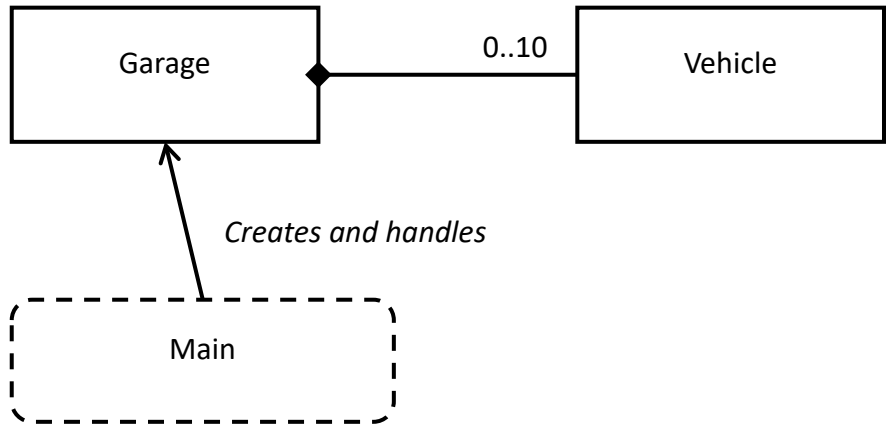
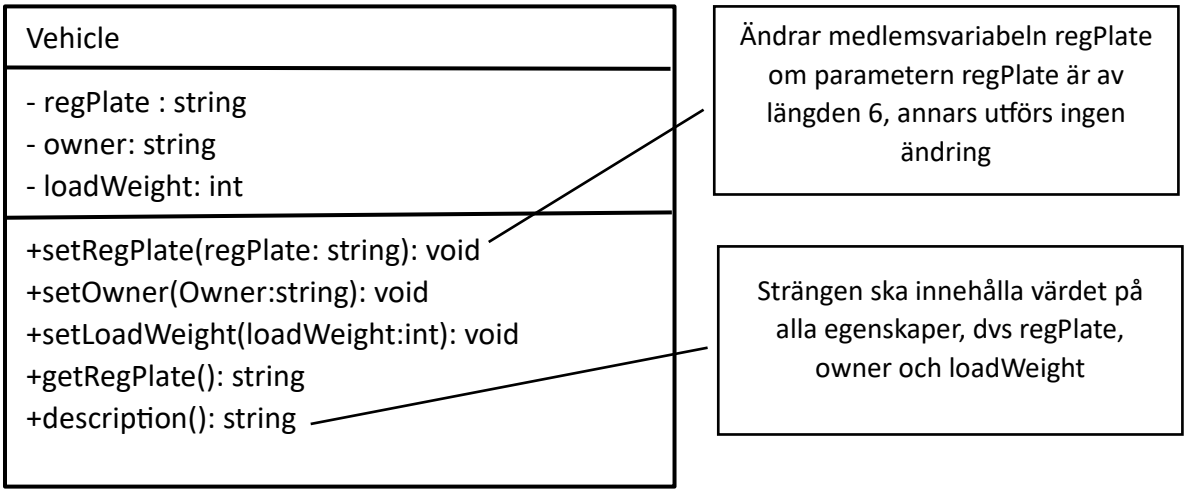
Övergripande klassdiagram

Du ska implementera klasserna ***Vehicle*** och ***Garage*** enligt relationen i klassdiagrammet:



Detaljerade klassdiagram för respektive klass

Vehicle



Det vara möjligt att

- skapa ett Vehicle-objekt givet *regPlate, owner* och *loadWeight*
- skapa ett Vehicle-objekt utan att bifoga något
- jämföra Vehicle-objekt genom användande av <operatorn vilken ska baseras på *loadWeight*
- jämföra Vehicle-objekt genom användande av ==operatorn vilken ska baseras på *regPlate, owner* och *loadWeight*

Alla medlemsvariabler ska vara **privata**.

I övrigt får inga andra funktioner eller variabler tillföras i klassen.

Testprogrammet **VehicleTest.cpp** ska användas för att testa din implementation av Vehicle - klassen.

Garage

Garage
- name: string - currentNrOfVehicles: int
+ getName(): string + getCurrentNrOfVehicles (): int + findVehicle(regPlate: string): int + addVehicleAt(regPlate: string, owner: string, loadWeight:int, index:int): bool + removeVehicleAt(index: int): bool

Relationen mellan Garage och Vehicle ska implementeras genom användande av en **array innehållande pekare (antingen statiskt allokerad eller dynamiskt allokerad)**

Vid sökning (funktionen findVehicle) returneras det index i arrayen som det fordonet med registreringsnummer som motsvarar regPlate finns på. Om det inte finns returneras -1.

Vid tillägg (funktionen addVehicleAt) placeras det nya fordons-objektet på det index i arrayen som anges i parametern ***index*** om det är ledigt och vidare returneras true. Om det upptaget returneras i stället false.

Vid borttagning (funktionen removeVehicleAt) tas det fordons-objekt bort som finns på det index i arrayen som anges i parametern ***index*** och true returneras. Om det inte finns något fordons-objekt på det aktuella indexet returneras false i stället.

Vidare ska det vara möjligt att

- skapa ett Garage-objekt givet *name*
- skapa ett garage-objekt utan att bifoga något

För testningen krävs dessutom medlemsfunktionen **Vehicle* VehicleAt(int index)** som ska returnera adressen/pekaren till det Vehicle-objekt som finns i arrayen på det index som parametern motsvarar. Om det inte finns något objekt ska nullptr returneras.

Du behöver **tillföra medlemsvariabel/medlemsvariabler för att implementera relationen** till klassen Member. Du behöver även tillföra destruktor.

Alla medlemsvariabler ska vara **privata**.

I övrigt får inga andra funktioner eller variabler tillföras i klassen.

Din implementation får **inte generera några minnesläckor**.

Testprogrammet **GarageTest.cpp** ska användas för att testa din implementation av Garage-klassen.