```
----------------List part----------------

sig List{
     code: Int,
     cars: some Car,
     pair: one PairOfWorkers,
}{
     code > 0
#cars > 0
}

fact uniqueList{
     all l1,l2: List| (l1!=l2) implies (l1.code!=l2.code)
}

fact phaseOfCarInTheList{
     all c: List.cars|
c.state.phase=parkUnsafeOrChargeOnSite
}

fact differentList{
     all l1,l2: List| all c: l1.cars| (l1!=l2) implies
!(c.isInList[l2])
}

fact listConstraints{
     all c:Car| (c.state.phase=parkUnsafeOrChargeOnSite)
implies (c in List.cars)
}

----------------Assertions part----------------

assert NoAllSafeAreaInSpacialeArea{
     no s:SafeArea | all p:s.safePositions| p in
SpecialArea.specialPositions
}

assert NoCarUsedAndInFreePhase {
     all c: Car |all u: User | (c.state.usedBy = u) implies
!(c.state.phase = free)
}
```