

Box<'dyn error> 동적 크기의 타입을 나타내는 trait object?

사용자 자어의 error 타입 등 다양한 형태의 error를 수용할 수 있다.

trait object?

트레이트를 사용한 동적 디스패치를 가능하게 하는 개념?

트레이트? 동적 디스패치?

-**트레이트(Trait)**는 Rust에서 **공통된 동작**을 정의하고, 이를 여러 타입에 적용할 수 있게 해주는 **인터페이스**입니다. 객체 지향 언어의 **인터페이스** 또는 **추상 클래스**와 비슷하지만, Rust에서는 매우 유연하고 강력한 방식으로 동작합니다.

여러 타입에서 공통으로 사용할 수 있는 메서드의 시그니처를 정의

```
trait Speak { fn speak(&self); }
struct Dog; struct Cat;
impl Speak for Dog { fn speak(&self) { println!("Woof!"); } }
impl Speak for Cat { fn speak(&self) { println!("Meow!"); } }

let dog = Dog; let cat = Cat;
dog.speak(); // 출력: Woof!
cat.speak(); // 출력: Meow!
```

이후 trait 챕터 이해하는데 도움이 되었다.

Different types share the same behavior if we can call the same methods on all of those types.

-동적 디스패치는 메서드나 함수 호출을 **실행 시점에 결정**(동적 바인딩)

다형성과 유사, C++에서 다뤘었음