Problem 2

```
a.) SEE PROBLEM 1 .USE FILE
b.) context Customer
        inv rentals:rentals->notEmpty
c.) context Customer
        inv daysRented:rentals->select(daysRented>3)->notEmpty



    1.) context LoyaltyAccount::points
                init:0
    2.) context LoyaltyProgram
                inv atLeastOneService: partners.deliveredServices->size()>=1
    3.) context LoyaltyProgram::getServices():Set(Service)
                body:partners.deliveredServices ->asSet()
    4.) context LoyaltyProgram::(getServices(pp: ProgramPartner):Set(Service)
                body: if partners->includes(pp)
                        then pp.deliveredServices
                        else Set{}
                        endif
    5.)
context Service
        inv:
transactions->select(oclIsTypeOf(Earning)).points->sum()>transactions->select(oclIsTypeOf(Burning)).points->sum()

context ProgramPartner
        inv:
deliveredServices.transactions->select(oclIsTypeOf(Earning)).points->sum()>deliveredServices.transactions->select(oclIsTypeOf(Burning)).points->sum()
    6.) context LoyaltyProgram
                inv: partners.deliveredServices->forAll(pointsEarned=0 and pointsBurned=0)
                implies Membership.account->isEmpty
    7.) context Customer::birthday()
        post: age = age@pre + 1
```