

```
GARAGE[G]+
feature -- basic queries
     -- Number of cars in garage
 cars: SET[CAR[G]]
     -- Set of cars parked in garage.
feature -- commands
 add (a car: CAR[G])
     -- Park 'a car' in the garage.
   require a car ∉ cars
   ensure cars = old cars \cup {a car}
feature -- advanced queries
 sorted cars: ARRAY[CAR[G]]
     -- sorted array of cars in the garage, by year
     \forall i \in 1.. Result.count-1: Result[i].vear \leq Result[i+1].vear
 old cars (a year: INTEGER): like cars
     -- return all cars older than 'a year'
     Result = \{c \in cars \mid c.year \le a \ year\}
 search car (a year: INTEGER): INTEGER
     -- return index to 'sorted cars' for car released in 'a year'
     -- if it exists, otherwise zero
    sorted: \forall i \in 1.. sorted cars.count - 1:
               sorted cars[i].year ≤ sorted cars[i+1].year
    no duplicates: \forall i, j \in \text{INTEGER}:
                       sorted cars[i].year = sorted cars[i]
                        \Rightarrow i = j
```

 \Rightarrow Result = 0

⇒ sorted cars[Result].year = a year

target_found: ∃car ∈ cars : car.year = a+year

consistent count: count = cars.count

invariant

```
odometer: INTEGER
```

feature -- { NONE } make (a model: G; a year: INTEGER; a miles: like odometer) -- Initialize a new car with 'a model', 'a year', and an odometer read 'a miles'.

CAR[G]+

feature -- queries

vear: INTEGER

model: G

consistent year: a year > 1900 positive miles: a miles ≥ 0

ensure

sorted cars: ARRAY[..]

proper initialization:

 $model \sim a \mod n$ year = a year \wedge odometer = a miles

feature -- commands

update odometer (a mileage: INTEGER)

-- Increase the odometer read by 'a mileage'.

positive increase: a mileage > 0

correct update: odometer = old odometer + a mileage

invariant

```
valid car year: year >= 1900
valid mileage: odometer \geq = 0
```



```
feature -- queries
```

battery: INTEGER

feature -- { NONE }

make (a model: G; a year: INTEGER; a miles: like odometer)

-- Initialize a new car with 'a model', 'a year', and an odometer read 'a miles'.

battery set: battery = 26

feature -- commands

upgrade battery (a new battery: like battery)

-- Upgrade kilowatt hours.

require

a new battery > battery + 10

ensure

battery = a new battery

valid battery: battery $\geq = 26$