

JavaTürk

Java Kod İsimlendirme ve Şekil Standardı

Akin Kaldiroglu

akin@javaturk.org

İçindekiler

Giriş	3
1. En temel şekil kuralları.....	3
2. Genel isimlendirme kuralları	4
3. Paket isimlendirme kuralları	5
4. Tip isimlendirme kuralları.....	5
5. Değişken isimlendirme kuralları.....	6
6. Metot isimlendirme kuralları	7
Kaynaklar.....	8

Değişiklik Geçmişi

Tarih	Açıklama	Yapan
24 Mart 2014	İlk oluşturma.	A. Kaldiroglu

JavaTürk Java Kod İsimlendirme ve Şekil Standardı

Giriş

Bu dokümanda, Java kodunda kullanılacak isimlendirme ve şekil (format) standartları sıralanmıştır.

Bu doküman serbestçe kullanılabilir ve en güncel haline www.javaturk.org adresinden erişilebilir.

1. En temel şekil kuralları

1. Daima paragraf kullan. Kod yazarken parmakların sıklıkla format tuşlarında olsun.
2. Her satırda sadece bir cümle (statement) yaz.
3. Uzun satırları bir kaç satıra yay ki yatay scrolla ihtiyaç kalmasın.
4. Mekanı rahat kullan: Mümkün olan her yerde boşluk, “ “, ve boş satır kullanarak okunurluğu artır.
5. Blokları mümkünse “{ }” ile değilse boş bir satır ile ayır.
6. Zeka yarışına girme, “=” dahil en az 3 operatörlü ifadeleri anlamak operatörlerin öncelik ve ilişkilendirme bilgisine bağlı olmasın, parantez kullan.

Böyle yapma:

```
float rs = a + ++b * c/a * b;
```

```
a += b += c;
```

Böyle yap:

```
rs = a + (++b) * ((c / a) * b);  
b += c;  
a += b;
```

7. Zincirleme üye erişimi ile birleşik ifade yazma, her ifadede bir üyeye eriş.

Böyle yapma:

```
customer.getCompany().getAddress().getStreet();
```

Böyle yap:

```
Company company = customer.getCompany();  
Address address = company.getAddress();  
Street street = address.getStreet();
```

2. Genel isimlendirme kuralları

8. Daima İngilizce isimler kullan ve kesinlikle yanlış yazma, emin değilsen sözlüğe bak.
9. Daima anlamlı isimler kullan; uzun olsun, anlamsız olmasın.

```
addedValueTaxRate, getDefaultAccountInterestRate()
```

10. Okumayı zorlaştıran (özellikle sesli harfleri atarak elde edilen) kısaltmalardan kaçın.
11. Tanıdık isimler kullan. Aynı şeyler için her yerde aynı ismi kullan.
12. Kısaltmalarda büyük harflerle yazma.

Böyle yapma:

```
hTTPSession, TCPIPConnection, getXMLNode(),  
getHTTPMethod()
```

Böyle yap:

```
httpSession, TcpIpConnection, getXmlNode(),  
getHttpMethod()
```

13. Tutarlı ol. Aynı ismi sadece küçük-büyük harf ayrımıyla ya da hem kısa hem uzun şekliyle defalarca kullanma.

```
sqlQuery, sqlQry ya da session, ssn
```

14. İsimlendirmede daima “Camel Case” yaklaşımını kullan, alt çizgiden “_”, uzak dur. Camel Case’in iki türü vardır, her ismin baş harfinin büyük olduğu Upper Came Case (UCC) ile sadece ilk kelimenin ilk harfinin küçük, sonrasının UCC olarak devam ettiği Lower Camel Case (LCC).

```
StudentInformation is UCC  
getAllStudents() is LCC  
studentAddress is LCC
```

3. Paket isimlendirme kuralları

15. Paket isimlerine internet alan adınızı tersinden yazarak başla.

```
tr.com.selsoft, org.javaturk
```

16. Paketlerini küçük harfle yaz ve tek ve tekil isimler ver.

```
tr.com.selsoft.atm.domain  
org.javaturk.designpattern.customer
```

4. Tip isimlendirme kuralları

17. Sınıf, arayüz, enumeration gibi tiplerin adlandırırken isim kullan ve UCC yaz.

```
Account, CheckingAccountService, StudentInformation
```

18. Arayüzleri adlandırırken daima isim ya da sıfat kullan ve UCC yaz.

```
Payable, ActionListener
```

19. Bir konuyla ilgili özellikleri, sabiteleri ya da metotları bir araya getiren tiplere çoğul isim ver.

```
AtmProperties, StringUtils
```

20. Enum tiplere tekil isimler ver.

```
Day, Month, Size
```

5. Değişken isimlendirme kuralları

21. Değişken adlandırmalarında isim kullan ve daima LCC yaz.

```
count, firstName, taxRate, orderNumber
```

22. Torbalar için çoğul isimler kullan.

```
Collection<Student> students
```

```
Map<Integer, Player> players
```

23. Boolean değişkenler için uygunsa edilgen fiil (ya da sıfat-fiil) kullan öyle ki başına “is” getirildiğinde anlamlı bir soru olsun. Boolean değişken isimlerinde “is” ya da “are” kullanma.

```
married, tankFilled, seatBooked, tasksFinished
```

Eğer boolean değişken sahip olma durumunu gösteriyorsa ismin sonuna “Installed” gibi bir son ek getirilebilir.

```
gasTankSensorInstalled, radioInstalled
```

24. Özellikler (properties) için daima JavaBean (bean) gösterimini kullan. Bean gösteriminde tüm değişkenler “private” (kalıtım durumunda “protected”) tanımlanır ve bunlara LCC olarak yazılmış set/get metotları ile ulaşılır:

```
private String name;  
  
public String getName() {
```

```

        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

```

Boolean değişkenler için getter olarak “is” ön ekli metot kullanılır:

```

    private boolean deceased = false;

    public boolean isDeceased() {
        return deceased;
    }

    public void setDeceased(boolean deceased) {
        this.deceased = deceased;
    }

```

25. Sık kullanılan yerel değişkenler için aynı kısaltmaları kullan.

```

String s ya da String str
stream için in ve out, exception için e ya da ex

```

26. Kurucu ya da set metotlarına nesne değişkeni ile aynı isimde parametre geç, nesne değişkenine “this” ile ulaş. *Bkz. #24*

27. CamelCase yaklaşımının tek istisnası olarak sabitelerde (public, static ve final) araları alt çizgi “_” ile ayrılmış büyük harfli kelimeler kullan. Başka hiç bir isimde “_” kullanma.

```

public static final double ADDED_VALUE_TAX = 0.18;

```

6. Metot isimlendirme kuralları

28. Get/set metotlarını JavaBean gösterimiyle yaz. *Bkz. #24*

29. Metot isimlerinde daima emir kipi kullan ve LCC yaz.

```
calculateTax(), findOwnerOfAccount()
```

Kaynaklar

- Java Code Conventions September 12, 1997 (Oracle Java Code Conventions <http://www.oracle.com/technetwork/java/codeconv-138413.html>)
- <http://www.ambyssoft.com/downloads/javaCodingStandards.pdf>
- <http://www.ambyssoft.com/downloads/javaCodingStandardsSummary.pdf>
- Google Style of Java <http://google-styleguide.googlecode.com/svn/trunk/javaguide.html>
- Al Vermeulen et al., The Elements of Java Style, CU Press, 2007
- <http://collaboratory.emsl.pnl.gov/docs/collab/sam/CodeStandards.html>
- <http://www.ambyssoft.com/downloads/javaCodingStandards.pdf>
- <http://www.ambyssoft.com/downloads/javaCodingStandardsSummary.pdf>
- <http://www.javacodegeeks.com/2012/10/java-coding-conventions-considered-harmful.html>
- <http://www.iwombat.com/standards/JavaStyleGuide.html>