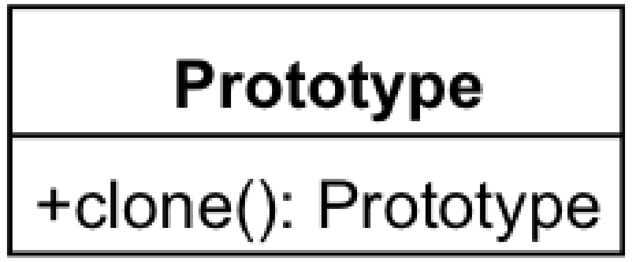
Prototype

Creates new objects by copying a prototype.

Instantiate a prototype

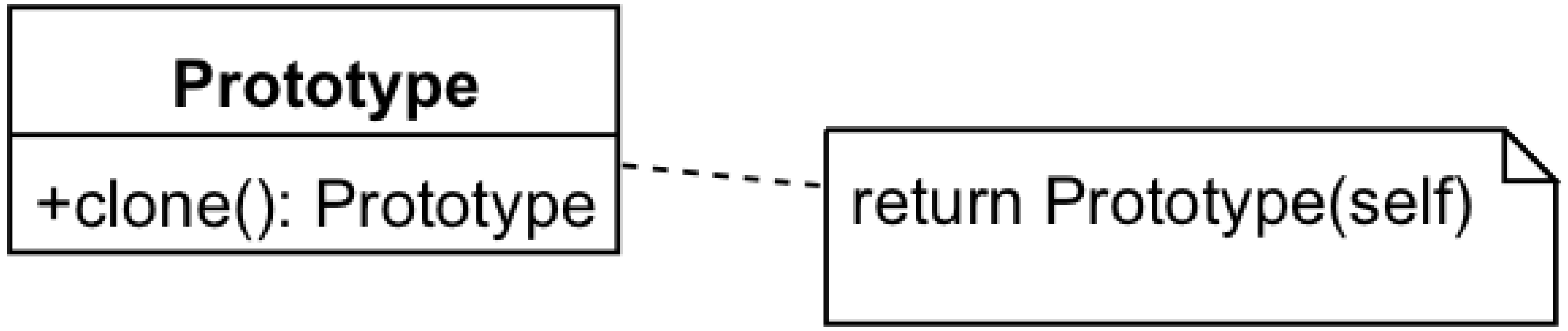


Prototype

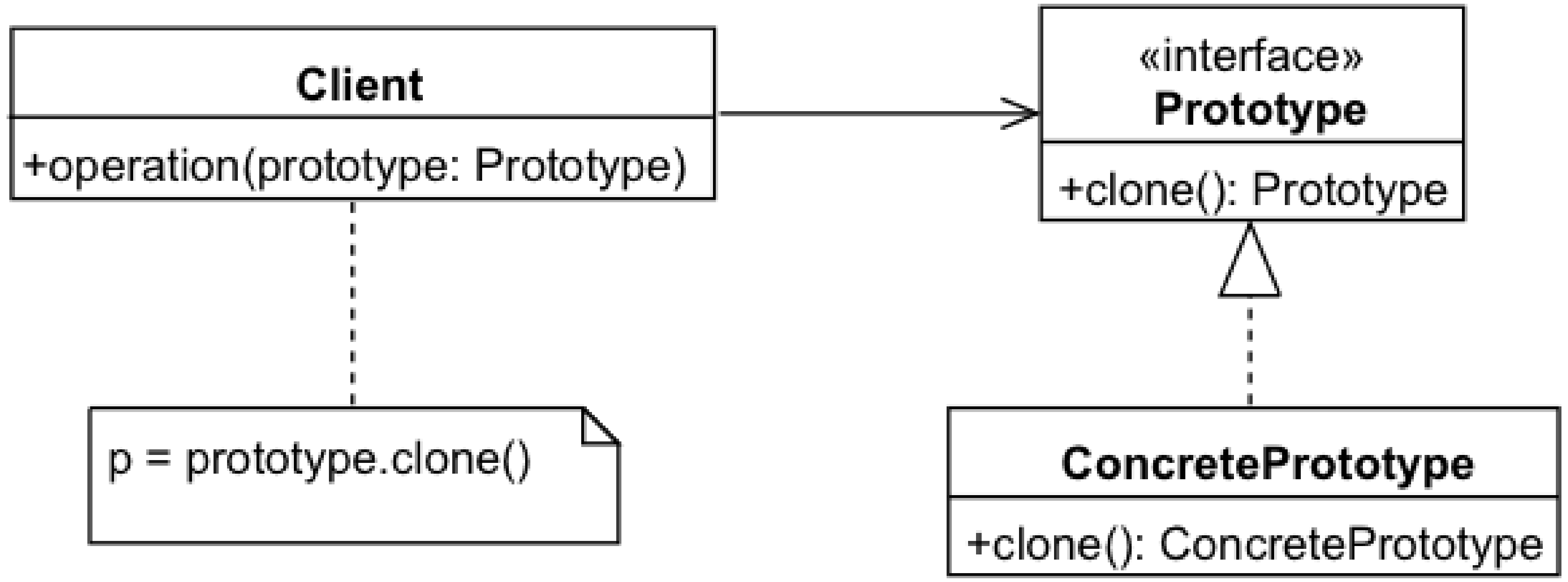
+clone(): Prototype

Instantiate a prototype

Create new instances
by cloning the prototype



Polymorphic instantiation



Automatic cloning of value types

How to Avoid Common Pitfalls?

How to Avoid Common Pitfalls?

Cloning of reference types requires extra coding

How to Avoid Common Pitfalls?

Cloning of reference types requires extra coding

Understand the difference between shallow and deep copying

How to Avoid Common Pitfalls?

Cloning of reference types requires extra coding

Understand the difference between shallow and deep copying

The clones need to be independent objects

Copying Value vs. Reference Types

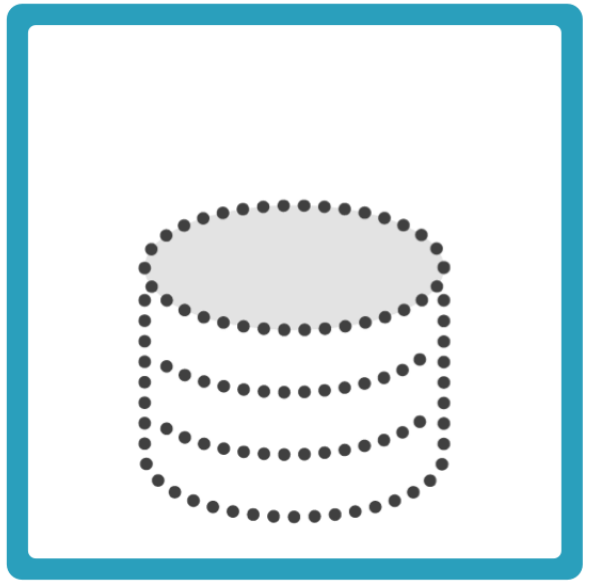
Swift Types

Swift Types

Value type

Swift Types

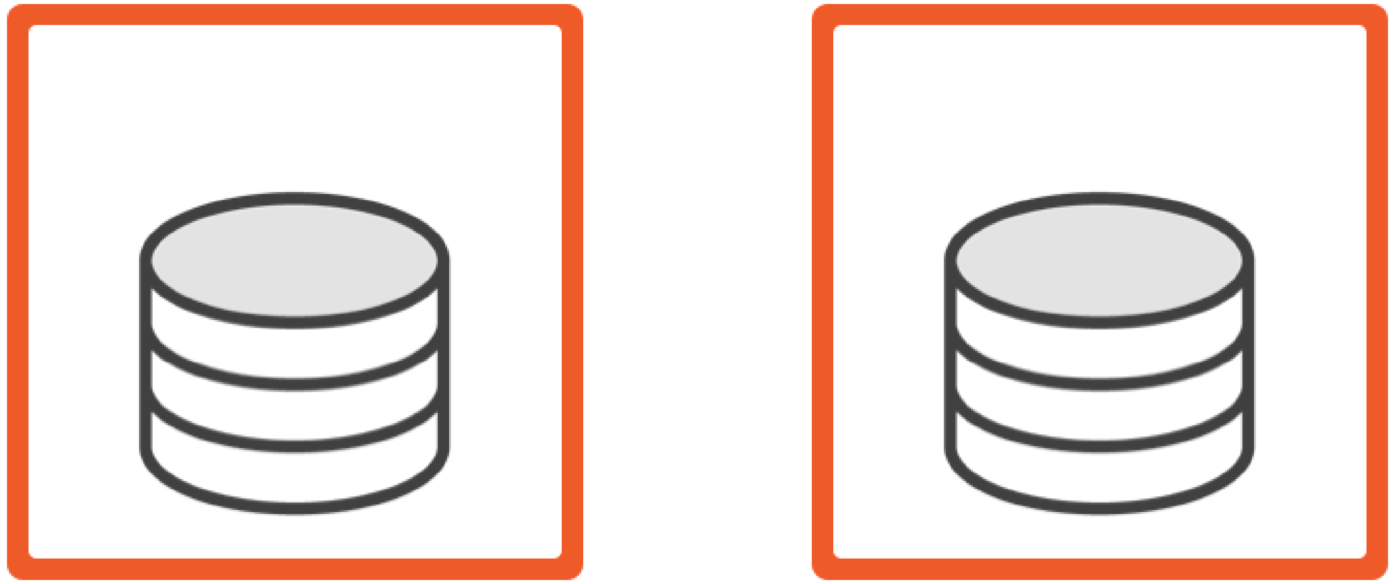
Value type



Reference type

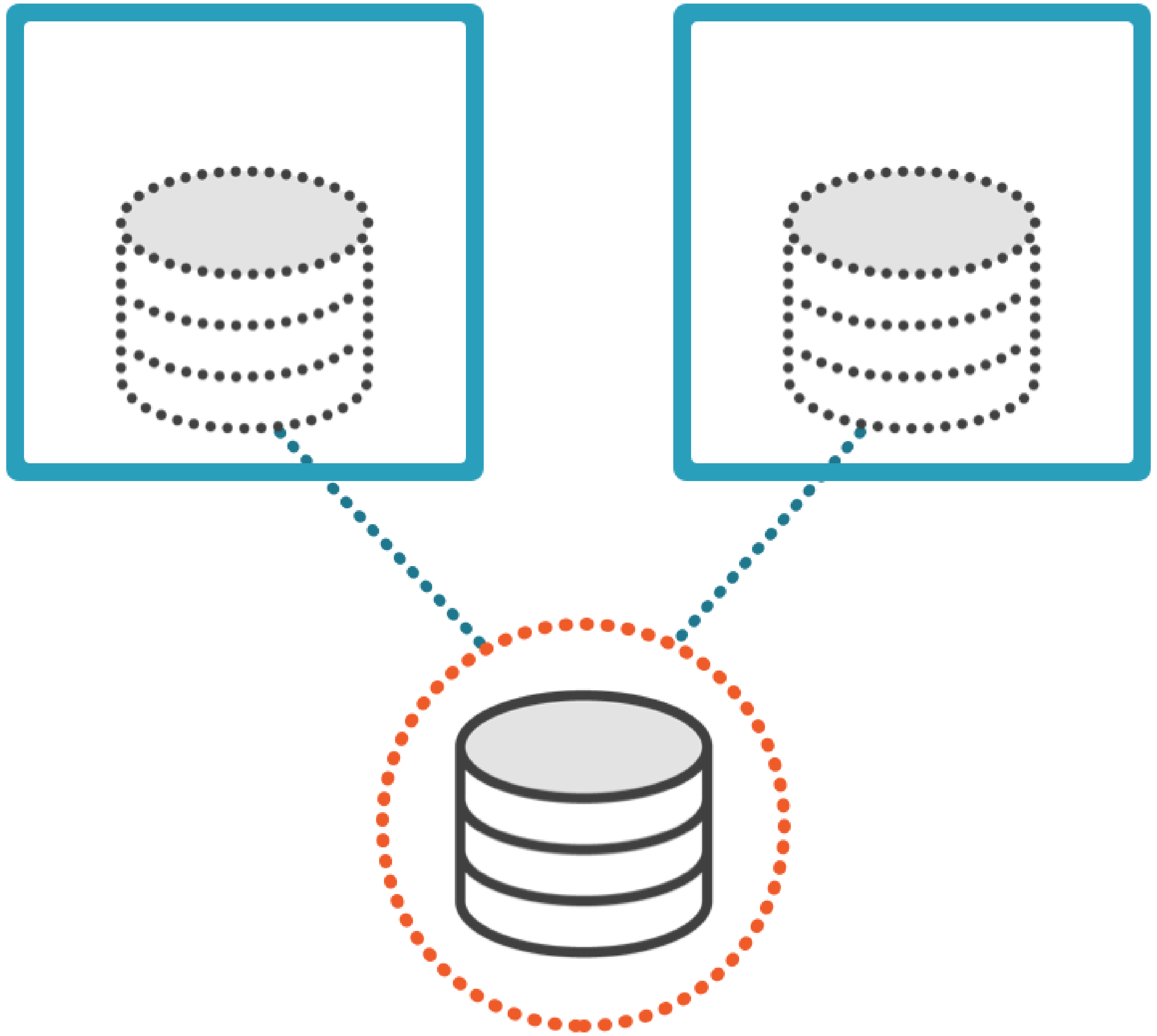
Copying Value Types

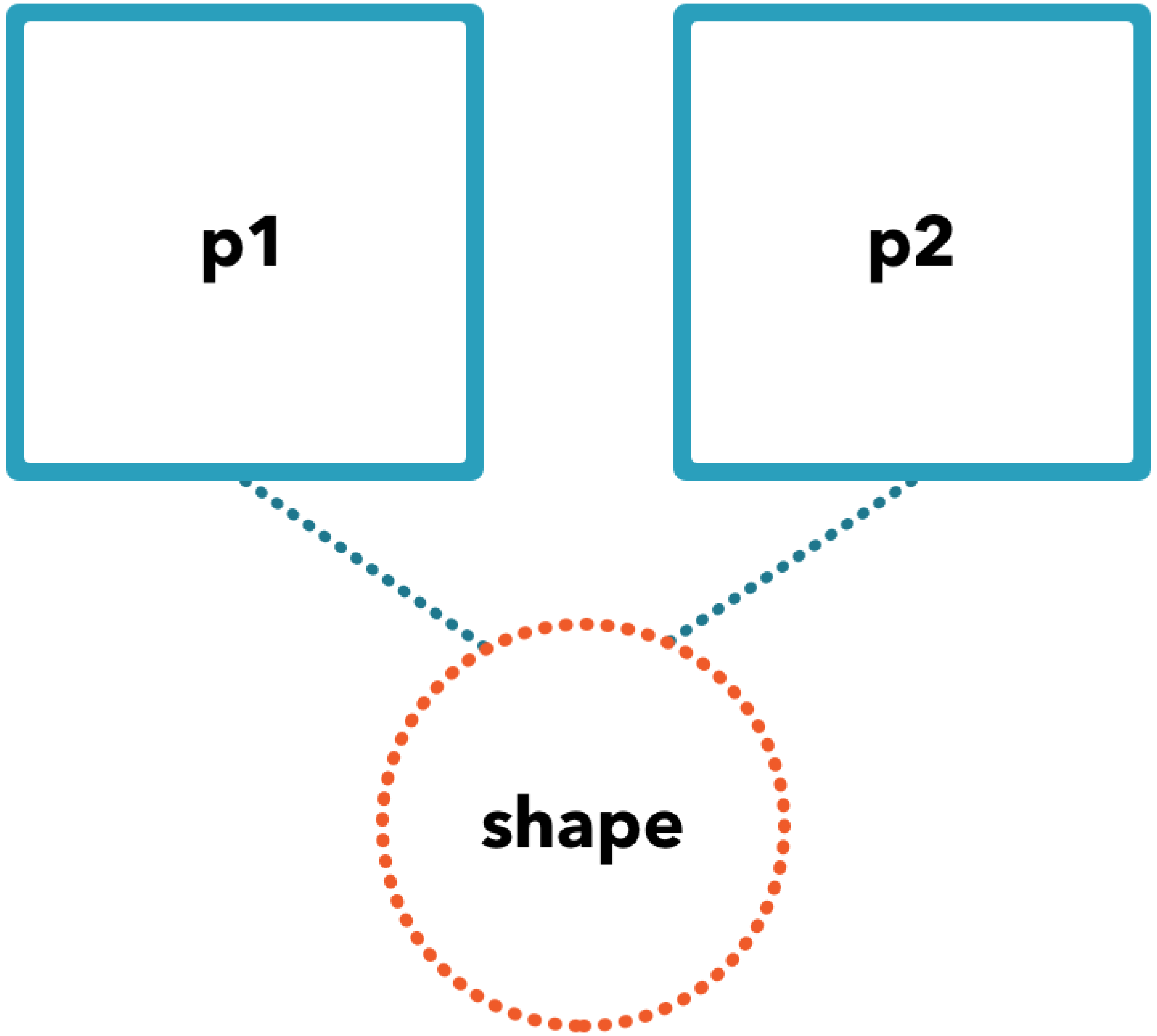
Copying Value Types

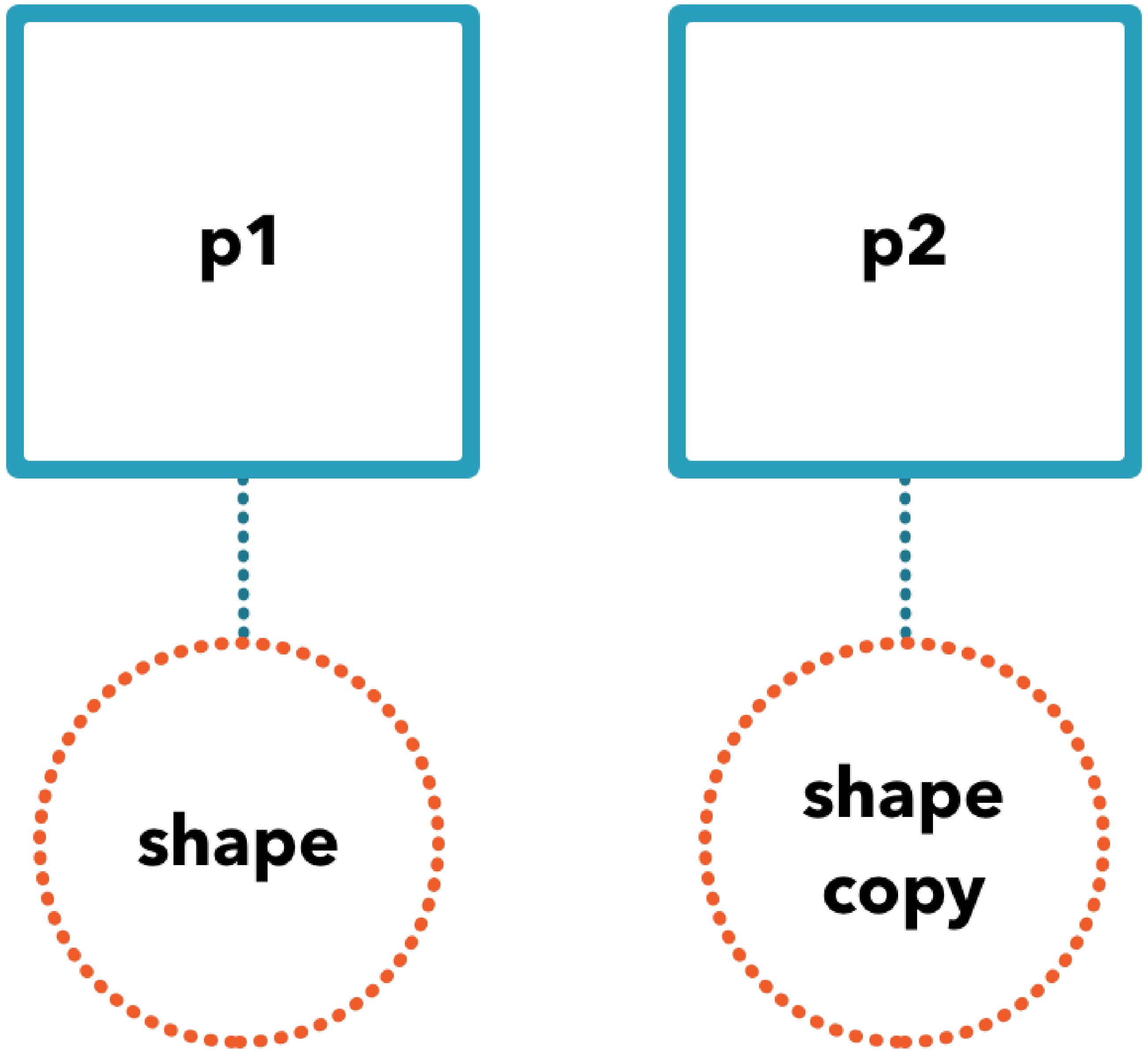


Copying Reference Types

Copying Reference Types




p1
p2
shape



p1

p2

shape

shape copy

The Prototype

Purpose

- Favor cloning over initialization
- Clones are standalone objects

Value types vs. classes

- Value types get copied by default
- Classes need to adopt NSCopying

Common pitfalls

- Deep copies duplicate the object’s data
- Shallow copies duplicate pointers only