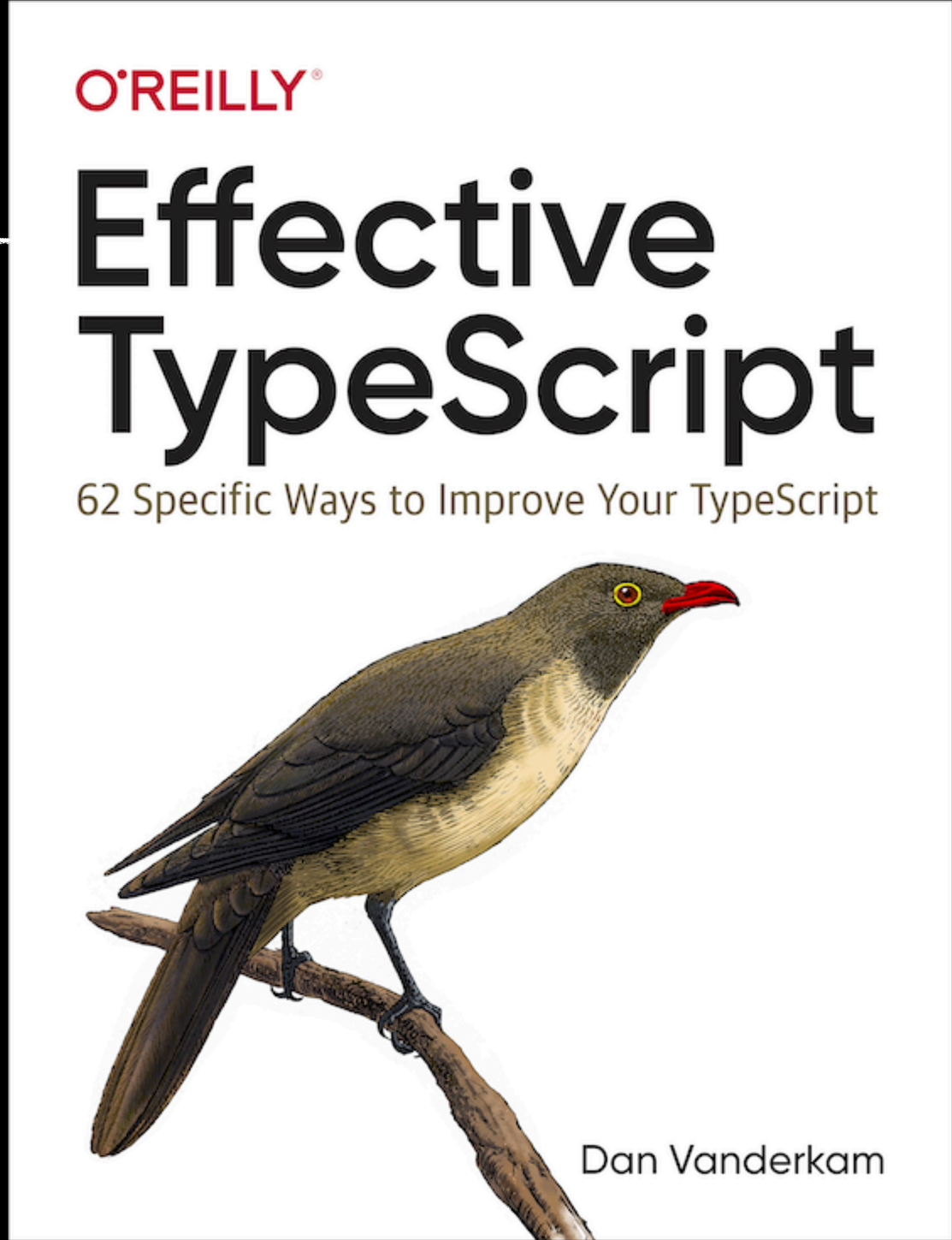
Item 24.

일관성 있는 별칭 사용하기

Effective TypeScript

@Bori-github(이보리)

목 차



1. 새로운 별칭 사용하기

자바스크립트에서는 어떤 값을 새로운 변수에 할당하여 사용

```
1  const bori = {
2    name: "Bori",
3    address: {
4      city: "Seoul"
5    }
6  };
7
8  // boriAddress라는 변수에 bori.address를 할당
9  const boriAddress = bori.address;
10
11 // 변수 boriAddress를 통해 city에 접근
12 console.log(boriAddress.city); // "Seoul"
13
14 // 변수 boriAddress를 통해 city의 값을 재할당
15 boriAddress.city = "Gwangju";
16
17 // 원본의 값도 변경
18 console.log(bori.address.city); // "Gwangju"
19 console.log(boriAddress.city); // "Gwangju"
```

- `boriAddress` 라는 변수(별칭) 선언
- 새로운 변수에 `bori.address`를 할당

별칭을 남용할 경우 문제점

⇒ 제어 흐름 분석이 어려워진다.

2. 별칭과 제어 흐름

```ts
interface Person {
  name: string;
  address?: {
    city: string;
  }
}

function getAddress (person: Person) {
  if (person.address) {
    console.log(person.address.city);
  }
}
```

`person.address` 에 새로운 별칭을 할당하여 사용하면 어떻게 될까요?

2. 별칭과 제어 흐름

```ts
function getAddress (person: Person) {
  const personAddress = person.address;
  if (person.address) {
    console.log(personAddress.city);
  }
}
```

```
'personAddress'은(는) 'undefined'일 수 있습니다. ts(18048)

const personAddress: {
    city: string;
} | undefined
```

`person.address` 는 속성 체크를 통해 타입 정제

하지만 `personAddress`는 정제되지 않아 오류가 발생

2. 별칭과 제어 흐름

```
function getAddress (person: Person) {
  const personAddress = person.address;
  if(personAddress) { // personAddress의 속성 체크
    console.log(personAddress.city); // 오류가 발생하지 않음
  }
}
```

별칭을 일관성 있게 사용하면 오류를 방지할 수 있다.

아직 남아있는 문제점?

⇒ 변수 `personAddress`는 `person.address`와 같은 값인데 단지 이름만 다른 것

3. 객체 비구조화 사용하기

```
// 객체 비구조화를 이용하여 간결한 코드 작성하기
function getAddress (person: Person) {
  const { address } = person;
  if (address) {
    const { city } = address;
    console.log(city);
  }
}
```

객체 비구조화를 사용하여 일관된 이름을 사용하고, 코드를 보다 간결하게 작성

3. 객체 비구조화 사용하기

객체 비구조화 사용 시 주의해야할 부분

선택적 프로퍼티를 기준으로 속성 체크를 할 경우,

비구조화를 통해 사용한 별칭의 값과 원본 객체를 통해 접근한 속성의 값이 다른 값을 참조

4. 타입 정제 무효화

```ts
function removeAddress(person: Person) { /*...*/ };

if (bori.address) {
  bori.address

  // removeAddress 함수가 bori.address를 제거할 가능성이 있음
  removeAddress(bori);

  // removeAddress 호출 이후 bori.address가 다시 undefined일 가능성이 있음
  bori.address
}
```

```ts
(property) address: {
    city: string;
}
```

`removeAddress` 함수의 호출로 인해 `bori.address` 가 제거될 가능성이 있다.

⇒ 타입트스립트는 함수가 타입 정제를 무효화 하지 않는다고 가정하지만,

실제로는 **무효화될 가능성**이 있다.

5. 마무리

- 별칭을 남용한다면 제어 흐름을 분석하기 어려워집니다.

  따라서 변수에 별칭을 사용할 때는 **일관성있게 사용**해야 합니다.

- **비구조화 문법을 사용**해서 일관된 이름을 사용하는 것이 좋습니다.

- 함수 호출로 인해 객체 속성의 타입 정제를 무효화할 수 있으므로, 속성보다 비구조화하여 뽑아낸 **지역 변수를 사용**하면 타입 정제를 믿을 수 있습니다.

- 이 때 객체 비구조화하여 뽑아낸 변수와 원본 객체를 통해 접근한 속성의 값이 다를 수 있음을 주의해야합니다.

감사합니다 :)

Effective TypeScript

@Bori-github(이보리)