

Kurumsal Varlıkların Hibernate ile Modellenmesi

Fehmi Can SAĞLAM
10/2010

Amaç

- Geniřletilebilir/sürdürülebilir kurumsal uygulamalar elde etmek için taşınır, kullanıcı/müşteri, dosya, ürün, yazılım, sunucu gibi kurumsal varlıkların Hibernate ile etkin olarak modellenmesi

Ana Hatlar

- Java ile kalıtım(inheritance) ve çok biçimlilik(polymorphism)
- Hibernate Çatısının Kalıtım Eşleme(inheritance mapping) Özellikleri
- Varsayımsal bir kurumun varlıklarının modellenmesi

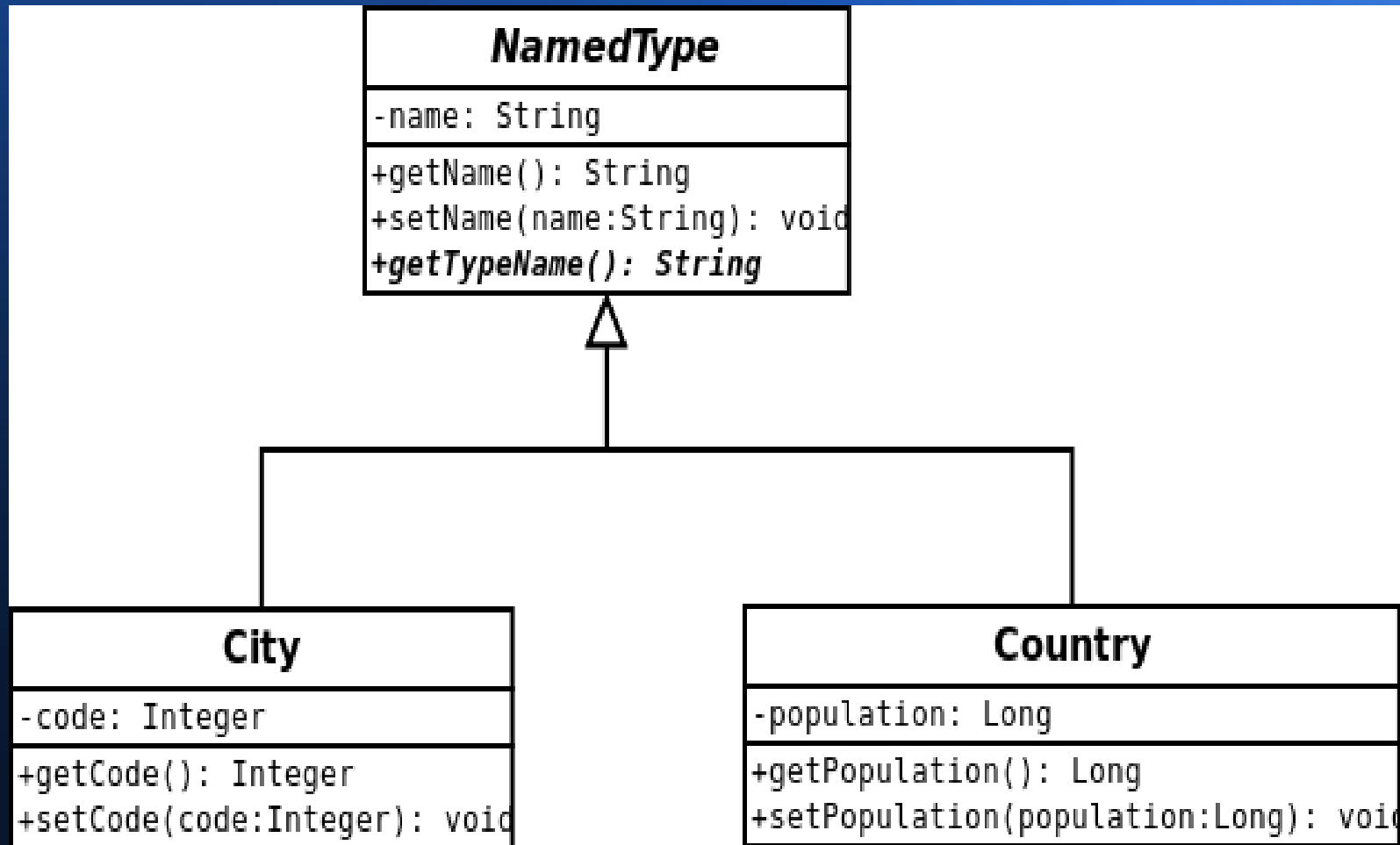
Kalıtım(Inheritance)

- Kalıtım bir sınıfın başka bir sınıfın üyelerini ve metodlarını kendi işlevlerini de ekleyerek kullanabilmesidir.
- Bu özelliğin kullanışlı olabileceği bir duruma örnek olarak bir şekil kütüphanesi verilebilir. Tüm şekiller için ortak olan özellikleri genel bir sınıfta toplayıp çember, dikdörtgen gibi özel şekiller için ayrı sınıflar oluşturabilirsiniz.

Kalıtım(Inheritance)

- Genel sınıf taban sınıf olarak adlandırılırken, özel sınıflara alt sınıflar denir.
- Kalıtım kodun yeniden kullanılabilirliğini büyük oranda artırırken tasarımın daha basit ve anlaşılır olmasını sağlar.

Örnek



Hibernate Kalıtım Eşleme

- Table per class hierarchy
- Table per subclass
- Table per concrete class

Table per class hierarchy

- Yalnızca 1 tablo gereklidir.
- Taban sınıf ve alt sınıflara ait tüm alanlar aynı tabloda yer alır.
- Bu strateji, alt sınıflar tarafından tanımlanan alanların NOT NULL constraint sahibi olmasını engeller.

Table per subclass

- Taban sınıf ve alt sınıfların toplam sayısı kadar tablo gereklidir.
- Alt sınıf tabloları ile üst sınıf tablosu arasında primary key ilişkisi vardır. Başka bir deyişle ilişkisel model one-to-one ilişkidir.
- En normalize veritabanı şeması bu yöntemle elde edilir.
- Bir alt sınıfı okumak bir ya da daha fazla join işlemi gerektirir.

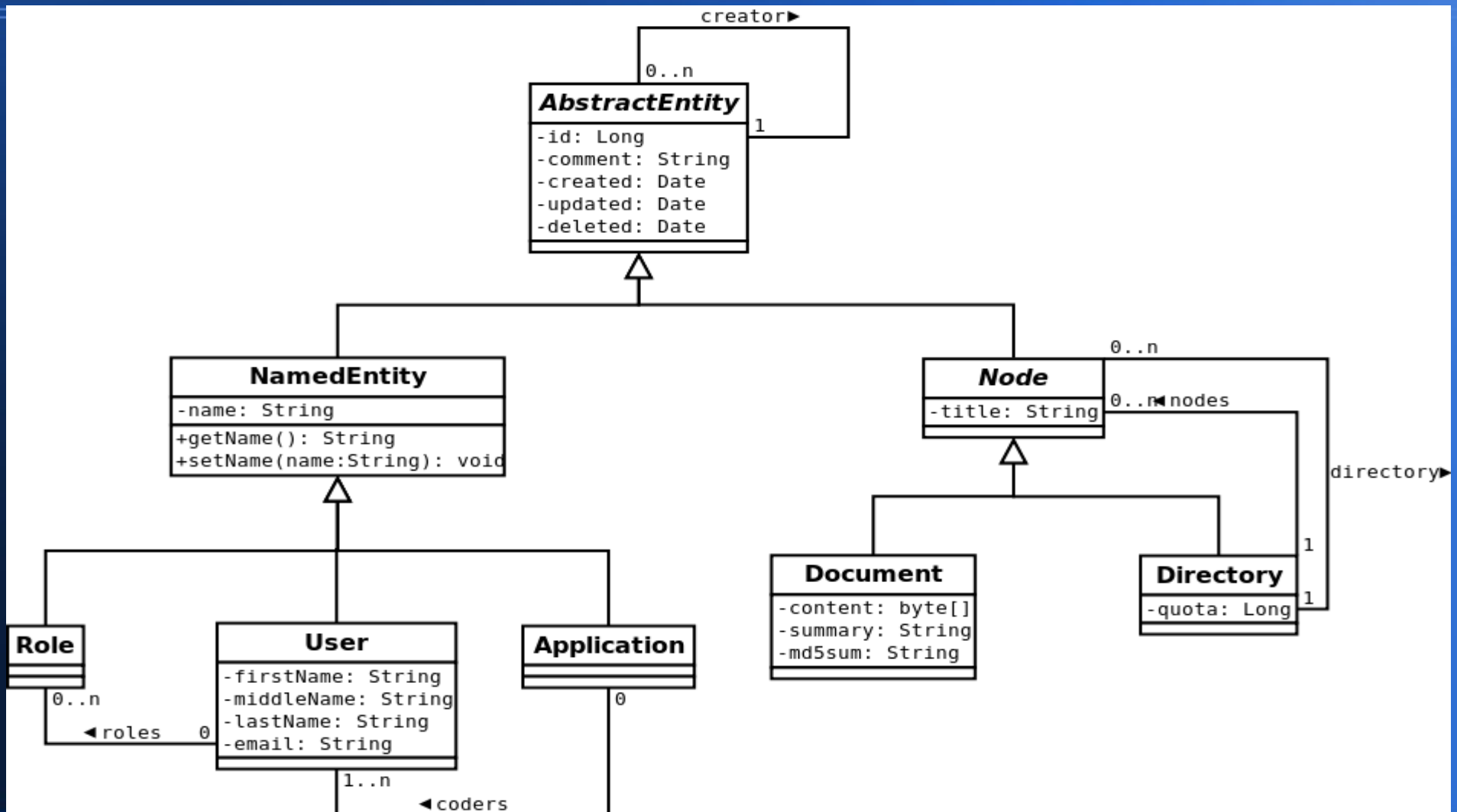
Table per concrete class

- Her bir sınıf için 1 tabloya ihtiyaç vardır.
- Her tabloda miras alınan alanlar da dahil olmak üzere alt sınıfın alanları bulunur.
- Alt sınıflar üzerinde join işlemine gerek olmaksızın çok hızlı işlem yapılabilir.
- Alt sınıf bilinmediğinde aranan nesne herhangi bir alt sınıf tablosunda olabileceğinden karmaşık union sorguları ortaya çıkabilir.

Uygulama

- Her kurumsal varlığın bir ID'si vardır.
- Hiç bir kurumsal varlık gerçekten silinmez.
- Her kurumsal varlığın yaratıcısı, güncelleyeni ve silenini vardır.
- Her kurumsal varlığın yaratılma, güncelleme ve silinme tarihi vardır.

Uygulama



Teşekkürler

- Sorular?