

دومین موضوعی که در برنامه نویسی امن با زبان جاوا مورد توجه قرار می‌گیرد مربوط به اعلان‌ها و مقدار دهی اولیه است. در ادامه به اولین قانون (DCL00-J) ذیل این موضوع پرداخته خواهد شد.

قانون J-DCL00 – جلوگیری از دوره‌های مقدار دهی اولیه کلاس (ادامه)

در بخش اول مقاله، این قانون توضیح به همراه نمونه کد ناسازگار و راه حل آن شرح داده شد. در این بخش از مقاله، دو نمونه دیگر از کد ناسازگار و راه حل‌های سازگار با آن ارائه خواهد شد.

یک نمونه ناسازگار با قانون (دور بین کلاسی)

در این مثال ناسازگار، دو کلاس با متغیرهای استاتیک وجود دارد که مقادیر آنها به همدیگر وابسته هستند. این دور هنگام دیدن کلاس‌ها در کنار یکدیگر آشکار خواهد شد، اما در زمان دیدن کلاس‌ها به طور مجزا، احتمال در نظر نگرفتن آن وجود دارد.

```
class A {
    public static final int a = B.b + 1;
    // ...
}

class B {
    public static final int b = A.a + 1;
    // ...
}
```

ترتیب مقدار دهی اولیه کلاس‌ها می‌تواند متغیر باشد، به طوریکه مقادیر متفاوتی را برای $A.a$ و $B.b$ محاسبه کند. اگر کلاس A ابتدا مقداردهی اولیه شود، $A.a$ برابر با مقدار ۲ و مقدار $B.b$ ، ۱ است و اگر ابتدا کلاس B مقدار دهی اولیه شود، این مقادیر برعکس خواهد بود.

راه حل سازگار با قانون (دور بین کلاسی)

این راه حل سازگاری، دور بین کلاسی را از طریق حذف وابستگی A به B برطرف می‌کند:

```
class A {
    public static final int a = 2;
    // ...
}

class B {
    public static final int b = A.a + 1;
    // ...
}
```

از طریق حذف دور و مستقل از ترتیب مقدار دهی اولیه، همیشه مقادیر اولیه $A.a$ برابر با ۲ و $B.b$ برابر با ۳ است.

یک نمونه ناسازگار با قانون

برنامه نویس در این مثال، متغیر استاتیک کلاس اول را با استفاده از متد استاتیک در کلاس دوم مقدار دهی اولیه می‌کند اما این متد وابسته به متدی استاتیک در کلاس اول است:

```
class A {
    public static int a = B.b();
    public static int c() { return 1; }
}

class B {
    public static int b() { return A.c(); }
}
```

این کد، به درستی با استفاده از زبان ماشین جاوا و بدون توجه به اینکه آیا A اول بارگذاری شده است یا B، مقدار اولیه A.a را به 1 مقداردهی اولیه می‌کند. اما مشخصات زبان جاوا (JLS) این تضمین را فراهم نمی‌کند که A.a به درستی مقدار دهی اولیه شود. به علاوه، دور در مقداردهی اولیه، نگهداری سیستم را سخت‌تر می‌کند.

راه حل سازگار با قانون

این راه حل سازگاری، متد () c را به کلاس B منتقل و دور را می‌شکند:

```
class A {
    public static int a = B.b();
}

class B {
    public static int b() { return B.c(); }
    public static int c() { return 1; }
}
```

ارزیابی خطر

دوره‌های مقدار دهی اولیه ممکن است منجر به نتایج غیرمنتظره شوند.

Rule	Severity	Likelihood	Remediation Cost	Priority	Level
DCL00-J	Low	Unlikely	Medium	P2	L3