

Kódolási konvenciók

Horváth Gábor

Miért kellenek konvenciók ?

- ◎ Konvenció ~ szokás, megegyezés, megállapodás
- ◎ Egységességet biztosít
 - Kinézet – olvashatóság
 - Jelölés – értelmezhetőség
- ◎ Karbantartást megkönnyíti
- ◎ Későbbi felhasználás akár saját részről, akár másvalaki által – fontos az átláthatóság

Mit szoktak szabályozni ?

◎ Elnevezések

- Fájlok kiterjesztése (.c, .h, .js, ...)
- Fájlok neve (rövid, egyszerű, legtöbbször nagybetűvel kezdődjön)
- Változók – gyakori pl.: `nameOfVariable`
- Konstansok – pl. C: `GLOBAL_CONSTANT`
- Metódusok, függvények – pl. ige legyen az elején -> `szamoljKiValamit`
- Ne legyenek nemzetközi karakterek...

Magyar jelölés

- ◎ Hungarian notation, Charles Simonyi
- ◎ Eredetileg C nyelvhez
- ◎ A változók elnevezésében a prefixek extra információt tartalmaznak
 - rendszer magyar jelölés: adat típusa
pl. arru6Tomb -> arr + u + 6 + Tomb
előjel nélküli egészek 6 elemű tömbje
 - alkalmazás magyar jelölés: felhasználás típusa
pl. rwNev: egy írható és olvasható név

Mit szoktak szabályozni ?

◎ Felépítés

- Milyen sorrendben legyenek az importálások, deklarációk, függvények, egyáltalán hol lehet ezeket elhelyezni
- Kötelező elemek, mint kezdő megjegyzések a fájl elején, vagy lezárás a végén (end.), szükséges metódusok (pl. konstruktor)
- Sorok hossza és száma (olvashatóság)
pl. C és JS is 78-80 karakter hosszút javasol
- Fájlonként 1 osztály legyen (ez általános)

Mit szoktak szabályozni ?

◎ Megjelenés

- Sortörések megvalósításának módja
- Zárójelek helye
- Kommentek helye (sor végén / új sor)
- Behúzás mértéke (hány szóközt jelent)
- Soronként 1 parancs / 1 deklaráció
pl. `char **a, *x;` helyett
`char **a = 0; // lehet magyarázni`
`char *x = 0; // mindegyiknél külön`

Mit szoktak szabályozni ?

- ◎ Megjegyzések
 - Mihez fűzzünk?
 - Változók mögé
 - Eljárások elé
 - Nem egyértelmű művelethez
 - Ne magyarázzunk feleslegesen!
`i = 0; // Set i to zero.`
 - Ügyeljünk a kommentek mennyiségére!
(a túl sok megjegyzés zavaró lehet)

Java konvenciók

◎ Fájlnevek

- .java kiterjesztés a forrásfájl esetén
- .class a lefordított bájtkód
- osztály neve = fájlnev

◎ Fájlok felépítése

- Hossz < ~2000 programsor
- Minden fájlba kell publikus osztály / interfész
Privátok is lehetnek egy fájlon belül, de a publikus szerepeljen első helyen.

Java konvenciók

◎ Felépítés – sorrend:

- Kezdő megjegyzések

```
/*  
 * Osztálynév  
 *  
 * Verzióhoz kötődő információk  
 *  
 * Dátum  
 *  
 * Copyright megjegyzések  
 */
```

- Csomag- és importdeklarációk

- `package java.awt;`
`import java.awt.peer.CanvasPeer;`

Java konvenciók

- Osztály- és interfészdeklarációk (sorrend)
 1. Dokumentációs megjegyzés
 2. Osztály / interfész fej
 3. Implementációs megjegyzések (ha kellenek)
 4. Osztályváltozók
 5. Példányváltozók
 6. Konstruktorkok
 7. Metódusok

Java konvenciók

- ◎ Behúzás: alapvetően 4 szóköznyi hely
 - ◎ Sorhossz < 80 karakter
 - ◎ Sortörési elvek:
 - Egy vessző után vagy egy operátor előtt
 - Inkább magas szintű helyen, ha lehet
 - Az új sor igazítása az előző sorban lévő, azonos szintű kifejezés kezdetéhez
 - Esetleg 8 szóközös behúzás, ha sűrű a kód
- Pl.: `longName1 = longName2 * (longName3 + longName4 - longName5)
+ 4 * longName6; // Ajánlott`

Java konvenciók

```
//Ne használjuk ezt a behúzást
if ((condition1 && condition2)
    || (condition3 && condition4)
    || (condition5 && condition6)) { // A rossz sortörések miatt
doSomethingAboutIt(); // ez a sor könnyen
} // eltéveszthet ő
```

```
//Helyette használjuk ezt
if ((condition1 && condition2)
    || (condition3 && condition4)
    || (condition5 && condition6)) {
doSomethingAboutIt();
}
```

Java konvenciók

◎ Megjegyzések

- Implementációs: a forráskód magyarázata
 - Blokk: nagyobb méret, üres sor előtte + utána

```
/*  
 * Ez egy blokk megjegyzés.  
 */
```

- Egysoros megjegyzések: külön sorba, előtte üres

```
/* Egysoros megjegyzés. */
```

Java konvenciók

- Utómegjegyzések: egy sorba a kódrészlettel, kellő távolságra attól.

```
Pl.      return TRUE;      /* speciális eset */
```

- Sorvégi megjegyzések: // karakterekkel. Sor végére szokták, de teljes sorra is használható.

```
return false;      // miért itt?
```

- Dokumentációs megjegyzések

- A specifikációval kapcsolatos magyarázat
- /** tartalom */
- Automatikusan feldolgozható (javadoc)

Java konvenciók

- Jellemzik a Java osztályokat, interfészeket, konstruktorokat, metódusokat és adattagokat
- Pontosán a kapcsolódó deklaráció előtt kell szerepelnie, a feldolgozó arra vonatkoztatja

◎ Deklarációk

- Soronként 1 ajánlott (átláthatóság, megj.)
- Érdeemes egyből inicializálni is, ha lehet (ha a kezdőérték nem egy számítás eredménye)
- A blokk elejére tegyük (kivéve: for ciklus)
- Ne használjunk ugyanolyan névvel változókat!

Java konvenciók

◎ Zárójelek

- Nyitó { a sor végén legyen
- Záró } új sorban, önállóan

◎ Utasítások

- Minden sor 1 utasítást tartalmazzon
- Mindig legyen zárójel a megfelelő helyen, még ha nem is lenne muszáj, pl.:

```
if (feltétel) {  
    utasitas;  
}
```


Java konvenciók

◎ Üres sorok

- A logikailag összetartozó kódrészek kiemelésére
- Növeli az olvashatóságot
- 2 üres sor kell
 - A forrás fájl két bekezdése között
 - Osztály- és interfészdefiníciók között
- 1 üres sor
 - Metódusok között. Blokk v egysoros megj. előtt
 - Metódus változói és az első utasítás között
 - Metódusban a különböző logikai szakaszok között

Java konvenciók

◎ Szóközök

- Kulcsszót követő kerek zárójel elé, pl.:
`while (true) {`
- Minden kétoperandusú operátor elé és mögé (a `.` kivételével), pl.: `a += c + d`
- For utasítás kifejezéseinél: `for (expr1; expr2; expr3)`
- Típuskényszerítések után, pl.: `(byte) aNum`
- TILOS: unáris operátor és operandusa között, valamint metódus neve és nyitó `(` között, pl.:
`i++; -i; pelda(int i);`

Java elnevezési konvenciók

● Csomagok

- Egyedi csomagnevek kisbetűkkel, prefix egy top-level domain név (pl. com, org) vagy országok kétbetűs angol azonosítója
- Többi rész tetszőleges, általában divízió, hivatal, projekt, gép, vagy felhasználó név

● Osztályok, interfészek

- Főnevek legyenek, minden részszó nagybetűvel kezdődjön -> class SajatOsztaly
- Egyszerű, kifejező legyen, rövidítés nélkül

Java elnevezési konvenciók

● Metódus

- Kisbetűvel kezdődő ige legyen az elején
- Minden belső szó nagybetűvel kezd
- Pl.: `getBackgroundColor()`;

● Változók

- Kisbetűs kezdés, belső szavak nagybetűvel
- Nem kezdhet `_` vagy `$` jellel
- A név tükrözze a változó célját

Java elnevezési konvenciók

- Egy-karakteres változó csak ideiglenes, eldobható legyen (i, j, k, m, n – egész és c, d, e – karakter típusúak számára)

○ Konstansok

- Végig nagybetűk
- Szavak elválasztása _ jellel
- Pl.: `static final int MAX_WIDTH = 999;`

Megjegyzések

- ◎ A konvenciók hasznosak, de vannak, akik úgy látják, hogy
 - feleslegesek, amíg az emberek logikusak
 - túl sok struktúrát erőltetnek az emberekre
 - csökkentik a kreativitást
 - „People ignore standards anyway.”
- ◎ Az „általános” konvenciók mellett az egyes cégek gyakran alkalmaznak egyéni, „házon belüli” szabályokat is.