دومین موضوعی که در برنامه نویسی امن با زبان جاوا مورد توجه قرار می‌گیرد اعلان‌ها و مقدار دهی اولیه است. در ادامه به سومین قانون (DCL02-J) ذیل این موضوع پرداخته خواهد شد.

قانون DCL02-J – عدم تغییر مولفه‌های مجموعه در حین استفاده از عبارت for بهبود یافته

عبارت for بهبود یافته برای تکرار(iteration) از طریق مجموعه‌ها و آرایه‌ها طراحی شده است.

مشخصات زبان جاوا (JLS) مثال زیر را برای عبارت for بهبود یافته ارائه کرده است:

قالب عبارت for بهبود یافته با عبارت for پایه‌ای یکی است:

```
for (I #i = Expression.iterator(); #i.hasNext(); ) {
    {VariableModifier} TargetType Identifier =
        (TargetType) #i.next();
    Statement
}
```

i#، شناسه ای است که به صورت خودکار تولید شده است و از هر شناسه دیگری در محدوده ای (Scope) که در آن عبارت forبهبود یافته وجود دارد متمایز است.

برخلاف عبارت for پایه‌ای، انتساب‌ها به متغیر حلقه نمی‌تواند ترتیب تکرار حلقه را برای مجموعه شی‌های مربوطه تحت تاثیر قرار دهد. در نتیجه، انتساب مقداری به متغیر حلقه معادل تغییر متغیری محلی در بدنه حلقه است که مقدار اولیه آن متغیر محلی، همان شیء است که توسط متغیر تکرار حلقه مورد ارجاع قرار می گیرد. این تغییر لزوماً نادرست نیست اما می‌تواند منجر به ابهام در عملکرد حلقه یا برداشت اشتباه در پیاده سازی عبارت for بهبود یافته گردد.

تمام متغیرهای حلقه عبارت for بهبود یافته را به صورت final اعلان کنید. اعلان به صورت final منجر به برچسب زدن و رد کردن هر انتسابی به متغیر حلقه توسط کامپایلرهای جاوا می شود.

یک نمونه ناسازگار با قانون

در این مثال مجموعه‌ای از اعداد صحیح با استفاده از حلقه for بهبود یافته پردازش می‌شود. هدف از این برنامه، تغییر مولفه‌ای در مجموعه در حال پردازش است:

```
List<Integer> list = Arrays.asList(new Integer[] {13, 14, 15});
boolean first = true;

System.out.println("Processing list...");
for (Integer i: list) {
  if (first) {
    first = false;
    i = new Integer(99);
  }
  System.out.println(" New item: " + i);
  // Process i
}

System.out.println("Modified list?");
for (Integer i: list) {
  System.out.println("List item: " + i);
}
```

همانطور که خروجی این برنامه نشان می‌دهد، این کد محتوای لیست را تغییر نمی‌دهد:

```
Processing list...
New item: 99
New item: 14
New item: 15
Modified list?
List item: 13
List item: 14
List item: 15
```

راه حل سازگار با قانون

اعلان متغیر i به صورت final منجر به عدم اجازه انتساب مقدار جدیدی به متغیر i توسط کامپایلر می‌شود:

```
// ...
for (final Integer i: list) {

// ...
```

راه حل سازگار با قانون

در این راه حل، لیست "تغییریافته" پردازش شده اما لیست اصلی بدون تغییر باقی می‌ماند:

```
// ...

for (final Integer i: list) {
  Integer item = i;
  if (first) {
    first = false;
    item = new Integer(99);
  }
  System.out.println(" New item: " + item);
  // Process item
}

// ...
```

ارزیابی خطر

انتساب مقادیر به متغیر حلقه یک حلقه for بهبود یافته ترتیب کلی تکرار را تحت تاثیر قرار نمی‌دهد، منجر به اشتباه کردن برنامه نویس می‌شود، و می‌تواند داده را در حالتی شکننده یا ناسازگار قرار دهد.

Rule	Severity	Likelihood	Remediation Cost	Priority	Level
DCL02-J	Low	Unlikely	Low	P3	L3

تشخیص خودکار

این قانون با تحلیل استاتیک به راحتی اعمال می‌شود.