Übung: 1 - Smarte Vererbung

Über einen eShop werden smart Autos verkauft. Es stehen dabei smarts in verschiedenen Varianten zur

Karosserie:

smart city-coupé	Grundvariante
smart cabrio	Grundvariante mit stufenlos verstellbarem Textilverdeck

Linie / Ausstattungsmöglichkeiten:

smart & **pure**	
smart & **pulse**	temperamentvolle Variante (z.B. stärkerer Motor)
smart & **passion**	edle Variante (luxoriöse Ausstattungen)

1. Modellieren Sie die sechs Varianten (Kombinationen aus Karosserie und Linie) als Klassen und benutzenSie dazu (wo sinnvoll) Vererbung und (statische) Attribute und Methoden. Beschreiben Sie Ihre Lösung mit einem kleinen UML Klassendiagramm. Ihre Implementierung soll die folgenden Aspekte berücksichtigen:
   - Jedes smart-Auto kann in einer beliebigen Farbe geliefert werden. Beim Erzeugen einer neuen Instanz soll die gewünschte Farbe angegeben werden. Diese soll jederzeit abgefragt aber nicht geändert werden können. Typ: java.awt.Color
   - Jeder Smart kann als Benziner oder als Dieselvariante geliefert werden (Aufpreis Diesel: CHF 1200).
   - Beim Smart cabrio kann zusätzlich ein Radio+Kasettengerät eingebaut werden (Aufpreis CHF 460). DasRadio kann auch nachträglich ein- und ausgebaut werden.
   - Es gelten folgende Preise:

smart city-coupé pure:	CHF 14390
smart city-coupé pulse:	CHF 15590
smart city-coupé passion:	CHF 16790

   - Die cabrio Varianten kosten generell CHF 3100 mehr.
   - Der Verkaufspreis/Wert eines Autos hängt von den jeweiligen Ausstattungen ab und soll für jede Instanz abgefragt werden können (Typ: int).
   - Jedes Auto soll eine eindeutige Seriennummer erhalten (Typ: int) die über die Lebensdauer des Autos nicht geändert werden darf. Bei der Erzeugung einer neuen smart-Instanz soll dieser eine eindeutige Seriennummer zugewiesen werden. Sie soll über eine Methode abgefragt werden können.
   - Es soll von aussen weder auf die Klassen- noch auf die Instanzvariablen direkt zugegriffen werden können.
2. Implementieren Sie in Ihren Klassen die Methode *toString* aus der Klasse *Object* welche einen smart als Stringrepräsentation zurückgibt.
3. Implementieren Sie in Ihren Klassen die Methode *equals* aus der Klasse *Object*. Überlegen Sie sich, wann zwei smart-Objekte als identisch betrachtet werden sollen. Geben Sie dann eine zu *equals* konformeImplementierung der Methode *hashCode* an.
4. Schreiben Sie ein Testprogramm, das einige smart Autos in verschiedenen Varianten (smart city-coupé pure, smart cabrio pure, ...) erzeugt und sie in einer Menge (*java.util.Set*) ablegt.Traversieren Sie anschliessend alle Elemente der Menge und geben Sie für jedes Auto den Typ, die Seriennummer, die Verkaufskosten sowie Angaben zur Ausstattung aus.
5. Optional kann zusätzlich eine Maske implementiert werden, mit welcher neue Instanzen erzeugt und existierende angezeigt werden können.