Java
Visibility and Atomicity (VNA)

Mario Alviano

University of Calabria, Italy

A.Y. 2016/2017

Ensure visibility when accessing shared primitive variables

- ✗ Don't read stale cached copies of variables
- ✓ Declare shared variables `volatile`
- ✓ Use `Atomic*` classes
- ✓ Properly synchronize methods

```
https://www.securecoding.cert.org/confluence/
display/java/VNA00-J.+Ensure+visibility+when+
accessing+shared+primitive+variables
```

Ensure visibility of shared references to immutable objects

- ✗ Mutable references may be cached
- ✓ Declare them `volatile`
- ✓ Use `AtomicReference`
- ✓ Synchronize access

```
https://www.securecoding.cert.org/confluence/
display/java/VNA01-J.+Ensure+visibility+of+shared+
references+to+immutable+objects
```

Ensure that compound operations on shared variables are atomic

- ✗ Just because it's one expression, doesn't mean it's atomic!
- ✗ Use of `volatile` may be insufficient
- ✗ Use of `Atomic*` may be insufficient
- ✓ Synchronize methods

```
https://www.securecoding.cert.org/confluence/
display/java/VNA02-J.+Ensure+that+compound+
operations+on+shared+variables+are+atomic
```

Do not assume that a group of calls to independently atomic methods is atomic

- ✗ Race conditions on the group of calls are still possible
- ✓ Synchronize methods or blocks of instructions

```
https://www.securecoding.cert.org/confluence/
display/java/VNA03-J.+Do+not+assume+that+a+group+
of+calls+to+independently+atomic+methods+is+atomic
```

Ensure that calls to chained methods are atomic

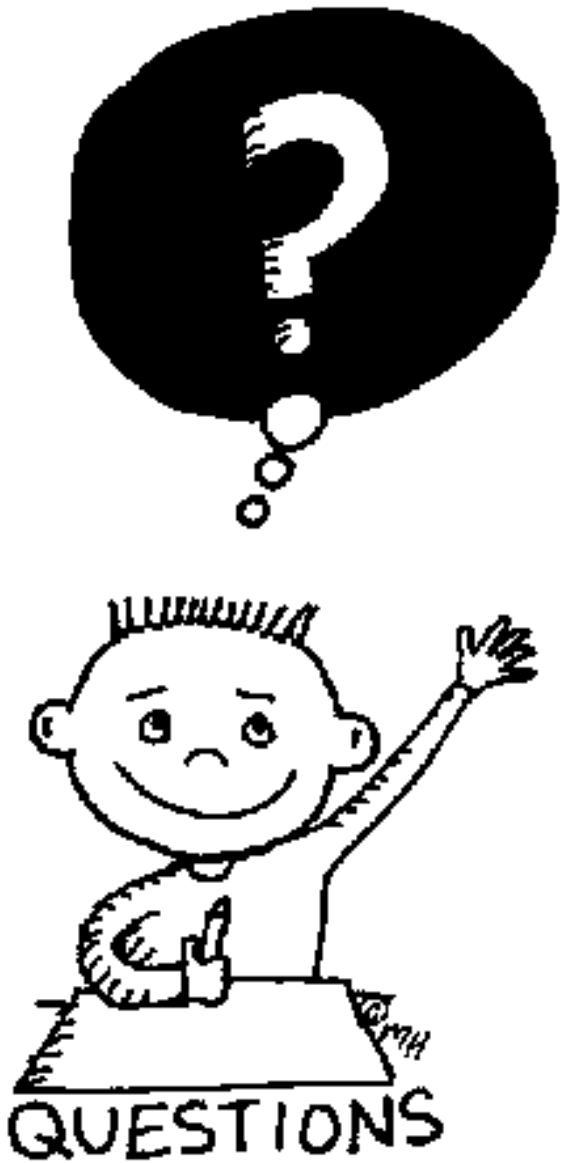
- ✗ Don’t let the object to change in the middle of a chain
- ✗ Don’t assume that the user will synchronize the chain
- ✓ Implement the Builder pattern

```
https://www.securecoding.cert.org/confluence/
display/java/VNA04-J.+Ensure+that+calls+to+
chained+methods+are+atomic
```

Ensure atomicity when reading and writing 64-bit values

- ✗ A `long` variable may be considered as two 32-bit variables
- ✗ A `double` variable may be considered as two 32-bit variables
- ✓ Declare shared `long` and `double` variables as `volatile`
- ✓ Synchronize the access to these variables

```
https://www.securecoding.cert.org/confluence/
display/java/VNA05-J.+Ensure+atomicity+when+
reading+and+writing+64-bit+values
```



END OF THE LECTURE