Composition

Michael Van Sickle

@vansimke

Overview

Inheritance and composition

Strategies

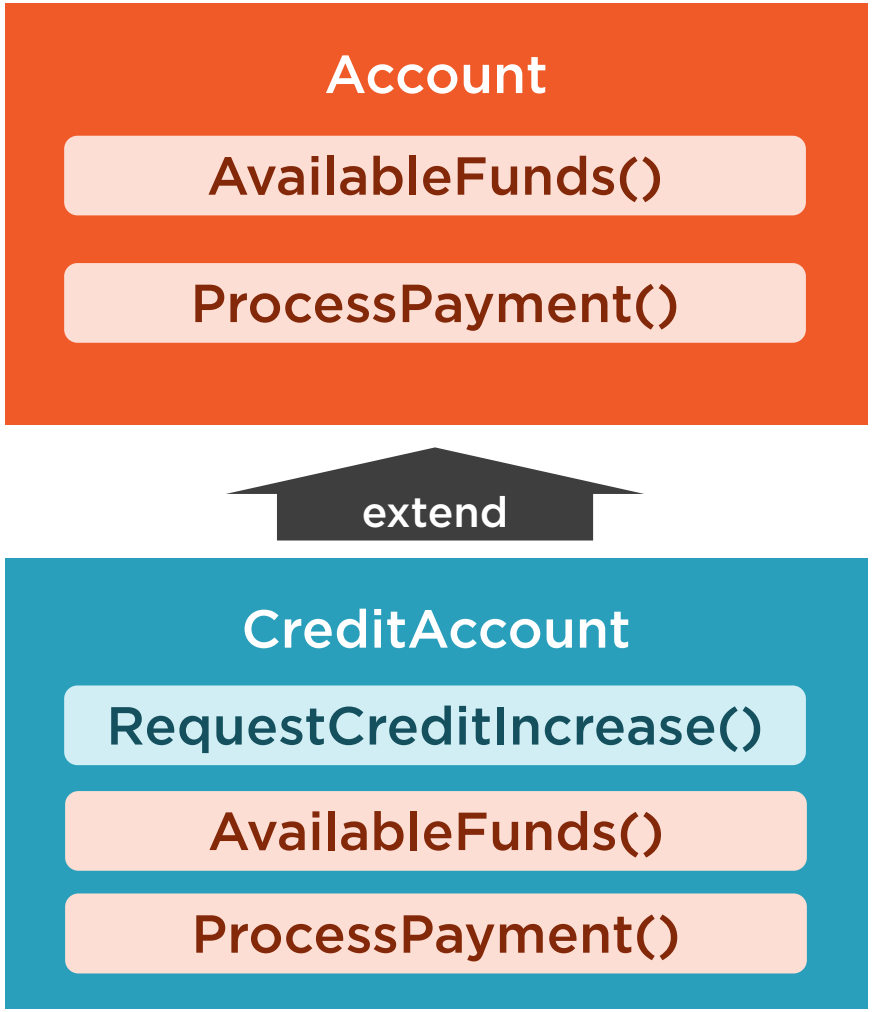
Inheritance

Behavior reuse strategy where a type is based upon another type, allowing it to **inherit** functionality from the base type.

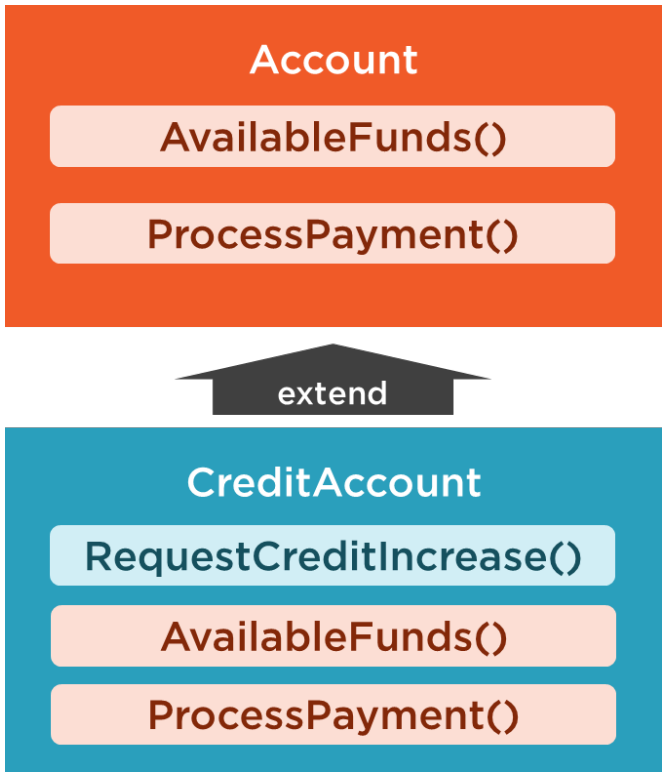
Composition

Behavior reuse strategy where a type contains objects that have desired functionality. The type delegates calls to those objects to use their behaviors.

Inheritance Relationship



Challenges with Inheritance



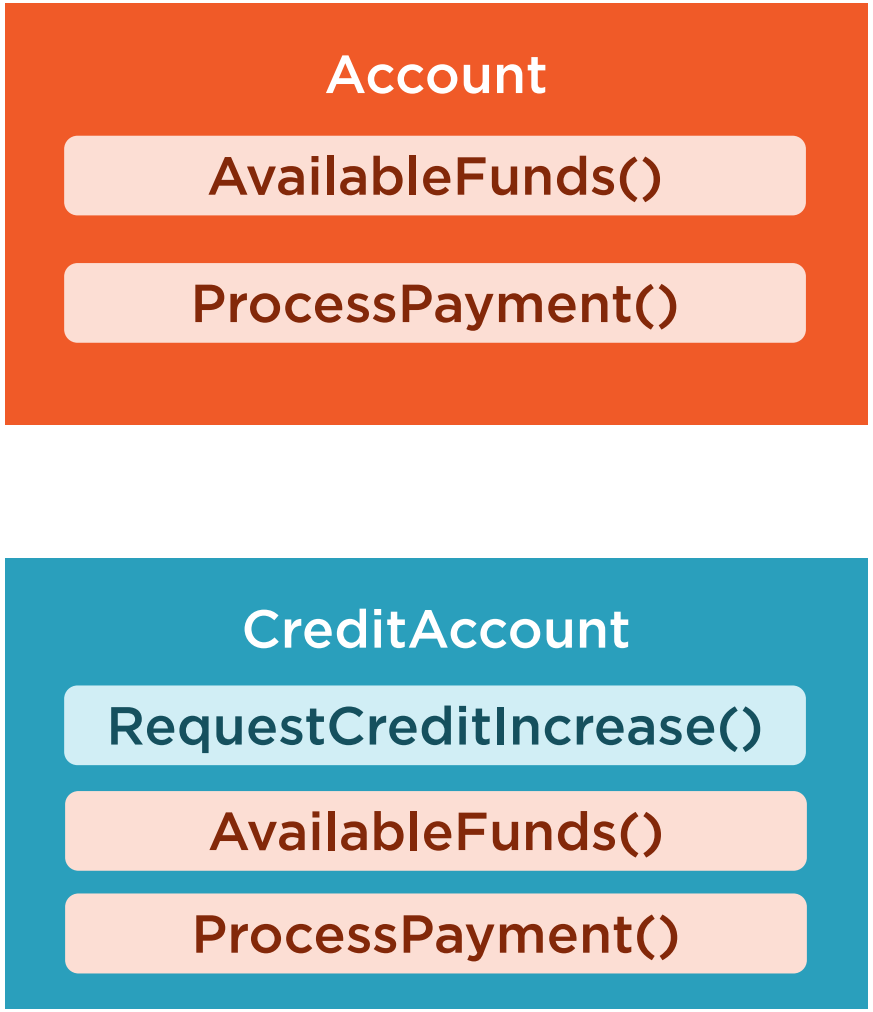
Tightly couples parent and child

Hard to debug and maintain

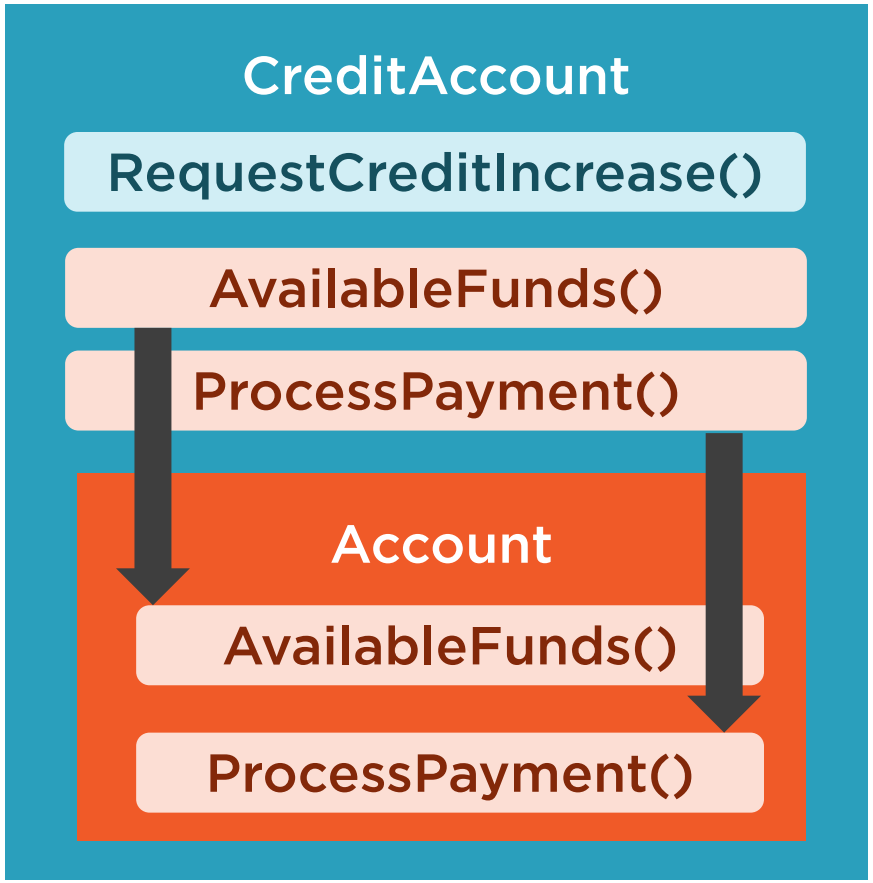
All or nothing

Not supported in Go!

Composition Relationship



Composition Relationship



Type Embedding

Composition

```
type Account struct { ... }
func (a *Account) AvailableFunds() float32 { ... }
func (a *Account) ProcessPayment(amount float32) bool { ... }

type CreditAccount struct {
    Account
}

ca := &CreditAccount{}
funds := ca.AvailableFunds()
```

Resolving Conflicts

```
type CreditAccount struct { ... }

func (c *CreditAccount) AvailableFunds() float32 { ... }

type CheckingAccount struct { ... }

func (c *CheckingAccount) AvailableFunds() float32 { ... }


type HybridAccount struct {
    CreditAccount
    CheckingAccount
}
func (h *HybridAccount) AvailableFunds() float32 {
    return h.CreditAccount.AvailableFunds() + h.CheckingAccount.AvailableFunds()
}
```

Overview

Inheritance and composition

Strategies

- Embedding