An associated const has been referenced in a pattern.

Erroneous code example:

```
enum EFoo { A, B, C, D }

trait Foo {
    const X: EFoo;
}

fn test<A: Foo>(arg: EFoo) {
    match arg {
        A::X => { // error!
            println!("A::X");
        }
    }
}
```

`const` and `static` mean different things. A `const` is a compile-time constant, an alias for a literal value. This property means you can match it directly within a pattern.

The `static` keyword, on the other hand, guarantees a fixed location in memory. This does not always mean that the value is constant. For example, a global mutex can be declared `static` as well.

If you want to match against a `static`, consider using a guard instead:

```
static FORTY_TWO: i32 = 42;

match Some(42) {
    Some(x) if x == FORTY_TWO => {}
    _ => {}
}
```