**:ASSIGNMENT 4:**

**2 FAYDALI BİLGİ:**

Bu bölümde, bu proje için ihtiyaç duyacağınız bazı yeni başlangıç seviyesi bilgileri bulabilirsiniz. Her konuyla ilgili daha fazla bilgi edinmek için ek araştırma yapmanız ve diğer kaynaklara (örneğin, ders notları, ders kitabı, İnternet kaynakları vb.) Danışmanız gerekir.

**2.1 SWING'İN KISA BİR BAKIŞ VE BİRİNCİ BİR GUI BAŞVURUNUNUZU:**

Java GUI programlama iki paket içerir: özgün soyut pencere seti (AWT) ve daha yeni Swing araç seti. Swing, bir uygulamanın her bileşeninin görünümünü ve uygulamasını, uygulama kodunda önemli değişiklikler yapmaksızın değiştirilebilmesini sağlayan bir mekanizma geliştirdi. Geçişli bir görünüm ve hisse desteği getirildiğinde Swing bileşenleri, platform bağımsızlığının avantajlarını koruyan yerel bileşenlerin görünümünü taklit ediyor. Başlangıçta ayrı ayrı indirilebilir bir kütüphane olarak dağıtılan Swing, sürüm 1.2'den bu yana Java Standard Edition'ın bir parçası olarak dahil edilmiştir. Swing sınıfları ve bileşenleri javax.swing paketi hiyerarşisinde bulunur. Swing, önceki Soyut Pencere Araç Seti (AWT) 'den daha karmaşık GUI bileşenleri seti sağlamak üzere geliştirildi. Swing, birkaç platformun görünümünü ve öyküsünü taklit eden yerli bir görünüm ve his sağlar; ayrıca, uygulamanın temel alınan platformla ilişkili olmayan bir görünüm ve his elde etmesini sağlayan takılabilir bir görünüm ve hisse destekler. AWT'den daha güçlü ve esnek bileşenlere sahiptir. Düğmeler, onay kutuları ve etiketler gibi tanıdık bileşenlere ek olarak, Swing, sekmeli panel, kaydırma bölmeleri, ağaçlar, tablolar ve listeler gibi çeşitli gelişmiş bileşenler sağlar.

AWT bileşenlerinin aksine, Swing bileşenleri platforma özel kod tarafından uygulanmaz. Bunun yerine, tamamen Java ile yazılmışlardır ve bu nedenle platformdan bağımsızdırlar. "Hafif" terimi böyle bir unsuru tanımlamak için kullanılır.

Salınım bileşenleri, onları orijinal AWT'lerden ayırmak için önek J'ye sahiptir (örneğin Çerçeve yerine JFrame). Swing bileşenlerini ve yöntemlerini projenize eklemek için java.awt. \*, Java.awt.event. \* Ve javax.swing. \* Paketlerini almalısınız. Görüntülenebilir kareler, işletim sisteminin pencere yöneticisi ile arayüz sağlayan JFrame, JWindows, JDialog ve JApplet gibi üst düzey kaplardır. Görüntülenmeyen içerik bölmeleri, JPanel, JOptionsPane, JScrollPane ve JSplitPane gibi ara kaplardır. Bu nedenle, konteynırlar, metin kutuları, onay kutuları, radyo düğmeleri ve benzeri diğer aletleri tutmak ve gruplamak için kullanılan aletler veya GUI denetimleri şeklindedir. Her GUI şeyleri göstermek için kullanılan bir pencere ile başlar. Salınımda üç tür pencere vardır: Applet, Dialog ve Frame. Bu arayüz ile pencere yöneticisi. Salınımda, bir çerçeve nesnesine bir JFrame denir. Bir JFrame en çok konteyner kabul edilir. Bunlara görüntülenebilir çerçeveler de denir. Görüntülenmeyen içerik bölmeleri, JPanel, JScrollPane, JLayeredPane, JSplitPane ve JTabbedPane gibi çoklu kontroller uygulandığında düzen yapısını düzenleyen ara kaplardır. Basitçe belirttiğimiz gibi, içerik bölmesi, metin alanlarının diğer widget'lar olduğu yerdir, bu yüzden GUI kontrollerini eklemek ve görüntülemek için eklediğimiz içerik bölmesinin olduğunu belirtmeliyiz. İçerik bölmesi daha sonra, bu ağaç benzeri hiyerarşinin üst düzey bir kap (bizim durumumuzda JFrame) içerdiği bir kapsama hiyerarşisinin en üstündedir. Ağacın altında çalışırken, bileşenleri tutmak için JPanel gibi diğer üst seviye konteynırları buluruz. Java swing API hiyerarşisi aşağıda verilmiştir.

**2.2 "WINDOWS BUILDER" a GİRİŞ:**

Bir GUI oluşturucu, bir kullanıcının paletten tasarım yüzeyine sürükleyip bırakarak bir grafik kullanıcı arabirimi oluşturmasına olanak tanıyan görsel bir programlama araçtır. GUI oluşturucular genellikle GUI oluşturma sürecini otomatikleştiren sihirbazları ve şablonları içerir. GUI oluşturma araçları, geliştiricilerin Java GUI'leri oluşturarak daha az zaman ve para harcamanıza izin verir, çünkü GUI'lerin çalışması için gerekli olan düşük düzey mantığı kodlamaktan ziyade uygulamaya özel işlevler oluşturmaya odaklanabilirler. Ben Galbraith'e göre, Başarılı GUI Oluşturma blogunda "Benim deneyimime göre, GUI geliştiricilerin itirazda bulunmalarının bir numaralı nedeni, çoğunlukla ortalama seviyedeki Swing geliştiricileri arasında, el kodlamasına kıyasla üretkenlikte büyük bir gelişme sağladığı yönündedir . Kendimi ortalamanın üstünde değerlendiriyorum ve üretkenliğindeki iyileştirmeleri kendi çalışmalarımızda 2x-10x (belki de daha yüksek) görüyorum. "2003 yılında, Anında Uygulamalar tarafından oluşturulmuş olan WindowBuilder, Java dünyasında en iyi GUI oluşturucu olarak kabul ediliyor ( 2009'da En İyi Ticari Eclipse Aracı). SWT, XWT, Swing ve Google'ın GWT'si için bir GUI tasarım aracına örnek teşkil eden bir örnek olmanın ötesinde WindowBuilder, herhangi bir kullanıcı arayüzüne dayalı yeni GUI tasarım araçlarını yaratmak için son derece genişletilebilir ve özelleştirilebilir bir çerçeve (50+ mevcut uzantı noktaları ile birlikte) 'dir dil. Google, WindowBuilder'ı, 2010'da Anonimleştirmeler'i satın almasından kısa süre sonra Eclipse topluluğunun herhangi bir üyesi tarafından ücretsiz olarak temin edilebilir hale getirdi ve şimdi Framework (WindowBuilder Motoru) ve ilişkili ürünleri (SWT Designer ve Swing Designer) Eclipse Foundation'a katkıda bulunmak istiyor . Google, kendi şirket içi GWT Designer aracının temelini oluşturmaya devam edeceği için projeye yoğun bir yatırım yapmaya devam edecektir.

WindowBuilder, SWT Designer ve Swing Designer'dan oluşur ve kod yazmak için çok fazla zaman harcamadan Java GUI uygulamaları oluşturmayı çok kolaylaştırır. Karmaşık pencerelere basit formlar oluşturmak için WYSIWYG görsel tasarımcı ve düzen araçlarını kullanın; Java kodu sizin için oluşturulacak. Sürükle ve bırak kullanarak denetimleri kolayca ekleyin, denetleyicilere olay işleyicileri ekleyin, bir mülk editörü kullanarak denetimlerin çeşitli özelliklerini değiştirin, uygulamanızı uluslararasılaştırın ve çok daha fazlasını yapın.

WindowBuilder, Eclipse'e ve çeşitli Eclipse tabanlı IDE'lere (RAD, RSA, MyEclipse, JBuilder vb.) Eklenti olarak geliştirilmiştir. Eklenti, kaynak kodunda gezinmek için soyut bir sözdizimi ağacı (AST) oluşturur ve görsel sunuyu görüntülemek ve yönetmek için GEF kullanır. Üretilen kod, derlemek ve çalıştırmak için herhangi bir ek özel kütüphane gerektirmez: oluşturulan tüm kod, indowBuilder Pro yüklenmeden kullanılabilir. WindowBuilder Pro, hemen hemen her biçimi okuyabilir ve yazabilir ve en elle yazılmış Java GUI kodunu ters mühendislik yapabilir. Ayrıca, serbest form kodu düzenlemeyi (yalnızca özel alanlarda değil de değişiklikler yapmak için) ve çoğu kullanıcı yeniden faktoringini (yöntemleri sorunsuz hareket ettirebilir, yeniden adlandırabilir ve alt bölümlere yerleştirebilirsiniz) destekler. WindowBuilder araç seti, pazarda arama yaparak Eclipse'in Yardım menüsünden kolaylıkla indirebilir.

**3 UYGULAMAYA BİR BAKIŞ:**

Bu programlama ödevinde, çok iyi bilinen bir sosyal medya platformu olan Facebook'un basitleştirilmiş bir versiyonunu geliştireceksiniz. Buna Mini-Facebook diyoruz. 3. ve 4. Sınıf Programlamanın ardışık olduğunu ve Ödev 3'te daha önce oluşturduğunuz derslerinizi pek az değişiklikle birlikte kullanabileceğinizi unutmayın.

**3.1 PROBLEM TANIMI:**

Bu görevde, Swing bileşenlerini kullanarak bir GUI tabanlı ön uç oluşturmanız gerekir. Bu GUI'yi kullanarak bir kullanıcının Mini-Facebook sistemiyle etkileşim kurması gerekir. Önce, sistemi başlatmak için bunları kullanacağınız iki metin dosyası ("users.txt" ve "commands.txt") sağlanacaktır. Tam olarak anlaşılacağı üzere, tüm kullanıcılar, arkadaşlık / engellenen kullanıcılar listeleri bu iki girdi dosyasının kullanılmasıyla oluşturulacaktır. Dahası, tüm ilk mesajlar oluşturulacak ve ilgili kullanıcılara atanacaktır. Bu dosyaların adını statik ana işlevdeki args parametresinden alabilirsiniz. Adları ve sıraları sırasıyla users.txt ve commands.txt olacaktır. Bu görevde, sistemin kullanıcı sınıfı ve iş mantığı biraz değiştirildi. Aşağıdaki listede, bu değişiklikler listelenmiştir:

1. Bu ödevde tüm kullanıcıların kendi kullanıcı adlarını ve şifrelerini girerek giriş yapabilmeleri sağlanırken, 3. ödev tek kullanıcılı giriş üzerinde yoğunlaşmaktadır. Başka bir kullanıcının oturum açmasına izin vermek için daha önce oturum açmış olan kullanıcının ilk önce çıkış yapması gerektiğini unutmayın. Bunun nedeni, sisteminizin oturum açmış yalnızca bir kullanıcıyla ilgilenebilmesidir.

2. Kullanıcı sınıfı şu an {"Bir ilişki içinde", "Boşanmış", "Karmaşık", "Tek"} değerlerinden birini depolamak için tasarlanmış yeni bir özellik "relationship\_status" içeriyor.

3. Kullanıcı adı alanının benzersiz bir değer özelliği olması nedeniyle, Kullanıcı sınıfının userid mülkü artık gerekli değildir. Bu nedenle, username mülkünü koleksiyonlarda bir anahtar olarak kullanabilirsiniz.

4. Kullanıcı sınıfının bazı işlevleri (örneğin şifre değiştirme) kaldırılmıştır. Kullanıcı sınıfının yeni kuralları için lütfen Bölüm 3.1.1'i okuyun.

5. 3. ödevde, "Kullanıcıların gönderilerini görüntüleme" olarak belirtilen UserCollection sınıfının bir gereksinimi vardı. Bununla birlikte, bu ödev dahilinde, Yayınların sorumluluğu Kullanıcı sınıfının kendisine taşındı. Başka bir deyişle, "posta ekleme" veya "yazıları göster" gibi bazı işlevler Kullanıcı sınıfında ele alınacaktır.

Kullanıcı, Kullanıcı sınıfında bir koleksiyonda saklanmalıdır. Kullanıcı koleksiyon sınıfının hâlâ mevcut olduğunu ve her kullanıcının koleksiyonuna kaydedilmesinden sorumlu olduğunu unutmayın. Aşağıdaki alt bölümlerde işletme sınıflarının ayrıntıları sunulmaktadır.

**3.1.1 Kullanıcılar:**

Kullanıcılar Mini Facebook sistemimizde bir hesabı olan kişilerdir. Kullanıcıların farklı roller olmaz (ör. Yönetici veya sistem yöneticisi vs. diğer kullanıcılar); Yani hepsi akran olurlar. Kullanıcının nitelikleri ve davranışları aşağıda açıklanmıştır:

< Sistemdeki her Kullanıcı benzersiz bir kullanıcı adı, bir ad, bir şifre, doğum tarihi, kullanıcının mezun olduğu okul hakkındaki bilgileri, son giriş tarihi, arkadaş topluluğu, engellenen kullanıcı topluluğu ve Bir yazı koleksiyonu.

< Kullanıcıların doğum tarihleri ve son giriş tarihleri Tarih veya Takvim değişkeni olarak saklanmalıdır. Bununla birlikte, hem giriş hem de GUI oluşturma biçimi "gg / aa / yyyy" olmalıdır (ör. 03/11/1997)

< Tüm koleksiyonlar Java Koleksiyonları olarak uygulanmalıdır.

Kullanıcılar;

O Oturum açın,

Oturumu kapatın,

O Kullanıcı profil bilgilerini (isim, doğum tarihi, mezun olan okul ve ilişki durumu) güncelleyin,

O Arkadaş listelerine arkadaş ekleme,

O arkadaşlarını arkadaş listelerinden çıkartın,

O Mesaj ekleyin,

O Kullanıcıları engelleyin,

O Kullanıcıların engelini kaldırın,

O Arkadaşlarını listeleyin,

O Kullanılabilir tüm kullanıcıları listeleyin (burada, kullanılabilir terim, engellemediğiniz kullanıcılara ve onları engellediğiniz kullanıcılara yönlendirmek için kullanılır),

O Engellenen kullanıcıları listele.

Bir kullanıcı oturum açmadığında, yalnızca oturum açmasına izin verilmelidir. Kullanıcı oturum açtıktan sonra, diğer işlemlere izin verilmelidir. Hassas kişisel kullanıcı bilgilerini gizlemeniz ve setter ve getter yöntemlerini gerektiği gibi uygulamanız gerekmektedir.

**3.1.2 Mesajlar:**

Mesajlar, kullanıcıların paylaştığı metin veya multimedya içeriğini temsil eder. Post sınıflarınızı hiyerarşik bir yapıda tasarlamanız bekleniyor. Yayınlarla ilgili gerekli bilgiler aşağıdaki gibidir:

< Birincisi, Post sınıfının uygulayacağı davranışları (yöntemleri) belirtecek bir arabirim uygulamanız gerekir.

< İkinci olarak, post arayüzünü uygulayacak ve diğer tüm posta sınıflarına üst sınıf olacak soyut bir Post sınıfı uygulamanız gerekiyor.

< Bu soyut Post sınıfı, post özniteliklerini tanımlamalı, arayüzde tanımlanan yöntemleri uygulamalı ve etiketli kullanıcıları gösterecek ve post konumunu gösterecek iki ek soyut yöntemi tanımlamalıdır. Bu yöntemler Post sınıfının derhal alt sınıflarında uygulanmalıdır.

< Mesajlar, TextPosts, ImagePosts veya VideoPosts olarak kategorize edilir. Ancak dikkatli olun: Tüm yayınların metin bölümleri var (tüm yayınlar aslında metin gönderileri) ve isteğe bağlı olarak resim veya video içerebilir (ancak ikisini birden değil!).

< Tüm yayınlar, rasgele bir UUID (değiştirilemez evrensel benzersiz tanımlayıcı), bir metin, gönderildiği tarih, bir yer ve etiketlenmiş arkadaşlar topluluğu olarak atanması gereken benzersiz bir postID'ye sahiptir.

< Konum şu özniteliklerle ayrı bir sınıf olarak uygulanmalıdır: enlem ve boylam (çift değerler halinde saklanır).

< TextPost sınıfı soyut Post sınıfından soyut yöntemler uygulamalıdır. Etiketli arkadaşlarınızı göstermek için isimlerinizi basmanız yeterlidir (varsa). Posta konumunu göstermek için, enlem ve boylamın her ikisi de yazdırılmalıdır.

< ImagePosts'da resim dosya adı ve görüntü çözünürlüğü bulunur (genişlik ve yükseklik piksel cinsinden).

< VideoPost'ların video dosya adı, video süresi ve maksimum video uzunluğunu dakika cinsinden belirten sabit bir nitelik vardır (izin verilen maksimum video süresi 10 dakikadır).

< Setter ve getter yöntemlerini gerektiği gibi uygulamak zorunda olduğunuzu unutmayın.

**3.1.3 Kullanıcı Koleksiyonu:**

UserCollection ana sınıfının aşağıdaki eylemlere izin vermesi gerekir:

< Tüm kullanıcıları takip etmek,

< Sisteme yeni kullanıcılar ekleme (yeni kullanıcı bilgileri sağlayarak),

< Mevcut kullanıcıları sistemden kaldırmak (kullanıcı adını vererek),

< Kullanıcı oturum açma (kullanıcı adı ve parola ile),

< Üçüncü görevde "Kullanıcıların yazılarını göster" şeklinde bir gereklilik vardı. Bununla birlikte, Postaların sorumluluğu Kullanıcı sınıfının kendisine taşındı.