

**Dersin Adı:** Nesneye Dayalı Programlama

**Proje:** Student Assistant

**Rapor Teslim Tarihi:** 04.01.2017 Çarşamba

Harun UYAR – 05140000633

Naim YÜREK – 05140000649

Sedat GİRGİN – 05130000265

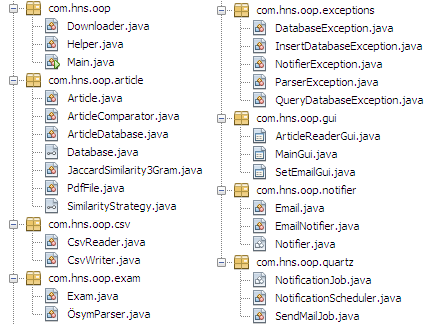
PROJE TANIMI

Proje kapsamında Lisansüstü öğrencilerin üzerinde çalışacağı makalelerin yönetimi ve yaklaşan ALES, YDS gibi sınavların bilgilendirilmesi gibi gereksinimlerini karşılayan bir uygulama geliştirilecek.

* *ACM.csv* dosyasından 10 tane makale seçilecek. Bu küçük CSV dosyasında yer alan “id” parametresiyle makalelerin PDF dokümanları JAVA kodu ile otomatik olarak indirilecek.
* Makale içerikleri MongoDB veri tabanına aktarılacak. Bunu yaparken DAO tasarım deseni kullanılacak.
* Makalelerden birbirine benzeyenler kullanıcıya önerilecek. Benzerlik hesabı bir ya da birden fazla şekilde yapılabilir. Bunlardan bir tanesi Jaccard benzerlik hesabı olacak. Benzerlik hesapları arasında seçim yapmak için strateji tasarım deseni kullanılacak.
* Makale veri tabanına aktarılırken, bu makaleye ilişkin bilgilerden id, title, author, venue ve year bilgileri CSV dosyasından elde edilecek. Makalenin içeriği ise indirilen PDF dosyasında elde edilecek. Ayrıca daha kolay arama yapılabilmesi için en çok geçen kelimeler de veri tabanına aktarılacak.
* Her makale için en sık geçen kelimeler listesi hazırlanacak. Ayrıca 3Gram listeleri de oluşturulacak.
* Kullanıcı, arayüz yardımıyla makalenin adını, yılını ya da anahtar kelimesini girerek arama yapabilecek.
* Kullanıcı, sınav takviminden seçtiği sınavların başvuru tarihleri yaklaşınca bildirim alacak. Sınav takvimi HTML etiketlerinden arındırılarak internet sitesinden çekilecek.
* Bildirimler E-Posta yoluyla gönderilebileceği gibi kullanıcı ara yüzünden de uyarı yoluyla gönderilebilir.
* Üstteki iki tasarım desenine ek en az bir tasarım deseni daha kullanılacak.
* JUnit kullanılarak birkaç test yapılacak.
* Proje Github’a yüklenecek.

PROJE TASARIMI

Projede kullanılan Java Sınıfları şunlardır:



* Main: Program başlatıldığında çalışan sınıftır. MainGui’yi çağırır.
* MainGui: Ana kullanıcı ara yüzüdür. Kullanıcı bütün işlemlerini burada gerçekleştirir. Bu işlemleri gerçekleştirmesi için Helper sınıfının defaultHelper nesnesini kullanır.
* Helper: Bütün işlemleri bu sınıf gerçekleştirir. Çok sayıda metodu ve değişkeni barındırır. Singleton tasarım deseni kullanılmıştır.
* Downloader: İndirme işlemlerini gerçekleştirir. Bu sınıfta da Singleton tasarım deseni kullanılmıştır.
* ArticleReaderGui: MainGui’de seçilen makaleyi okumak için oluşturulur. Bu makaleye benzer makaleler de veri tabanından bulunarak listelenir.
* SetEmailGui: Kullanıcıdan bir E-Posta adresi girmesini ister ve bildirim sıklığını sorar.
* Article: Bir makaleyi temsil eden sınıf.
* Article Comparator: İki makaleyi karşılaştırıp benzerlik oranını bulmak için üretilen sınıf. Strateji tasarım deseni kullanılmıştır.
* SimilarityStrategy: Benzerlik stratejisi ara yüzü.
* JaccardSimilarity3Gram: 3Gram oluşturma yoluyla iki makalenin benzerliğini hesaplama stratejisi. SimilarityStrategy’i implement eder.
* Database: Veri tabanı ara yüzü. DAO tasarım deseninin kullanımı sonucunda üretildi.
* ArticleDatabase: Makale veri tabanı sınıfı. Database’i implement eder ve veri tabanına veri giriş-çıkışını sağlar. MongoDB’yi kullanır.
* PDFFile: Bir PDF dosyasını temsil eden sınıf. PDF dosyanının okunup içeriğinin alınması amacıyla oluşturuldu. PDFBox hazır kütüphanesini kullanır.
* CsvReader: Csv dosyasını okumayı sağlayan sınıf.
* CsvWriter: Csv dosyası oluşturmayı sağlayan sınıf. Kişinin seçtiği sınavları kaydetmek için oluşturuldu.
* Exam: Bir sınavı temsil eden sınıf.
* ÖsymParser: Ösym’nin sitesinden sınav tarihlerini çeken sınıf. Singleton tasarım deseni kullanılmıştır. Jsoup hazır kütüphanesini kullanır.
* Email: Bir E-Postayı temsil eden sınıf.
* Notifier: Bildirim göndermek için kullanılan soyut sınıf. DAO tasarım deseni kullanılmıştır.
* EmailNotifier: E-Posta olarak ile bildirim göndermek için kullanılan sınıf. Notifier’ı kalıtımlar. JavaxMail hazır kütüphanesini kullanır.
* NotificationScheduler: Quartz hazır kütüphanesini kullanarak belli aralıklarla bildirim gönderiminin kontrol edilmesini sağlar.
* NotificationJob: Job’u implement eden iş bildirim işi soyut sınıfı. NotificationScheduler tarafından kullanılır. DAO tasarım deseni kullanılmıştır.
* SendMailJob: NotificationJob’u kalıtımlayan bildirim işi. Her bir SendMailJob, bir sınav için gönderilecek bildirimi temsil eder.

JUNIT TESTLERİ

Projede kullanılan JUnit testleri şunlardır:

* DownloaderTest: Bir dosyayı program dizinine indirebiliyor mu diye test eder.
* CsvWriterTest: ÖsymParser sınıfını kullanarak sınav bilgilerini Ösym sitesinden çekip CsvWriter kullanarak CSV dosyasına yazar. Ardından yazılan verileri CsvReader ile geri okur ve konsola yazdırır.
* ArticleTest: Veri tabanında kayıtlı ilk iki makaleyi alıp benzerlik oranlarını konsola yazdırır.
* DatabaseTest: Bir ArticleDatabase nesnesi oluşturarak veri tabanının içine veri ekler. Ardından bu veri tabanındaki kayıtları listeler.
* ÖsymParserTest: ÖsymParser ile sınavları çekip ekrana yazdırır.
* NotifierTest: Mail gönderme işlemini test eder.

KULLANICI KILAVUZU