



REALTY

Unset

```
{context Realty
  inv: fractions.fractionOwned->sum() = 1.0 // TODO confirmar este OCL
  inv: fractions->isUnique(f | f.owner)
  inv: saleHistory->select(s | s.status=OPEN)->size() <= 1
  inv: rentHistory->select(r | r.status=ONGOING or r.status=PENDING)->size() <= 1
}
```

- The sum of a realty's fractions owned by different entities must be equal to 1 (100%)
- There can't be more than one OPEN sale at the same time.
- There can't be more than one ONGOING or PENDING rent contract at the same time.

Transfer Ownership

Unset

```
{context Realty::transferOwnership(fraction:Real, from:Account, to:Account)
  pre: fractions->select(f | f.owner=from).fractionOwned > fraction

  post: fractions->select(f | f.owner=from).fractionOwned = fractions->select(f |
f.owner=from).fractionOwned@pre - fraction and
  fractions->select(f | f.owner=to).fractionOwned = fractions->select(f |
f.owner=to).fractionOwned@pre + fraction
}
```

- The account giving up the ownership must first have a fractional ownership greater or equal to the fraction being transferred
- After the transaction, the ownership must have been transferred from the sender to the receiver

REALTY SALE

Unset

```
{context RealtySale
  inv: status=OPEN or status=CLOSED or status=CANCELED
  inv: if status=OPEN then buyer=null and realty.fractions->select(f | f.owner =
issuer).fractionOwned >= fraction endif
  inv: if status=CLOSED then buyer <> null endif
  inv: issuer <> buyer
}
```

- A sale can have the status OPEN, CLOSED or CANCELED;
- If it has the status OPEN or CANCELED, then it does not have a defined buyer, and the issuer of the transaction must still have a fraction of the realty equal to or greater than the one being sold;
- The issuer of a sale must always be different from the buyer.

Unset

```
{context Bid
  inv: bidder <> sale.issuer
}
```

- The issuer of a sale cannot place a bid on it

Place Bid

May be called by any user of the system, except the issuer of the sale.

Unset

```
{context RealtySale::bid(bid : Bid)
  pre: status=OPEN and bid.bidder.availableBalance >= bid.amount

  post: self.bids->includes(bid) and bid.bidder.availableBalance =
bid.bidder.availableBalance@pre - bid.amount and if bid.amount >= price then
(postConditions da close) endif
}
```

- To place a bid, the sale must be OPEN, and the person making the bid must have an available balance equal to or greater than the bid amount;
- After placing a bid, the bid is listed in the bid list, and the amount equivalent to the bid value is locked in the bidder's wallet.

Retrieve Bid

May only be called by the issuer of the bid being canceled.

Unset

```
{context RealtySale::retrieveBid(bid : Bid)
  pre: status=OPEN and self.bids->includes(bid)

  post: self.bids->excludes(bid) and bid.bidder.availableBalance =
bid.bidder.availableBalance@pre + bid.amount
}
```

- It is only possible to undo a bid if the sale is still OPEN, and if the bid exists;
- After undoing a bid, the bid is no longer listed in the sale's bid list and the bidder's locked balance is unlocked.

Close Sale

May only be called by the issuer of the sale.

Unset

```
{context RealtySale::close(bid: Bid)
  pre: status=OPEN and self.bids->includes(bid)

  post: status=CLOSED and
    let sellerOwnershipBefore: Real = realty.fractions->select(f |
f.owner=issuer).fractionOwned@pre in
    let sellerOwnershipAfter: Real = realty.fractions->select(f |
f.owner=issuer).fractionOwned in
    let buyerOwnershipBefore: Real = realty.fractions->select(f |
f.owner=buyer).fractionOwned@pre in
    let buyerOwnershipAfter: Real = Real = realty.fractions->select(f |
f.owner=buyer).fractionOwned in
    sellerOwnershipAfter = sellerOwnershipBefore - fraction and
    buyerOwnershipAfter = buyerOwnershipBefore + fraction and

    buyer.realBalance = buyer.realBalance@pre - bid.amount and

    issuer.realBalance = issuer.realBalance@pre + bid.amount *
(1-bid.comission.fraction) and

    bid.comission.agency.realBalance = bid.comission.agency.realBalance@pre +
bid.amount * bid.comission.fraction and

    issuer.availableBalance = issuer.availableBalance@pre
+ bid.amount * (1-bid.comission.fraction) and

    bid.comission.agency.availableBalance =
bid.comission.agency.availableBalance@pre
+ bid.amount * bid.comission.fraction and

bids->select(b| b.bidder<>buyer)->forAll(b| b.bidder.availableBalance =
b.bidder.availableBalance@pre)
}
```

- Only an OPEN sale can be closed, and it is necessary to specify the bid to be accepted;
- After closing, the sale is CLOSED, and the ownership sold is transferred to the buyer;
- The locked value in the buyer's wallet is transferred to the seller and the real estate agent, in proportion to the commission fraction;

- The remaining locked amounts in the remaining bidders' wallets are unlocked.

Cancel Sale

May only be called by the issuer of the sale.

Unset

```
{context RealtySale::cancel()
  pre: status=OPEN

  post: status=CANCELED and bids->forAll(b| b.bidder.availableBalance =
b.bidder.availableBalance@pre + b.amount)

}
```

- Only an OPEN sale can be canceled;
- After canceling, the amounts locked in all bidders' wallets are unlocked;

RENT CONTRACT

Unset

```
{context RentContract
  inv: status=PENDING or status=OPEN or status=ONGOING or status=TERMINATED
  inv: if status=PENDING or status=OPEN then tenant=null endif
  inv: if status=ONGOING or status=TERMINATED then tenant<>null endif
}
```

- A rent contract can have the states PENDING, OPEN, ONGOING and TERMINATED;
- A contract in the PENDING or OPEN states does not yet have an associated tenant;
- A contract in ONGOING or TERMINATED states must have an associated tenant.

Approve Rental Contract

May be called by any landlord.

Unset

```
{context RentContract::approve(user:Account)
  pre: status=PENDING and realty.fractions->select(f | f.owner =
user).fractionOwned >= 0

  post:landlordSigns->includes(user) and
  if landlordSigns->size() <> realty.fractions->collect(owner)->size() then
status=PENDING endif and
  if landlordSigns->size() == realty.fractions->collect(owner)->size() then
status=OPEN endif
}
```

- It is assumed that for a contract to be in OPEN status, it must have approval from all landlords
- Only a PENDING contract may be approved.
- After approving, if all landlords have approved the rent contract, then its status is set to OPEN

Start Renting

May be called by any user of the system.

Unset

```
{context RentContract::rent(duration: int, user: Account)
  pre: status=OPEN and tenant.availableBalance >= 2*price

  post: status=ONGOING and paymentCounter=1 and self.duration=duration and
tenant.availableBalance = tenant.availableBalance@pre - 2*price and tenant.realBalance
= tenant.realBalance@pre - price and

landlords->forall(1 | 1.availableBalance = 1.availableBalance@pre
+ price * 1.inventory->one(o|o.realty=realty).fractionOwned) and
landlords->forall(1 | 1.realBalance = 1.realBalance@pre
+ price * 1.inventory->one(o|o.realty=realty).fractionOwned)
}
```

- To start a lease, the rental contract must be in OPEN status, and the tenant's balance must be equal to or greater than twice the rental price (1st month + security deposit);
- After starting the lease, the contract becomes ONGOING, the value of the 1st month is decremented from the tenant's wallet, and the value of the deposit is locked;
- The value of the 1st month's rent is distributed among the various landlords, in proportion to the ownership fraction acquired.

Collect Rent

May be called by any landlord.

Unset

```
{context RentContract::collectRent()
  pre: status = ONGOING and tenant.availableBalance >= price and
paymentCounter<duration and today().monthsBetween(startDate)>=paymentCounter

  post: paymentCounter = paymentCounter@pre + 1 and
tenant.realBalance = tenant.realBalance@pre - price and
tenant.availableBalance = tenant.availableBalance@pre - price and

landlords->forall(1 | 1.availableBalance = 1.availableBalance@pre + price *
1.inventory->one(o|o.realty=realty).fractionOwned) and
landlords->forall(1 | 1.realBalance = 1.realBalance@pre + price *
1.inventory->one(o|o.realty=realty).fractionOwned)
}
```


- Paying rent is only possible on a contract that is ONGOING and whose number of payments has not reached the duration specified in the contract;
- The tenant needs to have sufficient available balance;
- After paying rent, the payment counter is increased, and the rent value is distributed among the various landlords, in proportion to the ownership fraction acquired.

Comment:

Razões para collectRent em vez de payRent - na interface dos landlords ter um 1 button collect all rents (automatizando o processo). Porque idealmente teríamos um evento que todos os meses fizesse a transferência automaticamente, mas em fabric é tough

Terminate Contract

May be called by any landlord

Unset

```
{context RentContract::terminate(safetyDepositPenalty :Int)
  pre: status=ONGOING and paymentCounter=duration and
  safetyDepositPenalty<=price

  post: status=TERMINATED and

  landlords->forall(1 | 1.availableBalance = 1.availableBalance@pre +
  safetyDepositPenalty * 1.inventory->one(o|o.realty=realty).fractionOwned) and
  landlords->forall(1 | 1.realBalance = 1.realBalance@pre + safetyDepositPenalty *
  1.inventory->one(o|o.realty=realty).fractionOwned) and

  tenant.realBalance = tenant.realBalance@pre - safetyDepositPenalty
  and tenant.availableBalance = tenant.availableBalance@pre +
  (price-safetyDepositPenalty)
}
```

- Only an ONGOING contract in which all payments have been done can be TERMINATED;
- In the terminate function, the value of *safetyDepositPenalty* must be specified, indicating what part of the security deposit will be taken by the landlords (depending on the condition of the house);
- The remaining part of the security deposit will be returned to the tenant.

Comment:

Em relação ao *safetyDepositPenalty*, talvez a forma mais apropriada de fazer isto seria enviar também evidência de que a casa não está em condições e por isso o valor da caução deve ser debitado.

Request Early Termination

This function behaves differently depending on whether it is called by a landlord or by a tenant:

Unset

```
{context RentContract::requestEarlyTermination(months :Int)
  pre: status=ONGOING and paymentCounter<duration-months

  post: duration=duration@pre-months
}
```

- A rent contract can be terminated earlier without paying fees if both parties agree
- TODO - this one is hard to represent since there's currently no way to represent who is calling the function

Force Early Termination

This function behaves differently depending on whether it is called by a landlord or by a tenant:

Unset

```
{context RentContract::forceEarlyTermination_tenantVer(months :Int)
  let fee:Real = policies.earlyTerminationFee in

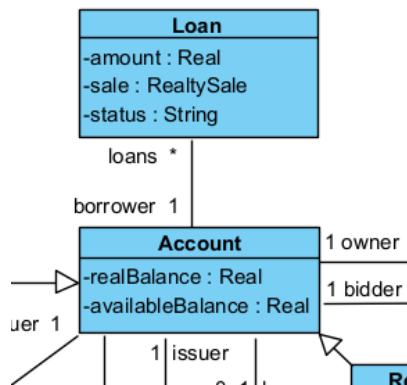
  pre: status=ONGOING and tenant.availableBalance >=fee and
  paymentCounter<duration-months and

  let monthsUntilTermination:Int = duration-months-paymentCounter in
  monthsUntilTermination > policies.earlyTerminationNotice

  post: duration=duration@pre-months and
  tenant.realBalance = tenant.realBalance@pre - fee and
  tenant.availableBalance = tenant.availableBalance@pre - fee and
  landlords->forall(1 | 1.availableBalance = 1.availableBalance@pre + fee *
  1.inventory->one(o|o.realty=realty).fractionOwned) and
  landlords->forall(1 | 1.realBalance = 1.realBalance@pre + fee *
  1.inventory->one(o|o.realty=realty).fractionOwned)
}
```

- A rent contract can be terminated earlier by the tenant if the tenant pays the fee established in the RentAgreement.
- The new termination date must be in the future and must respect the noticing time limit also defined in the RentAgreement;
- If successful, the duration of the contract is reduced by the number of months specified, and the fee value is distributed among the various landlords, in proportion to the ownership fraction acquired.

Concept Idea for Mortgage loans



Unset

```

{context Account
  inv: loans->select(1 | 1.status=AVAILABLE)->size() <= 1
}
  
```

And bid():

Unset

```

{context RealtySale::bid(bid : Bid)
  pre: status=OPEN and bid.bidder.availableBalance >= bid.amount
  pre: status=OPEN and (bid.bidder.availableBalance >= bid.amount or
  bid.bidder.loans->exists(1 | 1.sale=self and 1.status=AVAILABLE))

  post: self.bids->includes(bid) and bid.bidder.availableBalance =
  bid.bidder.availableBalance@pre - bid.amount and if bid.amount >= price then
  (postConditions da close) endif
  post: self.bids->includes(bid) and

  if bid.bidder.loans@pre->exists(1 | 1.sale=self and 1.status=AVAILABLE) then
  bid.bidder.loans->exists(1 | 1.sale=self and 1.status=AWAITING) and
  bid.bidder.availableBalance = bid.bidder.availableBalance@pre - bid.amount +
  bid.bidder.loans@pre->one(1 | 1.sale=self and 1.status=AVAILABLE).amount
  else bid.bidder.availableBalance = bid.bidder.availableBalance@pre - bid.amount
  endif and
  if bid.amount >= price then (close() postConditions) endif
}
  
```

- A user can bid if he has enough balance, or if he has an available loan for that property

- If there was a loan, then the loan status is changed to AWAITING, and the available balance of the bidder is decremented by the difference between the loan and the bid value
- If the bidder took no loan, then the available balance is decremented by the bid value
- If the value of the bid is equal or greater than the desired price for the realty sale, then the sale is automatically closed

TODO close()