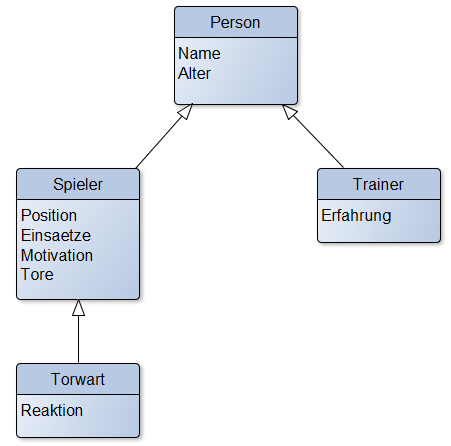
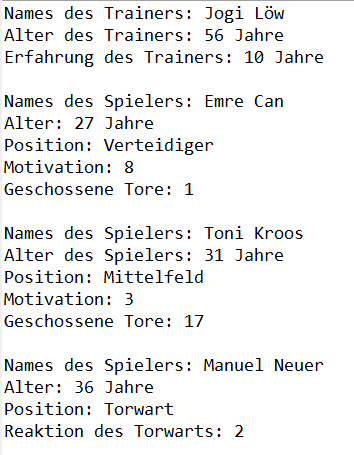
**Geführte Aufgabe zu Vererbung**

**Ziel:**

Sie sollen die folgende Klassenhierarchie in Java implementieren, die eine Spezialisierung der (Super-) Klasse „Person“ zu den (Sub-) Klassen „Spielern“ und „Trainer“ durch zusätzliche Eigenschaften beschreibt. Danach erzeugen Sie noch die Klasse „Torwart“, die wiederum eine Subklasse von „Spieler“ ist.



1. Legen Sie als erstes die (nicht ausführbare) Klasse „Person“ an. Die Attribute dieser Klasse (im Beispiel „name“ und „alter“) sollen gemäß dem Prinzip des „***Data Hiding***“ nur aus den Programmteilen heraus ansprechbar sein.
2. Erweitern Sie die Klasse „Person“ um einen allgemein zugänglich (parametrisierten) Konstruktor für eine einfachere Initialisierung eines Objekts mittels der Attribute „name“ und „alter“. Die dem Konstruktor übergebenen Argumente sollen genauso heißen, wie die Attribute!
3. Um die Werte der (verborgenen) Attribute dennoch zugänglich zu machen, formulieren Sie für jedes Attribut zwei „***Auskunftsmethoden***“ („Accessor-Methode“). Diese Getter/Setter-Methoden können Sie sich von Eclipse automatisch erstellen lassen.
4. Entwerfen Sie anschließend die beiden neuen, ebenfalls nicht ausführbaren Klassen „Spieler“ und „Trainer“. Beide Klassen sollen Subklassen der Superklasse „Person“ sein.
5. Mit dem Attribut „Position“ soll angegeben werden, ob es sich um einen Stürmer, Mittelfeldspieler oder Abwehrspieler handelt. Die drei anderen Attribute sind ganzzahlige Werte, wobei „Einsaetze“ die Anzahl der Einsätze in der Nationalmannschaft angibt und „Motivation“ einen Wert zwischen 0 und 10 aufweisen kann. Auch auf diese Attribute soll nur mittels Getter/Setter-Methoden zugegriffen werden können.
6. Nachdem Sie die Attribute für die Klasse „Spieler“ deklariert haben, verfassen Sie einen Konstruktor. Da ein Konstruktor der Sub-Klasse neintben den Eigenschaften seiner eigenen Klasse auch die Attribute der Basis-Klasse als Übergabe-Parameter definiert (im Beispiel Name, Alter), müssen Sie diesem Konstruktor insgesamt sechs Werte übergeben (nämlich Name, Alter, Position, Einsaetze, Motivation und Tore). Um auf den „***THIS***“ Operator im Body des Konstruktors zu verzichten, wählen Sie Übergabeparameter, die sich von der Bezeichnung her von den Attributnamen unterscheiden. Sie können dazu beispielsweise nur jeweils den Anfangsbuchstaben des Attributs benutzen.
7. Denken Sie im Body des Konstruktors daran, als erstes den Konstruktor der Basis-Klasse (***Super***klasse) anzusprechen. Danach folgt die Zuweisung der in der Subklasse definierten Attribute. Lassen Sie sich jetzt von Eclipse die zugehörigen Getter/Setter-Methoden erstellen.
8. Nun schreiben Sie sich analog der Klasse „Spieler“ noch eine Klasse „Trainer“ sowie eine Klasse „Torwart“, wobei letztere eine Subklasse von „Spieler“ ist.
9. Zum Schluss brauchen wir noch eine ausführbare Klasse „Test“. Darin erzeugen Sie sich vier Objekt, nämlich einen Trainer (Name des Objekts: „loew“), zwei Spieler („can“, „kroos“) sowie einen Torwart („neuer“). Die Angaben zu den vier Personen entnehmen Sie der folgenden Abbildung, die auch Ihre Ausgabe nach Aufruf der Klasse „Test“ sein soll.



**Ziel:**

Entwerfen Sie eine Klasse „Fahrzeuge“. Ein Fahrzeug hat einen Namen und eine Farbe, die auch beide bei der Generierung eines Objekts mit angegeben werden müssen. Außerdem soll es eine Methode „zeigeEigenschaften()“ geben, in der die beiden Attribute der Instanz angezeigt werden.

Daneben existiert die Klasse „Motorrad“, die von der Basisklasse „Fahrzeug“ erbt und über zwei weitere Attribute verfügt. So kann man bei einem Motorrad angeben, ob es eine Satteltasche hat und über wie viel PS es verfügt. Zur Instanziierung eines Motorrads müssen stets vier Parameter übergeben werden.

Die Methode „zeigeEigenschaften()“ der Klasse „Motorrad“ überschreibt die Methode der Basisklasse „Fahrzeuge“.

In einer Klasse „Fuhrpark“ sollen von jeder Klasse eine Instanz (nennen Sie die Instanzen f1 und m1) ge-bildet werden und die Eigenschaften mit Hilfe der Methode „zeigeEigenschaften()“ ausgegeben werden.

Verwenden Sie die in der folgenden Ausgabe/Darstellung benutzten Attribute.

