A Ferry System

Your task is to create a programming system for a ferry. The ferry transports passengers and vehicles (cars, busses, lorries and bicycles). The ferry has space for 200 passengers and 50 cars. A lorry needs as much space as two busses or 8 cars. A car needs as much space as 5 bicycles. There are different fees for different vehicles and an extra fee might be added for passengers. Use the following fees:

common attributes for Vechiles-> fee(vehicle,passenger),max-passanger,space.
Example-
Car(vehicle fee=100kr, passgerfee=15, max-passenger=4 and size=5
Bus vehicle fee=200kr, passgerfee=10, max-passenger=20 and size=20
Lorry vehicle fee=300kr, passgerfee=15, size=40, max-passenger=2 and size=20
Bicycle vehicle fee=50kr, passgerfee=0, max-passenger=1 and ize=1

1. Passenger without vehicle, 25 kr.
2. Bicycle 50 kr (passenger included).
3. Car 100 kr + 15 kr/passenger (maximum 4 passengers).
4. Bus 200 kr + 10 kr/ passenger (maximum 20 passengers).
5. Lorry 300 kr + 15 kr/ passenger (maximum 2 passengers).

Every type of vehicle (car, bus, lorry, bicycle) will inherit from the class Vehicle. The functionality of the ferry is given by the interface Ferry:

```
public interface Ferry extends Iterable<Vehicle>  {
   int countPassengers();             // Number of passengers on board
   int countVehicleSpace();           // Used vehicle space. One car is 1.
   int countMoney();                  // Earned money
   void embark(Vehicle v);            // Embark vehicle, warning if not enough space
   void embark(Passenger p);          // Embark passenger, warning if not enough room
   void disembark();                  // Clear (empty) ferry. The money earned remains,
                                      // i.e., is not reset to zero
   boolean hasSpaceFor(Vehicle v);    // true if we can embark vehicle v
   boolean hasRoomFor(Passenger p);   // true if we can embark passenger p
   String toString();                 // Nice looking ferry status print out

}
```

A vehicle cannot leave the ferry until the ferry has been disembarked and the same vehicle cannot embark twice. The ferry iterator should iterate over all vehicles embarked (not the passengers). Also write a program FerryMain.java, embarking a number of vehicles and passengers, showing the functionality of the methods.

Hint: It might be a good idea to internally (inside the Ferry class) use a bicycle as the basic space unit.