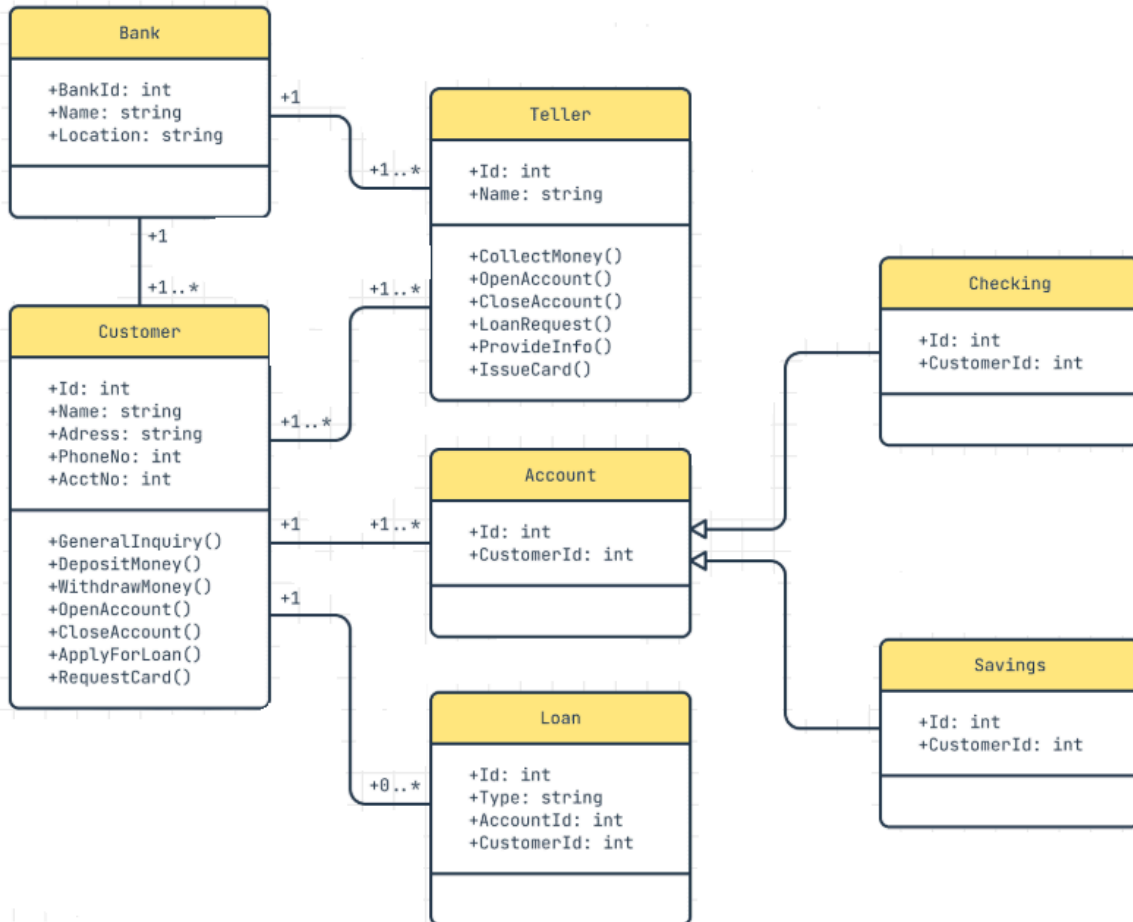


## Problem - Quiz B



1. If the customer has a name, then they must have a phone number and an address.

**context Account**

**inv AccountCheck:**

**self.customer.name <> "" implies**

**self.customer.phoneNo <> 0 implies**

**self.customer.address <> ""**

2. Every customer must have accounts that match their ID

**context Bank**

**inv AllCustomersHaveAccounts:**

**self.customers->forAll(c | c.accounts->forAll(a | a.customerId = c.id))**

3. The loan must be either Home, Auto, Personal, or Business.

**context Loan**

**inv ValidLoanType:**

**self.type = 'Home' or self.type = 'Auto' or self.type = 'Personal' or self.type = 'Business'**

4. Every teller in the bank where the customer has an account must have a name.

**context Account**

**inv NameExists:**

**self.customer.bank.tellers->forAll(t | t.name <> "")**

5. The customer must have at least one account

**context Customer**

**inv MustHaveAtLeastOneAccount:**

**self.accounts->size() > 0**

**inv MustHaveAtLeastOneAccount1:**

**self.accounts->notEmpty()**

6. The account must be either Checking, Savings, or Business

**context Account**

**inv TypeCheck:**

**self.ocllsTypeOf(Checking) or  
self.ocllsTypeOf(Savings) or  
self.ocllsTypeOf(BusinessAccount)**

7. If the loan is for a car, the amount must be less than 500,000. If it's for a home, the amount must be less than 1,000,000

```
context Loan
inv TypeCheckSmell:
  if self.type = 'Car' then self.amount <= 500000
  else if self.type = 'Home' then self.amount <= 1000000
  else true endif endif
```

---

## Solution

### 1. Implies Chain (smell: long chains of implies conditions)

```
context Customer
inv HasCompleteProfile:
  not self.name.isNull() and self.phoneNo <> 0 and self.accounts->notEmpty()
```

### 2. ForAll Chain

```
context Customer
inv AllAccountsMatchCustomerId:
  self.accounts->forAll(a | a.customerId = self.id)
```

### 3. Verbose Expression (overly long and complicated)

```
context Loan
inv ValidLoanType:
  Set{'Home', 'Auto', 'Personal', 'Business'}->includes(self.type)
```

OR

```
context Loan
def: allowedTypes: Set(String) = Set{'Home', 'Auto', 'Personal', 'Business'}
```

```
inv ValidLoanType:
  self.allowedTypes->includes(self.type)
```

### 4. Long Journey

```
context Bank
inv TellersHaveNames:
  self.tellers->forAll(t | not t.name.isNull() and t.name <> "")
```

### Duplication

5.

```
context Customer
inv MustHaveAtLeastOneAccount:
    self.accounts->notEmpty()
```

6.

```
context Account
inv ValidAccountType:
    Set{'Checking', 'Savings', 'Business'}->includes(self.accountType)
```

7.

```
context Loan
inv CarLoanLimit:
    self.type = 'Car' implies self.amount <= 500000

inv HomeLoanLimit:
    self.type = 'Home' implies self.amount <= 1000000
```