

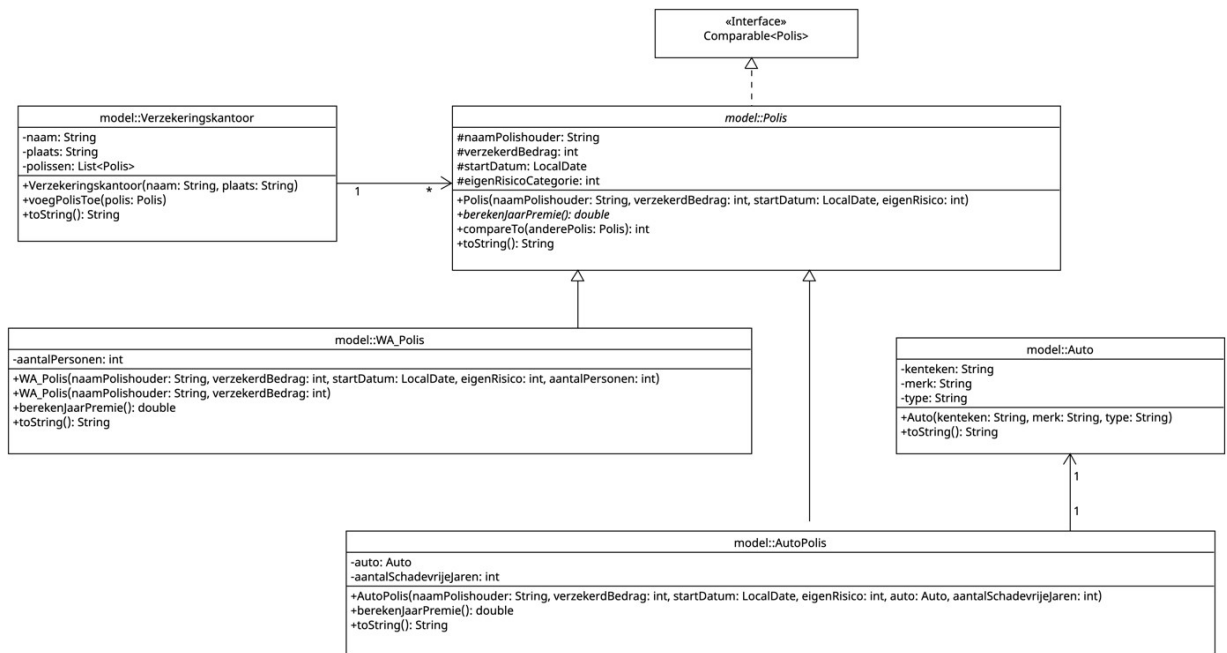
Sigorta şirketi OOP

Giriş

Sigorta şirketi Snits dijitalleşecek. Poliçe ve tazminat taleplerine ilişkin verileri kaydetmek için bir yazılım geliştirilmesi gerekmektedir. Bu ödevin amacı, uygulama için gerekli sınıfları oluşturmaktır.

Açıklama

Aşağıdaki sınıf diyagramının oluşturulması gerekmektedir. Bu, ödev 2'deki veritabanı tasarımının bir parçasıdır. Açıkçası, yinelenen kodlardan kaçınılmalıdır.



Genel talimatlar

- DLO'dan AssessmentInsurance Office başlangıç projesini alın. Paket denetleyicisinde bir Launcher sınıfı ve bir ana yöntem vardır ve paket modelinde zaten Auto sınıfı vardır.
- Başlatıcıyı kendi kodunuzu test etmek için kullanabilirsiniz. Başlatıcıda göstermeniz **gereken** kod için bu belgenin sonuna bakın. Diğer kodu göstermek zorunda değilsiniz, ancak Başlatıcı'da iyi çalışan bir kod bırakın. Kodunuzu düzenli olarak test etmenizi öneririz!
- Başlatıcıda, araçların bir listesini oluşturmak için kod vardır. Test için listedeki araçları kullanabilirsiniz ve ayrıca listeyi kullanan bir komut da vardır.
- Sınıf diyagramı alıcıları ve ayarlayıcıları içermez. İhtiyacınız olduğunu düşündüğünüz getter(lar)ı ve setter(lar)ı kendiniz ekleyin. Varsayılan olarak tüm getter ve setter'ları değil, yalnızca ihtiyacınız olanları ekleyebilirsiniz.
- `toString()` metotlarınızın çıktı örneklerinde olduğu gibi aynı çıktıyı ürettiğinden emin olun.
- Tarihlerin veri türü olarak `LocalDate` sınıfını kullandığını unutmayın. Böylece, geçerli tarih için bir `LocalDate` nesnesinin `now()` gibi yöntemlerini rahatlıkla kullanabilirsiniz.

- Elbette, Comparable arayüzü Java'da zaten mevcuttur, bu nedenle bunu kendiniz oluşturmanıza gerek yoktur.
- Sınıf diyagramında olmayan şeyleri kendiniz eklemek istiyorsanız (gerekli olduğunu düşündüğünüz için), bunu koddaki yorumlarla açıkça belirtin.

Soyut sınıf Polis

1. Varsayılan olarak, tarih ABD biçiminde (yyyy-mm-dd) görüntülenir. Bunun için `LocalDate` sınıfının `toString()` yöntemini kullanabilirsiniz.
2. Aşım kategorisi 1 (Düşük), 2 (Orta) veya 3 (Yüksek) seçilebilir. Aşım tutarı poliçe türüne göre belirlenir ve bu atamada bir rol oynamaz. Bu değerlerden birinden farklı bir aşım kategorisi girilirse bir hata mesajı verin ve bu poliçenin aşım kategorisini 1 olarak ayarlayın.
3. `compareTo()` yönteminin ilkeleri tarihe göre sıraladığından emin olun. Bunu yapmak için `LocalDate` sınıfının `compareTo()` yöntemini kullanın.
4. `toString()` yönteminin örneğin aşağıdaki çıktıyı döndürdüğünden emin olun:

```
O.B. Bommel adına poliçe Sigorta
tutarı: 10000 euro Başlangıç
tarihi: 2020-07-31
Kendi risk kategorisi: 2
```

Alt Sınıf WA_Policy

1. Yapıcıları mümkün olduğunca yinelenen kodlardan kaçınacak şekilde uygulayın.
2. Yalnızca poliçe sahibinin adı ve sigorta tutarı verilirse, başlangıç tarihi geçerli tarihe, aşım (kategori) 1'e ve kişi sayısı 1'e ayarlanmalıdır. Geçerli tarih için `LocalDate` sınıfının `now()` yöntemini rahatlıkla kullanabilirsiniz.
3. `calculateYearPremium()` yöntemini geçersiz kılın ve aşağıdaki gibi uygulayın:
 - a. Bir üçüncü taraf poliçesinin yıllık primi, sigorta bedelinin bir yüzdesidir. Bu temel tutardır. Bu yüzde, aşağıdaki tabloya uygun olarak seçilen aşım oranına bağlıdır:

Kategori	Yüzde
1	0,06%
2	0,05%
3	0,04%

- b. Yıllık prim, temel tutarın kişi sayısı ile çarpımıdır.
4. `toString()` yöntemini geçersiz kılın ve aşağıdaki örnekte olduğu gibi uygulayın. Yinelenen kodlardan kaçının ve mümkün olduğunca yeniden kullanın.
 5. Örneğin, `toString()` yönteminin aşağıdaki çıktıyı döndürdüğünden emin olun (sigortalı tutar 1 milyon Euro'dur):

```
Tom Poes adına poliçe Sigortalı
tutar: €1000000 Başlangıç tarihi:
2021-09-01
Kendi risk kategorisi: 1 Kişi
sayısı: 4
Yıllık prim: €2400.00
```

Oto poliçe alt sınıfı

1. Tazminatsız yıl sayısı minimum 0 ve maksimum 40'tır. Tazminatsız yıl sayısı bunu karşılamıyorsa ilgili hata mesajını verin ve ardından tazminatsız yıl sayısını 0 olarak ayarlayın. (Bkz. Programlama komutu: Tazminatsız yıl sayısı en az 0 olmalıdır! sırasıyla Hasarsız yıl sayısı maksimum 40 olmalıdır!)
2. `calculateYearPremium()` yöntemini geçersiz kılın ve aşağıdaki gibi uygulayın:
 - a. Bir otomobil poliçesinin yıllık primi, sigorta bedelinin bir yüzdesidir. Bu temel tutardır. Bu yüzde, aşağıdaki tabloya göre seçilen fazlalığa bağlıdır:

Kategori	Yüzde
1	10%
2	9%
3	8%

- b. Hasarsız yıl sayısına bağlı olarak bir indirim verilir. Bu indirim, hasarsız yıl başına yıllık prim üzerinden %5'tir ve maksimum %70'tir. Yani her zaman temel tutarın en az %30'unu ödersiniz.
3. `toString()` yönteminin örneğin aşağıdaki çıktıyı döndürdüğünden emin olun:

```
Wammes Waggel adına poliçe Sigorta
tutarı: 10000 euro Başlangıç
tarihi: 2021-07-31
Kendi risk kategorisi: 2
Araç: 74-OBB-3 tescil numaralı Volkswagen Beetle
Hasarsız yıl sayısı: 8
Yıllık prim: 540,00 Euro
```

Class Sigorta Ofisi

1. Bir sigorta ofisi poliçeleri yönetir.
2. `addPolicyAdd()` yöntemi, bir sigorta ofisine poliçe eklemenizi sağlar.
3. `toString()` yönteminin örneğin aşağıdaki çıktıyı döndürdüğünden emin olun:

```
Rommeldam'daki Bommelstein Sigorta Ofisi'ndeki poliçeler: Tom
Poes adına poliçe
  Sigorta tutarı: €1000000
  Başlangıç tarihi: 2021-09-01
  Kendi risk kategorisi: 1 Kişi
  sayısı: 4
  Yıllık prim: €2400.00

Wammes Waggel adına poliçe Sigortalı
  tutar: 10000 euro Başlangıç
  tarihi: 2021-07-31
  Kendi risk kategorisi: 2
  Araç: 74-OBB-3 tescil numaralı Volkswagen Beetle
  Hasarsız yıl sayısı: 8
  Yıllık prim: 540,00 Euro
```

4. `toString()` yöntemini, poliçeleri başlangıç tarihi sırasına göre gösterecek şekilde değiştirin.

Bu kodun Başlatıcıda mevcut olduğundan emin olun:

1. Adınızı ve öğrenci numaranızı içeren bir hoş geldiniz mesajı verin
2. Girilen aşım kategorisinin yanlış olması durumunda bir poliçenin hata mesajı verdiğini gösterin.
3. Bir araba poliçesinin hasarsız yıl sayısı çok düşükse bir hata mesajı verdiğini gösterin.
4. Bir araba poliçesinin hasarsız yıl sayısının çok yüksek olması durumunda hata mesajı verdiğini gösterin.
5. Bir sigorta ofisine poliçe ekleyebileceğinizi gösterin, en az üç poliçe ekleyin.
6. Tüm `toString()` yöntemlerinin doğru şekilde uygulandığını gösterin.
7. Launcher'a belirli bir türdeki arabaları yazdıran bir yöntem ekleyin. Yönteme `public static void showTypeAutos(List<Auto> autoList, String type)` imzasını verin. `autoList`'ten tüm 'elektrikli' araçları görüntülemek için yöntemi çağırın.

Kod kurallarına dikkat edin

1. Adınızın ve hedefinizin her sınıfın en üstünde yer aldığından emin olun (ICC #1).
2. Düzenle birlikte doğru girintiyi kullanın (ICC #2).
3. Büyük ve küçük harflerin doğru kullanımına dikkat edin (ICC #3).
4. İyi isimler kullanın (ICC #4).
5. Sihirli sayılardan kaçının, bu nedenle sabitleri kullanın (ICC#5).
6. Gerektiğinde kodunuz hakkında fikir veren yorumlar ekleyin (ICC#7).
7. Ölü koddan kaçının (ICC #8).
8. Kapsüllemeyi düşünün (ICC #9).