Java 9 ile gelen yenilikler

Öğrenci : Ozan Aydoğan
Öğretmen : Halit Mızrak

Java 9 ile gelen yenilikler

1. **The Java Platform module system**
2. **Linking**
3. **JShell: the interactive Java REPL**
4. **Improved Javadoc**
5. **Stream API improvements**
6. **Private interface methods**

The Java Platform module system

Java 9 için tanımlayıcı özellik, tamamen yeni bir modül sistemidir. Kod büyüdükçe, karmaşık kod yani "spagetti kodu" oluşturma olasılığı katlanarak artar. İki temel sorun vardır: Kodu gerçekten kapsüllemek zordur ve bir sistemin farklı bölümleri (JAR dosyaları) arasında açık bir bağımlılık kavramı yoktur. Her genel sınıfa, sınıf yolundaki diğer herhangi bir genel sınıf tarafından erişilebilir, bu da genel API olması amaçlanmayan sınıfların yanlışlıkla kullanılmasına yol açar. Ayrıca, sınıf yolunun kendisi sorunludur: Gerekli tüm JAR'ların orada olup olmadığını veya yinelenen girişler olup olmadığını nasıl anlarsınız? Modül sistemi her iki sorunu da ele alır.

The Java Platform module system

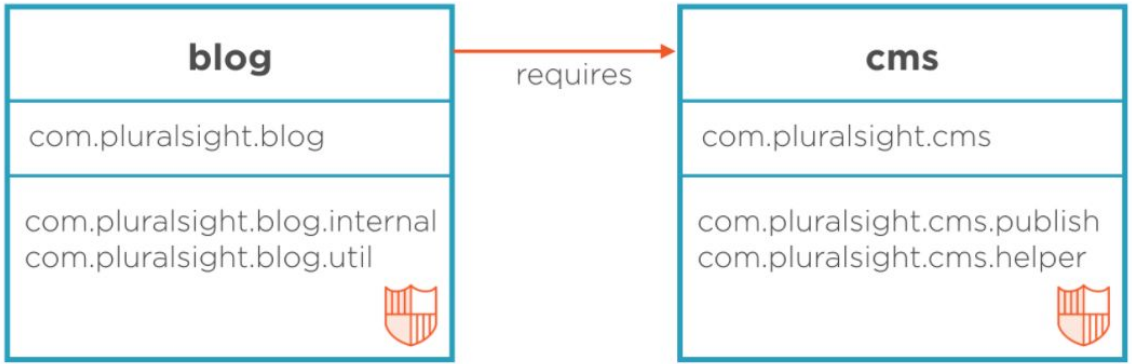
Modüler JAR dosyaları ek bir modül tanımlayıcı içerir. Bu modül tanımlayıcıda, diğer modüllere olan bağımlılıklar "requires" ifadeleri ile ifade edilir. Ek olarak, 'exports' ifadeleri, hangi paketlerin diğer modüller tarafından erişilebilir olduğunu kontrol eder. Dışa aktarılmayan tüm paketler varsayılan olarak modülde kapsüllenir. İşte "module-info.java" içinde yaşayan bir modül tanımlayıcı örneği:

The Java Platform module system

```
module blog {
  exports com.pluralsight.blog;

  requires cms;
}
```

We can visualize the modules as follows:



The Java Platform module system

Her iki modülün de dışa aktarılmadığı için kapsüllenmiş paketler içerdiğini unutmayın (turuncu kalkanla görselleştirilmiştir). Hiç kimse bu paketlerdeki sınıfları yanlışlıkla kullanamaz. Java platformunun kendisi de kendi modül sistemi kullanılarak modüler hale getirilmiştir. JDK dahili sınıflarını kapsülleyerek, platform daha güvenli ve geliştirilmesi çok daha kolay hale gelir.

The Java Platform module system

Modüler bir uygulama başlatırken, JVM tüm modüllerin "requires" ifadelerine göre çözülüp çözülemeyeceğini doğrular . Modüller, güçlü kapsülleme ve açık bağımlılıklar uygulayarak uygulamanızı daha iyi yapılandırmanıza olanak tanır.

Linking

Açık bağımlılıkları olan modülleriniz ve modülerleştirilmiş bir JDK'nız olduğunda, yeni olasılıklar ortaya çıkar. Uygulama modülleriniz artık diğer uygulama modüllerine ve JDK'dan kullandığı modüllere bağımlılıklarını belirtir. Neden bu bilgileri, yalnızca uygulamanızı çalıştırmak için gerekli olan modülleri içeren minimum bir çalışma zamanı ortamı oluşturmak için kullanmıyorsunuz? Bu, Java 9'daki yeni jlink aracıyla mümkün oldu. Uygulamanızı tam yüklü bir JDK kurulumuyla göndermek yerine, uygulamanız için optimize edilmiş minimum bir çalışma zamanı görüntüsü oluşturabilirsiniz.

JShell: the interactive Java REPL

Pek çok dilde etkileşimli bir Okuma-Değerlendirme-Baskı-Döngüsü bulunuyor ve Java şimdi bu kulübe katılıyor. Konsoldan jshell'i başlatabilir ve doğrudan Java kodunu yazmaya ve yürütmeye başlayabilirsiniz. jshell'in anında geri bildirimi, onu API'leri keşfetmek ve dil özelliklerini denemek için harika bir araç haline getirir.

Improved Javadoc

Javadoc artık aramayı doğrudan API belgelerinin kendisinde içeriyor. Ek bir bonus olarak, Javadoc çıktısı artık HTML5 uyumludur. Ayrıca, her Javadoc sayfasının, sınıfın veya arayüzün hangi JDK modülünden geldiğine dair bilgiler içerdiğini fark edilebilir.

Stream API improvements

Streams API, tartışmasız Java standart kütüphanesinde uzun zamandır yapılan en iyi geliştirmelerden biridir. Koleksiyonlarda bildirime dayalı dönüşüm hatları oluşturmanıza olanak tanır. Java 9 ile bu sadece daha iyi hale geliyor. Akış arabirimine eklenen dört yeni yöntem vardır: dropWhile, takeWhile, ofNullable. Yineleme yöntemi, yinelemeyi ne zaman durduracağınız konusunda bir Öngörü sağlamanıza olanak tanıyan yeni bir aşırı yük alır,

```
IntStream.iterate(1, i -> i < 100, i -> i + 1).forEach(System.out::println);
```

İkinci argüman, IntStream'deki geçerli öğe 100 olana kadar true değerini döndüren bir lambdadır. Bu nedenle, bu basit örnek konsolda 1'den 99'a kadar olan tam sayıları yazdırır.

Stream API improvements

Akışın kendisine yapılan bu eklemelerin yanı sıra, İsteğe Bağlı ve Akış arasındaki entegrasyon iyileştirildi. İsteğe bağlı bir nesneyi, İsteğe bağlı üzerindeki yeni "akım" yöntemiyle (muhtemelen boş) bir Akışa dönüştürmek artık mümkündür:

```
Stream<Integer> s = Optional.of(1).stream();
```

İsteğe bağlı bir Akışı Akışa dönüştürmek, özellikle karmaşık Akış işlem hatları oluştururken kullanışlıdır.

Private interface methods

Java 8, interface'lerde bize varsayılan metotları getirdi. Bir interface artık yalnızca yöntem metotları yerine davranış da içerebilir. Ancak, hemen hemen aynı şeyi yapan kodlu bir arabirimde birkaç varsayılan metotlarınız varsa ne olur? Normalde, paylaşılan işlevselliği içeren özel bir metodu çağırmak için bu metotları yeniden düzenlersiniz. Ancak varsayılan metotlar özel olamaz. Bu yardımcı metot, genel API'nin bir parçası haline geldiğinden, paylaşılan kodla başka bir varsayılan metot oluşturmak bir çözüm değildir. Java 9 ile bu sorunu çözmek için arayüzlere özel yardımcı yöntemler ekleyebilirsiniz:

```
public interface MyInterface {

    void normalInterfaceMethod();

    default void interfaceMethodWithDefault() {  init(); }

    default void anotherDefaultMethod() { init(); }

    // This method is not part of the public API exposed by MyInterface
    private void init() { System.out.println("Initializing"); }
}
```

Private interface methods

API'leri varsayılan yöntemlerle geliştiriyorsanız, özel arabirim yöntemleri, uygulamalarının yapılandırılmasında yardımcı olabilir.