In-Lab Demo

Thursday September 11

$$\forall i : \text{INTEGER} \mid 1 \le i \le 10 \bullet i^2 \le 100$$

type — range — constraint

collection

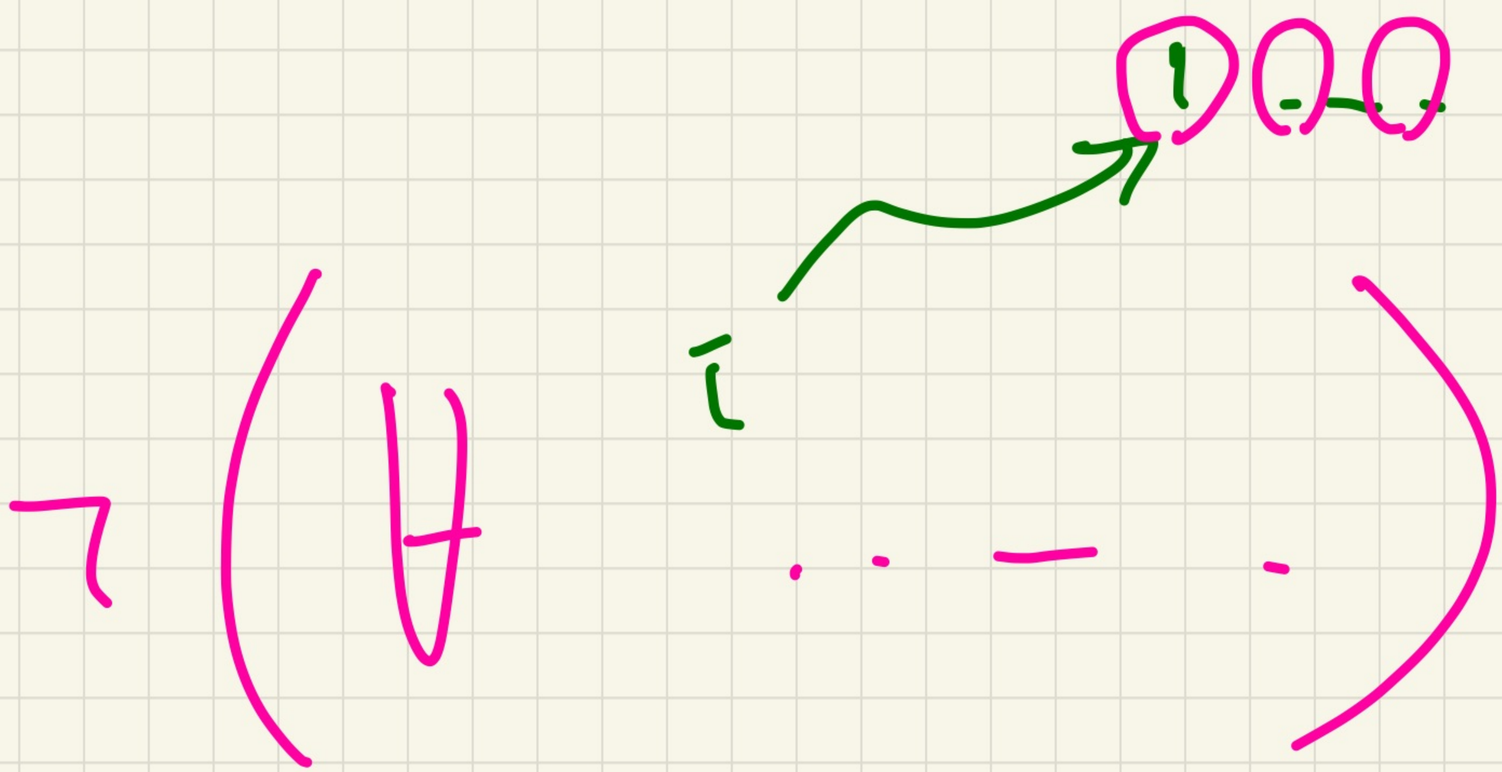
new local variable

across ... as ...

loop / all / some

body

end




```
Result :=
    not (across 1 |..| 10 as i
    all i.item * i.item < 100
    end)
check Result end
```

not an int,
neither
a cursor pointing to
an integer.

across ____ is ____

all

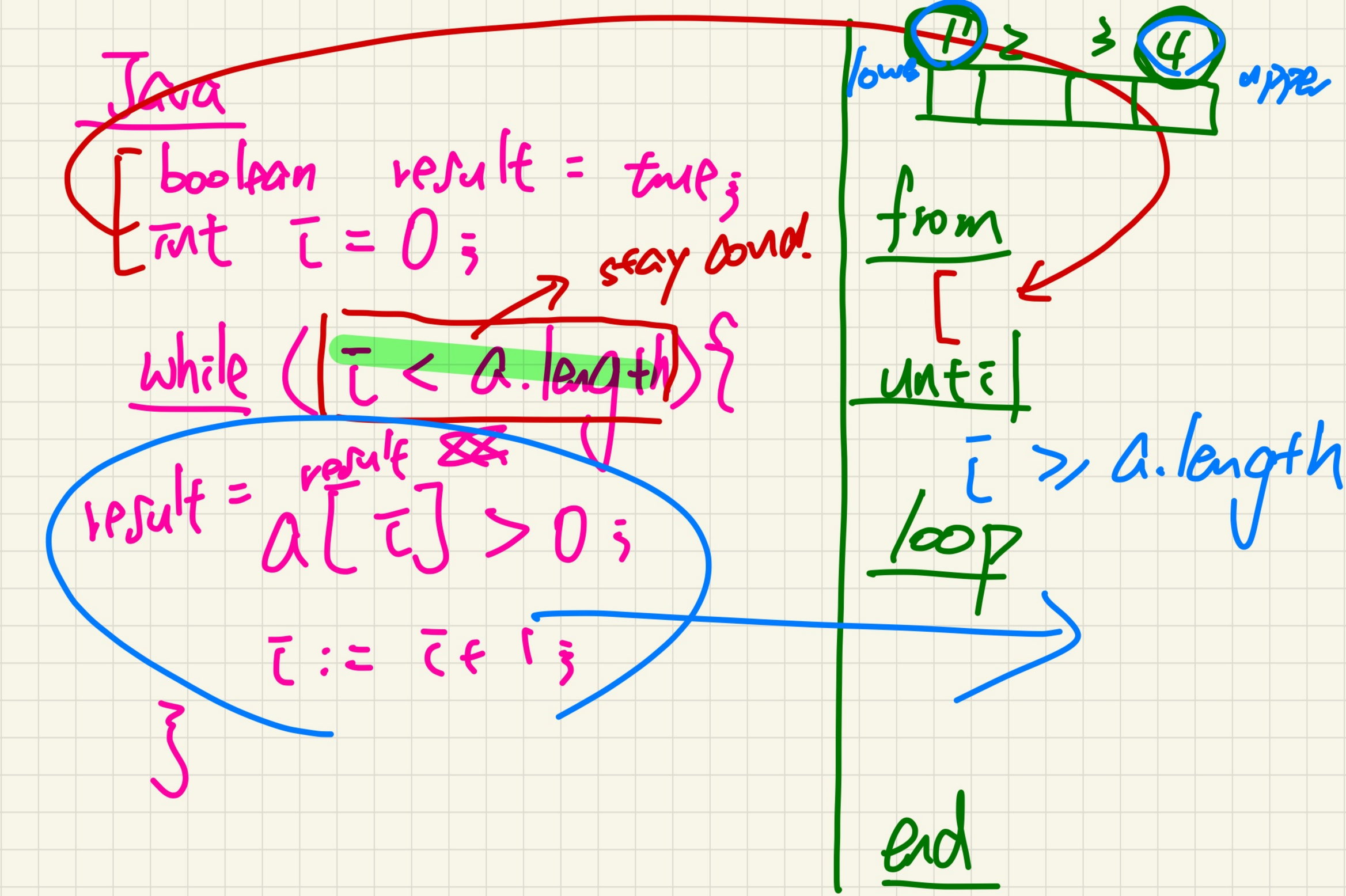
Boolean EXPRESSION

across ____ is ____

Jump

end

end


Java
boolean result = true;
int i = 0;
while (i < a.length) {
stay cond.
result = a[i] > 0;
i := i + 1;
}
1
2
3
4
lower
upper
from
until
i ≥ a.length
loop
end

f (. . . .)

-- comments for f

require

Boolean expressions

contracts

local

instructions

implementation

do

:= f from .. until

ensure

end

$$\neg(\forall i : \text{INTEGER} \mid 1 \le i \le 10 \cdot i^2 < 100)$$

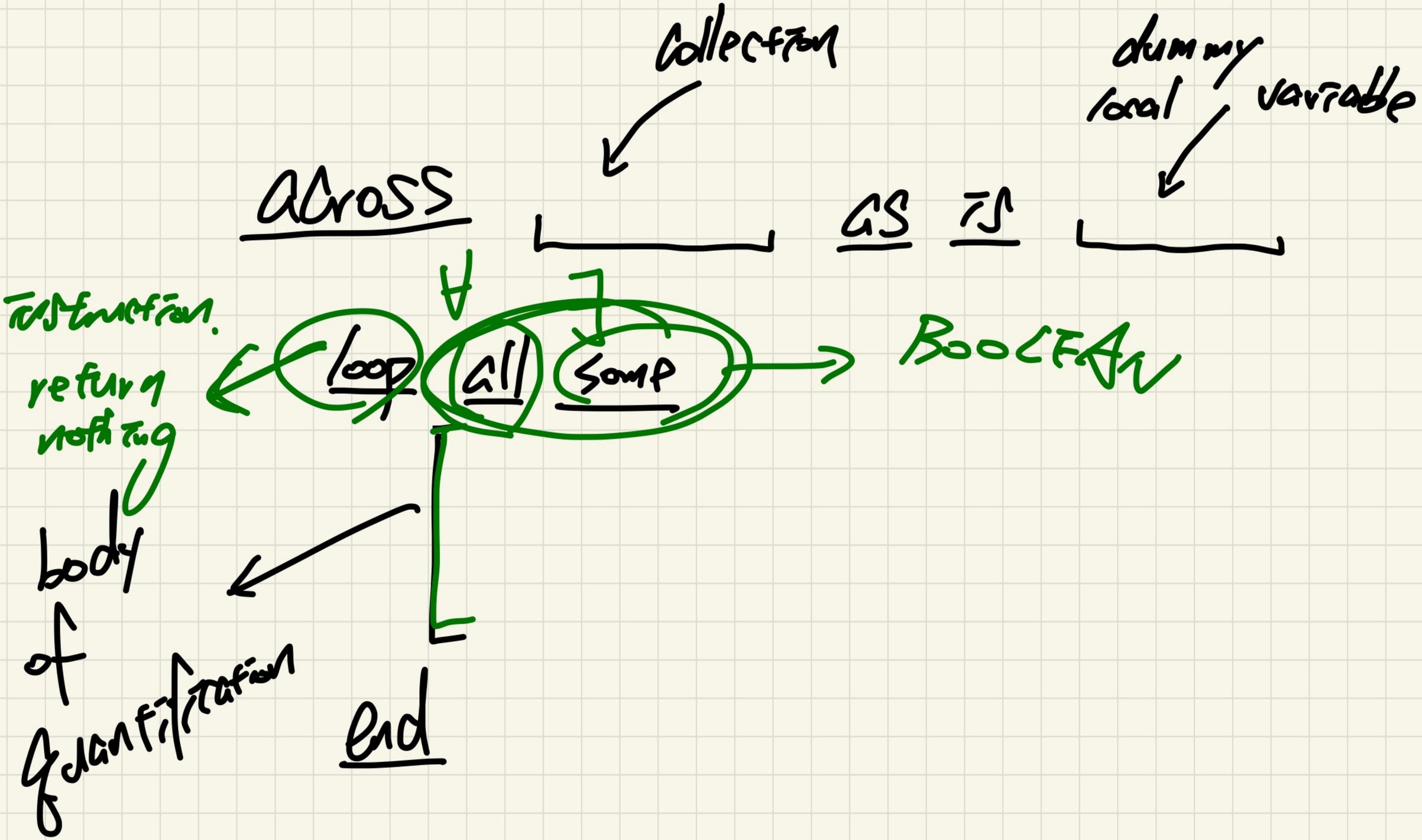
type

dummy variable

range

constraint

```
int x = while (...) {
        ⋮
        }
```


collection
dummy local variable
across
as
loop
all
some
BOOLEAN
Transformation.
returning
modifying
body
of
quantification
end

f(...)
-- comments for f

Guarantees
specification

require

Boolean
exp?

local

implementation

do

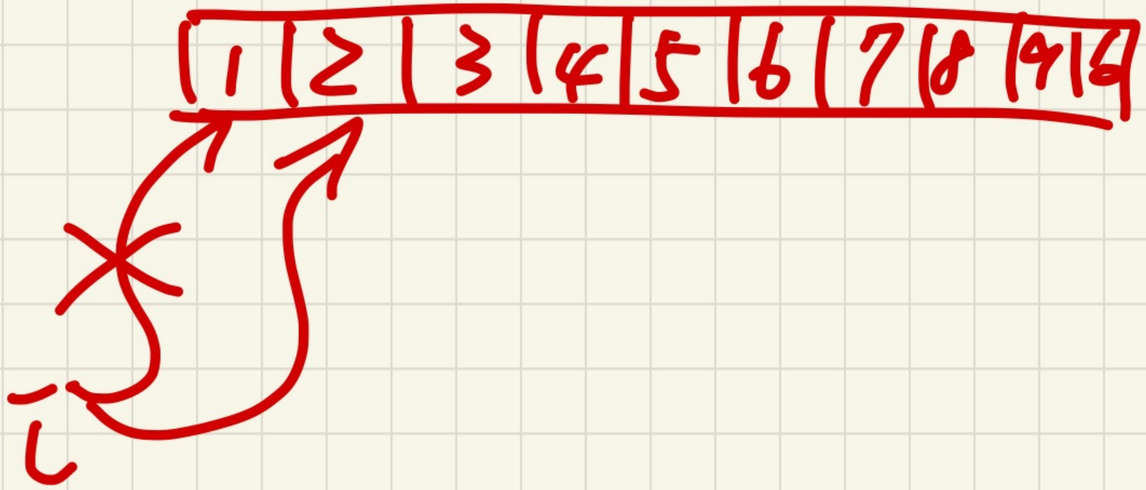
instructions
across loop
from until

ensure

end

```
Result :=
    across
        1 |..| 10 is i
    some
        i * i < 100
    end
```

```
Result :=
    across
        1 |..| 10 as i
    some
        i.item * i.item < 100
    end
```



Cursor pointing to an integer

$$\forall x \mid R(x) \cdot P(x)$$

$$\equiv \neg\left(\exists x \mid R(x) \cdot \neg P(x)\right)$$

Java

Init
```
int sum = 0;
int i = 0;
while (i < a.length) {
```
stay condition

Body
```
    sum += a[i];
    i++;
}
```

from

until

$\neg(i < a.length)$

$\equiv$

$i \geq a.length$

loop

end

1 2 3 4 5

a.lower

a.count

a.upper