A method or constant was implemented on a primitive type.

Erroneous code example:

```
struct Foo {
    x: i32
}

impl *mut Foo {}
// error: cannot define inherent `impl` for primitive types
```

This isn't allowed, but using a trait to implement a method or constant is a good solution. Example:

```
struct Foo {
    x: i32
}

trait Bar {
    fn bar();
}

impl Bar for *mut Foo {
    fn bar() {} // ok!
}
```

Instead of defining an inherent implementation on a reference, you could also move the reference inside the implementation:

```
struct Foo;

impl &Foo { // error: no nominal type found for inherent implementation
    fn bar(self, other: Self) {}
}
```

becomes

```
struct Foo;

impl Foo {
    fn bar(&self, other: &Self) {}
}
```