The `Copy` trait was implemented on a type which contains a field that doesn't implement the `Copy` trait.

Erroneous code example:

```
struct Foo {
    foo: Vec<u32>,
}

impl Copy for Foo { } // error!
```

The `Copy` trait is implemented by default only on primitive types. If your type only contains primitive types, you'll be able to implement `Copy` on it. Otherwise, it won't be possible.

Here's another example that will fail:

```
#[derive(Copy)] // error!
struct Foo<'a> {
    ty: &'a mut bool,
}
```

This fails because `&mut T` is not `Copy`, even when `T` is `Copy` (this differs from the behavior for `&T`, which is always `Copy`).