Programlama Standartları Dökümanı

Flash B2B

Versiyon 0.3

Hüseyin Ergün

30 Ekim 2004

A. İÇİNDEKİLER

A. İÇİNDEKİLER II
B. GEÇMİŞ DEĞİŞİKLİKLER III
1. GİRİŞ 4
2. İSİMLENDİRME 4
2.1 PAKET İSİMLERİ 4
2.2 CLASS İSİMLERİ 5
2.3 SINIF METODU (FONKSİYON) İSİMLERİ 5
2.4 CONSTRUCTORLAR 6
2.5 DEĞİŞKENLER VE ARABİRİM OBJELERİ 6
2.6 PARAMETRE İSİMLERİ 7
2.7 DÖNGÜ DEĞİŞKENLERİ 8
2.8 EXCEPTİON OBJELERİ 8
2.9 SABİT DEĞERLER 8
2.10 VERİTABANI VE TABLO İSİMLERİ 8
2.10.1 Tablo İsimleri 9
2.10.2 Kolon (Entity) İsimleri 9
3. KOD ORGANİZASYONU 10
3.1 CLASS INHERİT EDERKEN 10
3.2 DEĞİŞKENLERİ TANIMLARKEN 11
3.3 BAŞLIK 11
4. DÖKÜMAN VE AÇIKLAMA 12

B. GEÇMİŞ DEĞİŞİKLİKLER

İsim	Tarih	Değişiklik Nedeni	Versiyon
Hüseyin Ergün	1 Ekim 2004	Yazımdan doğan yanlış anlaşılmaların düzeltilmesi. Veritabanı standartlarının eklenmesi.	0.2
Hüseyin Ergün	23 Eyl. 04	Değişken ve arabirim objelerine yeni isimler eklendi. Standartları tamamlandı.	0.3

1. Giriş

Bu dökümanda, B2B Entegrasyon Sunucusu projesinin programlama standartları belirtilmiştir. Projelerde harcanan vaktin çoğu programı yazmak için değil düzenlemek ve devamlılığını sağlamak için harcanır. Bugün yazdığınız bir kod yıllar sonra başkaları tarafından kullanılıp devam ettiriliyor olabilir. Yazdığınız kodun başkaları tarafından kolay anlaşılır olması projenin devamlılığı açısından önemlidir. Kodunuzun "temiz" ve anlaşılır olması için özen göstermelisiniz.

2. İsimlendirme

Java kodu yazarken uyulması gereken standartlar aşağıda belirtilmiştir.

2.1 Paket İsimleri

a. Bütün paket isimleri küçük harfle yazılır.

b. Programa eklenen her bir modül için ayrı paket ismi verilecektir.

c. Paket isimleri **com.flashb2b** ile başlar.

d. Paketlerin standart dili İngilizce'dir. Proje kodlarının tüm dünyada kullanılması ve yurtdışında anlaşılır olması için İngilizce seçilmiştir.

e. Her bir paketin içinde 3 ayrı paket bulunacaktır. Sırasıyla bunlar **ui**, **bl**, **dal**. **ui** (User Interface) kullanıcı arabirimi anlamına gelir. Tüm arabirim parçacıkları bu katmanda yer alacaktır. **bl** (Business Logic) İş Mantığı katmanıdır. Programdaki hesaplama, algoritma ve kullanıcı tanımlı arabirim özellikleri bu katmanda oluşturulur. Veritabanı ile görsel arabirim arasında köprü oluşturur. **dal** (Data Access Layer) veritabanı erişim katmanıdır. Veritabanı için gerekli sorgular bu

katmandaki fonksiyonların içinde oluşturulur. Bağlantı bilgileri ise **com.flashb2b.engine.dal** paketinden çağrılır.

2.2 Class İsimleri

a. Sınıf isimleri büyük harfle başlar.

b. Sınıf isimlerinin içindeki bütün kelimeler büyük harfle başlar.

c. Sınıf isimlerinde resmi dil İngilizce'dir. Proje kodlarının tüm dünyada kullanılması ve yurtdışında anlaşılması için İngilizce seçilmiştir.

d. Sınıf isimlerinin uzun olmamasına (15 harften az) özen gösterilmelidir.

e. Class isimlerinin başında modül paket isminin ilk harfleri büyük harfle başlayıp küçük harfle devam ederek yazılır. Ardından büyük harflerle alt paketin isimleri yazılır. Engine paketi için Örnek bir sınıf ismi EngDALTransaction 'dır.

2.3 Sınıf Metodu (Fonksiyon) İsimleri

a. Sınıf metodları tam İngilizce kullanılarak tanımlanmalıdır.

b. Metod isimleri küçük harfle başlar, ardından her kelime büyük harfle devam eder.

c. Metod ismindeki ilk kelimenin işi temsil etmesine ve yüklem olmasına özen gösterilmeli.

Örneğin:

```
openAccount()
printMailingLabel()
save()
delete()
```

Bu sayede metodların yaptığı iş sadece ismine bakarak anlaşılabilir. Geliştirici tarafından yazması biraz daha uzun sürse de, anlaşılırlığı çok arttıracaktır.

d. Metodların sonuç döndürenleri Getters diye adlandırılır. Getters'larda geri dönen sonuç boolean (0/1) değilse başına **get** eklenir, öteki türlü "**is**" ile başlar. Örneğin

getFirstName()
getAccountNumber()
getLostEh()
isPersistent()
isAtEnd()

e. Metodların değerleri değiştirmek içinse Setters diye adlandırılır. Değiştirilen verinin tipinden bağımsız olarak bütün değiştirme metodlarının başına gelmelidir. Örneğin

setFirstName(String aName)
setAccountNumber(int anAccountNumber)
setReasonableGoals(Vector newGoals)
setPersistent(boolean isPersistent)
setAtEnd(boolean isAtEnd)

2.4 Constructorlar

Constructorlar obje ilk yaratıldığında yapılacak işleri belirleyen fonksiyonlardır. Bunların ismi Class ismiyle aynı olmalıdır. Sun firması tarafından belirlenen Java standardı olduğundan başka türlü çalışmaz.

2.5 Değişkenler ve arabirim objeleri

a. Değişken isimlerinde açıklayıcı İngilizce kelimeler kullanılmalıdır.
b. Değişken isimleri, veri yapısını belirtecek şekilde başlamalıdır. Kullanılacak değerler aşağıda belirtilmiştir.
c. Başlıklar küçük harfle yazılır. Ardından gelen açıklayıcı İngilizce kelimelerin her biri büyük harfle başlar. Örneğin
btnSave

intInvAmount

Veri Yapısı	**Başlık**
Sayı	int
Kelime	str
Label	lbl
Boolean	bool
Array	ary
Textbox	Txt
Decimal Textbox	txtDec
Numeric Textbox	txtNum
Düğme	Btn
Byte	Byt
Double	Dbl
Float	Flt
Long Int	Lng
Tab Folder	Tabfld
Tab Item	Tab
Table	Table
Big Decimal	Bigd
Decimal	Dec
Tree	Tree
Tree Item	Tim
Composite	Comp
Combo Box	Combo
List	List
Menu Item	Mit
Menu	Menu
Date	Date
Sash Form	Sash
Toolbar	Toolbar
Tool Item	Tool
Cool Bar	Coolbar
Cool Item	Cool

2.6 Parametre İsimleri

Parametreleri kullanırken, Değişken ve Arabirim objeleri ile aynı metodun kullanılması gerekir. Diğerlerinden farklı olarak parametre olduklarını belirtmek ve yerel değişkenlerden ayırmak için başlarına **a** veya **an** gelmelidir.

2.7 Döngü Değişkenleri

Genel olarak i, j, k gibi değişkenler yerine loopCounter veya counter1, counter2 gibi değişkenler kullanmak daha akıllıcadır. En büyük avantajı arama yaparken yanlış sonuçlarla boğuşmamıza gerek kalmaz. Fakat bu kısım kullanıcının tercihine bırakılmıştır.

2.8 Exception Objeleri

Java programlama dilinde exceptionlar sıkça geçtiği ve döngünün dışında kullanılmadığı için **e** harfini vermek yeterlidir.

2.9 Sabit Değerler

Java programlama dilinde static final olarak tanımlanan ve tüm programda değeri değişmeyen sabit değerlerdir. Açıklayıcı İngilizce kelimeler kullanarak büyük harfle yazılmalıdır. Kelimelerin arasına alt çizgi (_) konulmalıdır.

Örneğin

MINIMUM_BALANCE
MAX_VALUE
DEFAULT_START_DATE

Bu tarz kullanımın en büyük avantajı değişkenleri sabit değerlerden ayırabiliriz. Genel olarak sabit değerler tanımlamak yerine Getter fonksiyonları kullanarak bu değerleri ayrı bir class'tan almak işlerimizi daha kolaylaştırır.

2.10 Veritabanı ve Tablo İsimleri

Proje veritabanında tablo ve kolon (entity) isimlendirmesinde uyulması gereken standartlar belirtilmiştir.

2.10.1 Tablo İsimleri

a. Tablo isimlendirmesinde dil olarak İngilizce kullanılmalıdır.

b. Tüm tablo isimlerinin başına önek (prefix) olarak **flashb2b_** gelmelidir.

c. Tablo isimlendirmesinde küçük harfler kullanılmalıdır.

d. Kelimeler kısaltma yapılmadan ve birbirlerinden alttan çizgiyle ayrılarak yazılmalıdır. Örneğin:

flashb2b_smtp_transactions

e. Tablolar bir modüle aitse, bağlı bulundukları modülün ismiyle başlamalıdır.

flashb2b_smtp_transactions
flashb2b_xml_soap_entries

2.10.1.1 Görüntüleme (View) Tabloları

View tabloları veri görüntülemek için kullanılır. Diğer tabloların bilgilerini alarak sanal bir tablo oluşturur. Veri giriş çıkışı olmaz. Bu tabloları diğerlerinden ayırmak için isimlendirmede prefix'in ardından **_view** eklenmelidir. Örneğin flashb2b_view_xml_users

2.10.2 Kolon (Entity) İsimleri

a. Birincil anahtarın (Primary Key) ismi, prefix olmadan tablonun ismi ardından **_id** gelerek verilmelidir. Örneğin

flashb2b_transactions_id

b. Yabancı anahtarlar (Foreign Key) gösterdiği birincil anahtarın ismini aynen taşır.

c. Diğer kolon isimleri küçük harfle, tablonun yaptığı işin ismini içermelidir. Örneğin xml_transaction tablosu için
transaction_type, transaction_definition, transaction_amount

d. Her tabloda örneği görülen son değişiklik, kimin tarafından değiştirildiği gibi bilgiler için tablo isminden bağımsız, ortak standart bilgiler kullanılır. Aşağıda listesi verilmiştir.

Açıklama	Kolon ismi
Kimin tarafından oluşturulduğu	created_by
Ne zaman oluşturulduğu	creation_date
Kimin tarafından güncellendiği	updated_by
En son ne zaman güncellendiği	last_modified

3. Kod Organizasyonu

Kodlandırma esnasında yapılan standartlar aşağıda maddeler halinde belirtilmiştir.

3.1 Class Inherit Ederken

Classları import ederken paket içindelerse isimlerini tek tek yazmak yerine o paketin tamamı çağrılmalıdır. Örneğin

İmport.java.awt.Color;
İmport.java.awt.Button;
İmport.java.awt.Container; yerine

İmport.java.awt.*; yazılmalıdır. SDK compile ederken sadece gerekli class'ları ekleyecektir. Hem hafızada fazla yer kaplamaz hem de yeni Class eklerken veya çıkarırken kolaylık sağlar. Tek tek classlarla uğraşmazsınız.

3.2 Değişkenleri Tanımlarken

Aynı işi yapan değişkenler bir arada oluşturulmalı ve ardından satır bırakılmalıdır. Bu sayede kod daha okunabilir hale gelecektir. Örneğin

anObject.message1();
anObject.message2();
aCounter = 1;
anOblect.message3();
	anObject.message1();
	anObject.message2();
	anObject.message3();
	
	aCounter = 1;

Karşılaştırmalarda değişken isimleri solda yer almalıdır. Böylelikle kodlar daha anlaşılır hale gelecek ve olası hataların önüne geçilecektir. Örneğin

if(something == 1) { ... }
 	
if (x = 0) { ... }
	if(1 == something) { ... }
	
	if (0 = x) { ... }

3.3 Başlık

Her bir Class'ın başına Proje ekibi tarafından belirlenmiş aşağıdaki başlık gelmelidir. Ardından Class'ın importları, daha sonra da imin tarafından ve ne zaman değiştirildiği yazacaktır. Son ki değişikliklerin otomatik güncellenmesi için gerekli ayarlar geliştiriciler dökümanında belirtilecektir.

```
/*****************************************************************************/
/* flashB2B: Application Business Integration Engine                         */
/* ========================================                                  */
/* Copyright (c) 2004 by flashB2B Software Development Group                 */
/*                                                                           */
/* This program is free software. You can redistribute it and/or modify      */
/* it under the terms of the GNU General Public License as published by      */
/* the Free Software Foundation; either version 2 of the License, or         */
/* (at your option) any later version.                                       */
/*                                                                           */
/* This program is distributed in the hope that it will be useful,           */
/* but WITHOUT ANY WARRANTY; without even the implied warranty of            */
/* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE                       */
/* See the GNU General Public License for more details.                      */
/*****************************************************************************/
```

4. Döküman ve Açıklama

Her Class'ın üst kısmında değişiklik bilgisinin altında Class için gerekli açıklama gelmelidir. Bu açıklama /** ile başlar ve */ ile biter. Javadoc kullanarak otomatik döküman hazırlandığı için bu standartlara uymak zorunludur.

Class hakkında İngilizce kısaca, ne için oluşturulduğu, fonksiyonlarının özellikleri belirtilmelidir.

Her bir fonksiyonun başında aynı şekilde /** ile başlayan ve o fonksiyonun ne işe yaradığı, aldığı parametreler ve döndürdüğü değerleri içeren açıklama yazılması gerekli. Açık ve sade bir dille İngilizce yazılmalıdır.