control:worker

```
#nbTask: int
#planning: Planning
```
```
+Worker()
+getNbTask()
+getPlanning()
+setPlanning()
+increaseTask()
+addTask()
+action()
+toString()
```

control::ControlCenter

```
- stationsList : Map<integer, Station>
- stationsNeedsToBeVerify: List< Station>
-nbStation : int
```
```
+ControlCenter()
+getStationList() : Map<integer,Station>
+getWorkerList()
+getVehiclesList()
+getVehiclesOnRoad()
+addStation(station : Station)
+removeStation()
+addWorker()
+removeWorker()
+addVehicle()
+removevehicle()
+addVehicleOnRoad()
+removevehicleOnRoad()
+initStation(): void
+verification(): void
+sendWorker(station : Station): void
+collectVehicles:
Map<integer,twoWheeledVehicle>
+redistribution(): void
+run(): void
+show(): void
```

repairer: Repairer

```
+Repairer()
+controlVehicle()
+verifyOtherVehicle()
+action() : void
+toString()
```

RepairerElectric

```
+RepairerElectric()
+toString()
```

Station

```
#vehicles : Map<twoWheeledVehicle,>
#capacityMax : int
#id : int
#OutService : Map<twoWheeledVehicle,>

#worker: Worker
#controlCenter: controlCenter
```
```
+Station( id : int)
+getId() : int
+getCapacityMax()
+getControlCenter()
+getVehicles() : Map<twoWheeledVehicle,>
+getOutService()
+getWorker()
+addVehicle(): twoWheeledVehicle
+removeVehicle()
+addVehicleOutService()
+removeVehicleOutService()
+addWorker()
+removeWorker()
+needRepairer()
+stole()
+initCapacityMax()
+displayVehicles()
+toString()
```

TwoWheeledVehicle
<<interface>>

```
+applySkin(string)
+getStation()
+setStation()
+getNbUse()
+resetNbUse()
+getUseLimit()
+setUseLimit()
+takeDamage()
+repair()
+stole()
+askRepairer()
+isGoodRepairer()
+toString()
+startRental()
+stopRental()
+isDamaged()
+isAvailable()
```

feature::FeatureDeco
<>

```
#twoWheeledVehicle : v
```
```
+FeatureDeco(TwoWheeledVehicle)
+decorate() : String
+setStation(station)
+applySkin(string)
+repair()
+isDamaged()
+takeDamage()
+askRepairer()
+isGoodRepairer()
+stole()
+startRental()
+stopRental()
+isAvailable()
+getStation()
+getNbUse()
+getUseLimit()
+setUseLimit()
+resetNBUse()
```

Planning

```
#Map<Station, Pair<Time, List<TwoWheeledVehicle>>> planning;
#List<Station> stationToVisit
```
```
+Planning()
+getPlanning()
+getStationToVisit()
+addStationToVisit()
+removeStationToVisit()
+addTask()
+removeTask()
+addPlanning()
+removePlanningAndStationToVisit()
```

Bike

```
#currentStation : station
# station : Station
#skin : String
#isDamaged: bool
#nbUse: int
#useLimit: int
```
```
+ Bike(station : Station, skin : String, int
ul)
+getStation() : Station
+setStation(station : Station)
+getNbuse()
+setNbUse()
+resetNbUse()
+getUseLimit()
+setUseLimit()
+applySkin(string)
+isDamaged()
+takeDamage()
+startRental()
+stopRental()
+isAvailable()
+askReapair()
+isGoodReapair()
+repair()
+stole()
+attachToStation()
```

PannierRack

```
+ PannierRack(TwoWheeledVehicle)

+Decorate() : String
```

Folding

```
+ Folding(TwoWheeledVehicle)

+Decorate() : String
```

Basket

```
+Basket(TwoWheeledVehicle)

+Decorate() : String
```

FakePlanning

```
+FakePlanning()
+addPlanning()
```

ElectricBike

```
#capacity
#niveauBatterie
```
```
+ ElectricBike()
+askRepairer()
+isGoodRepairer()
+getCapacityBatterie()
+setCapacityBatterie()
+getNiveauBatterie()
+setNiveauBatterie()
+rechargeBatterie()
+BatterieUse()
+BatterieFaible()
+toString()
```

Vlille

```
-ControlCenter: c
```
```
+Vlille(ControlCenter : c)
+getControlCenter()
+week(): void
+day(): void
+event(): void
+randomNB() int
+chance(): int
+displayVehiclesForEachStation():void
+initVehicles(): void
+run(): void
```