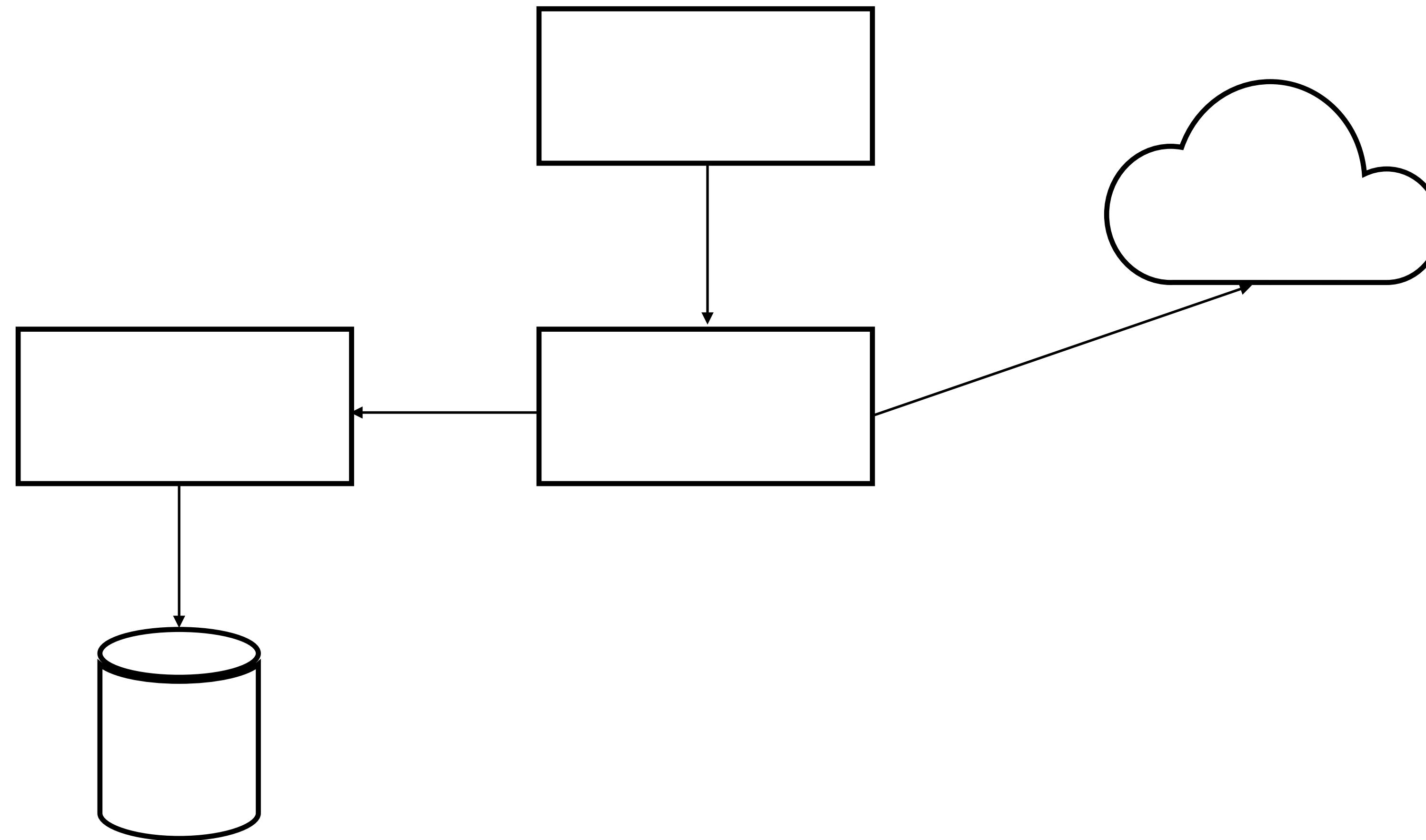


Better Understanding through Bytecode

JBCN, July 2022

Jason Clark
Principal Engineer & Architect
New Relic



Where We're Going

- **Tools for digging**
 - How the JVM runs
 - Just enough bytecode to be dangerous
- **Examples**
 - Losing concurrent updates
 - Tail recursion
 - Coroutines



You can understand
the layer below your
comfort zone



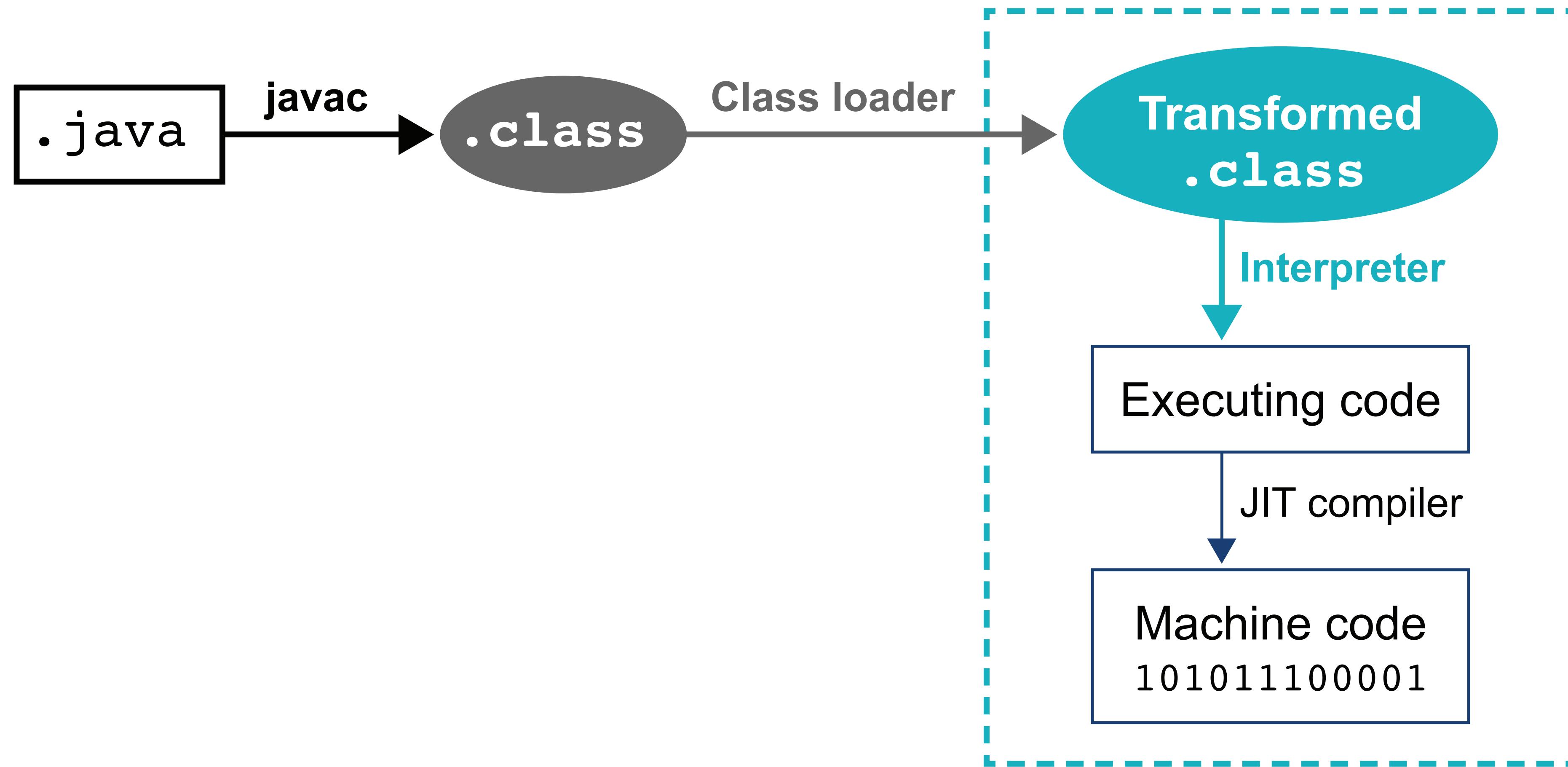
@jasonrclark



You don't have
to understand it
all *right now*

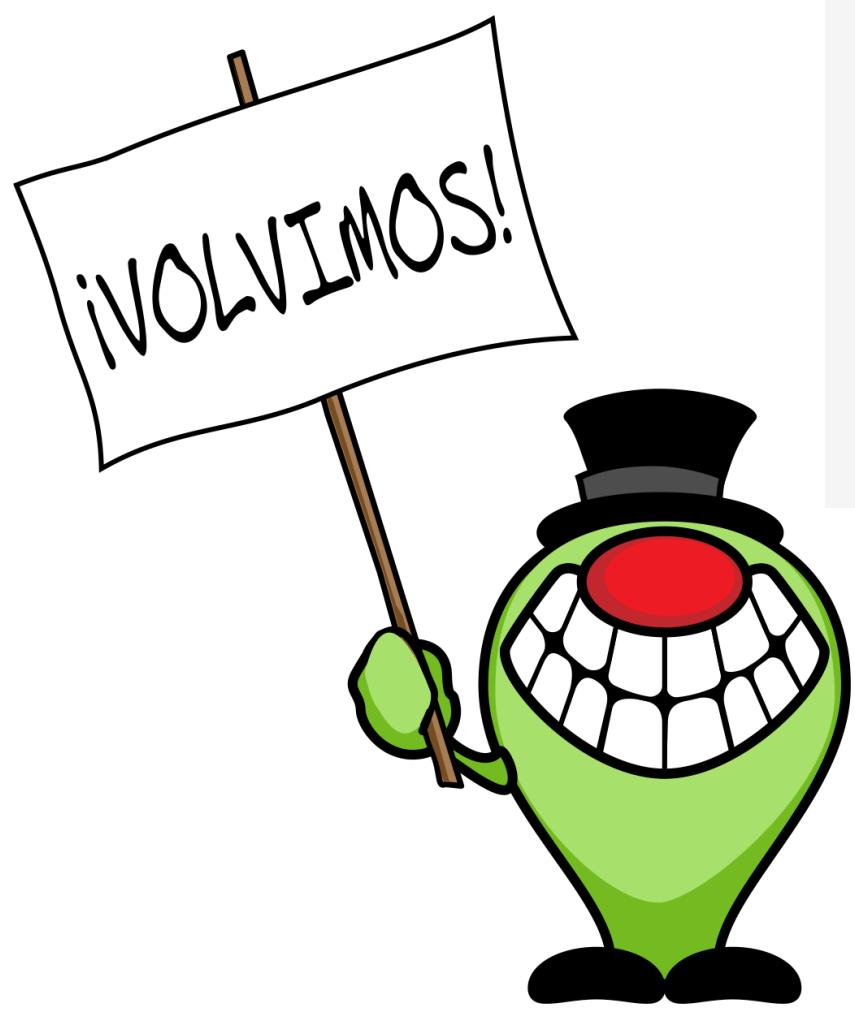
How the JVM Runs

JVM Lifecycle





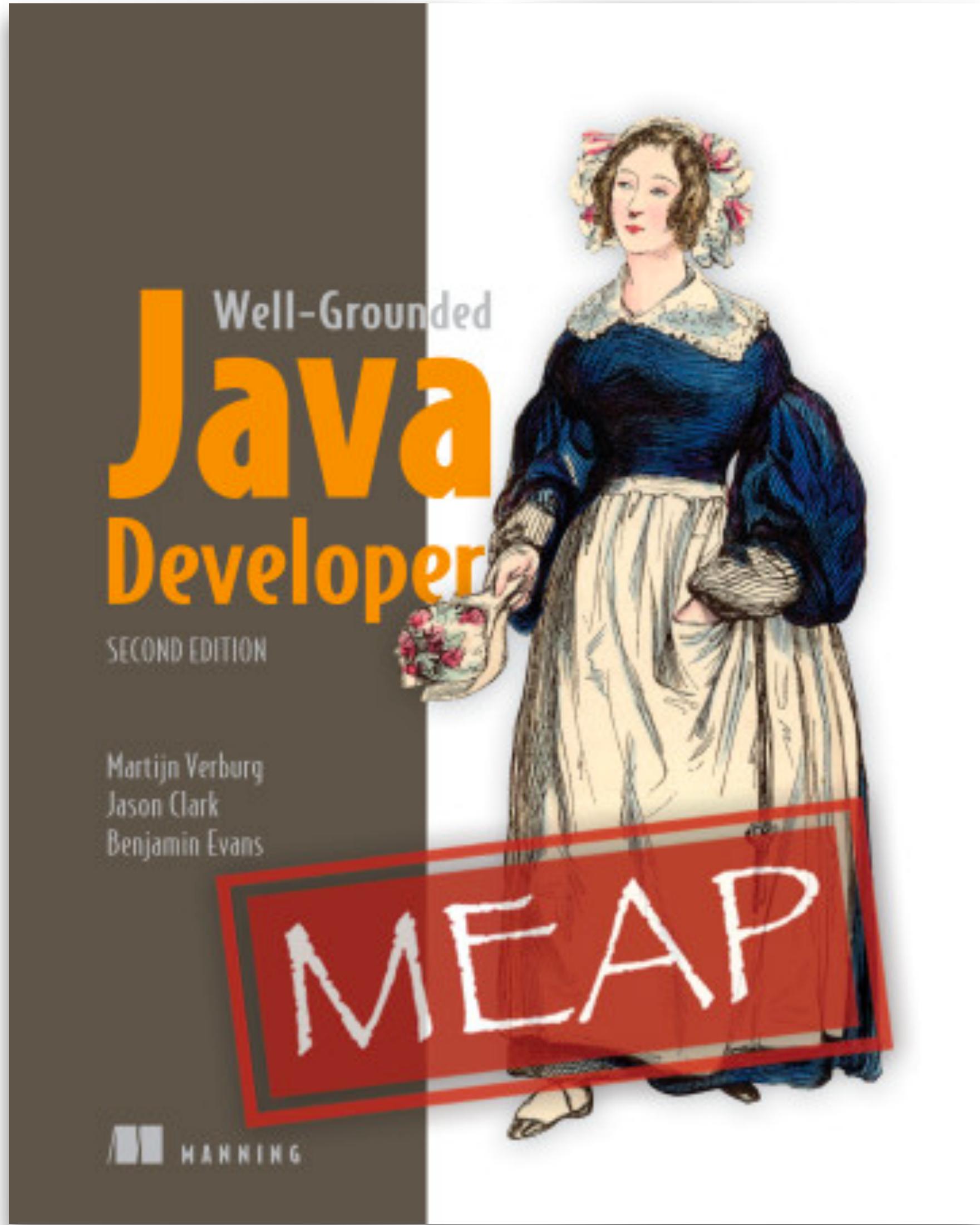
https://www.youtube.com/watch?v=fozu0H_5C4I



A YouTube video thumbnail for "Fantastic Bytecodes and How To Interpret Them" by Ben Evans. The thumbnail features a blue background with white clouds. The title "Fantastic Bytecodes & How To Interpret Them" is centered in a white, serif font. In the bottom left corner, there's a small image of Ben Evans speaking. The bottom right corner shows a small illustration of a dark, mechanical-looking robot or alien. The video player interface at the bottom includes a play button, volume control, and a progress bar showing 0:06 / 55:29. The URL "https://www.youtube.com/watch?v=fozu0H_5C4I" is overlaid at the bottom of the thumbnail.

https://www.youtube.com/watch?v=fozu0H_5C4I





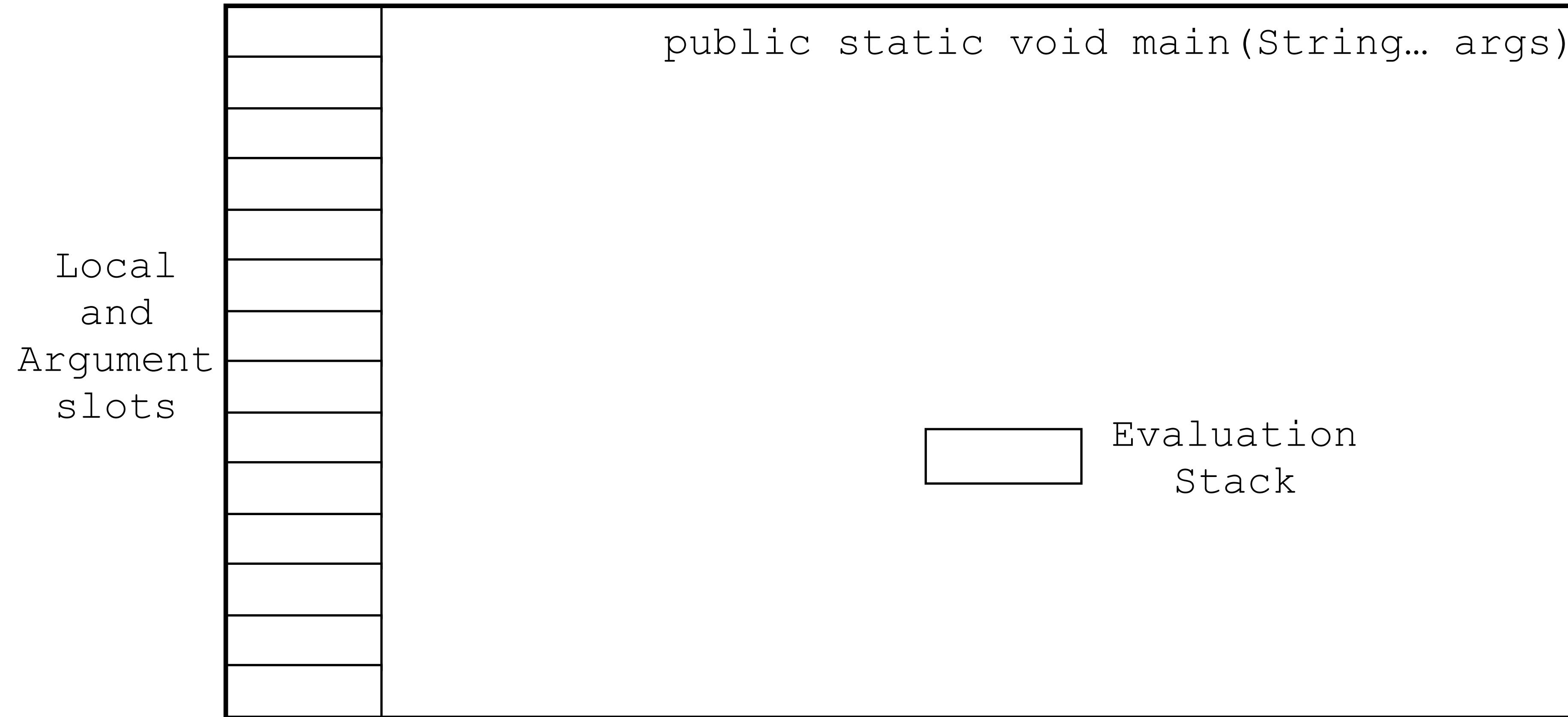
<https://www.manning.com/books/the-well-grounded-java-developer-second-edition>

Two Stacks

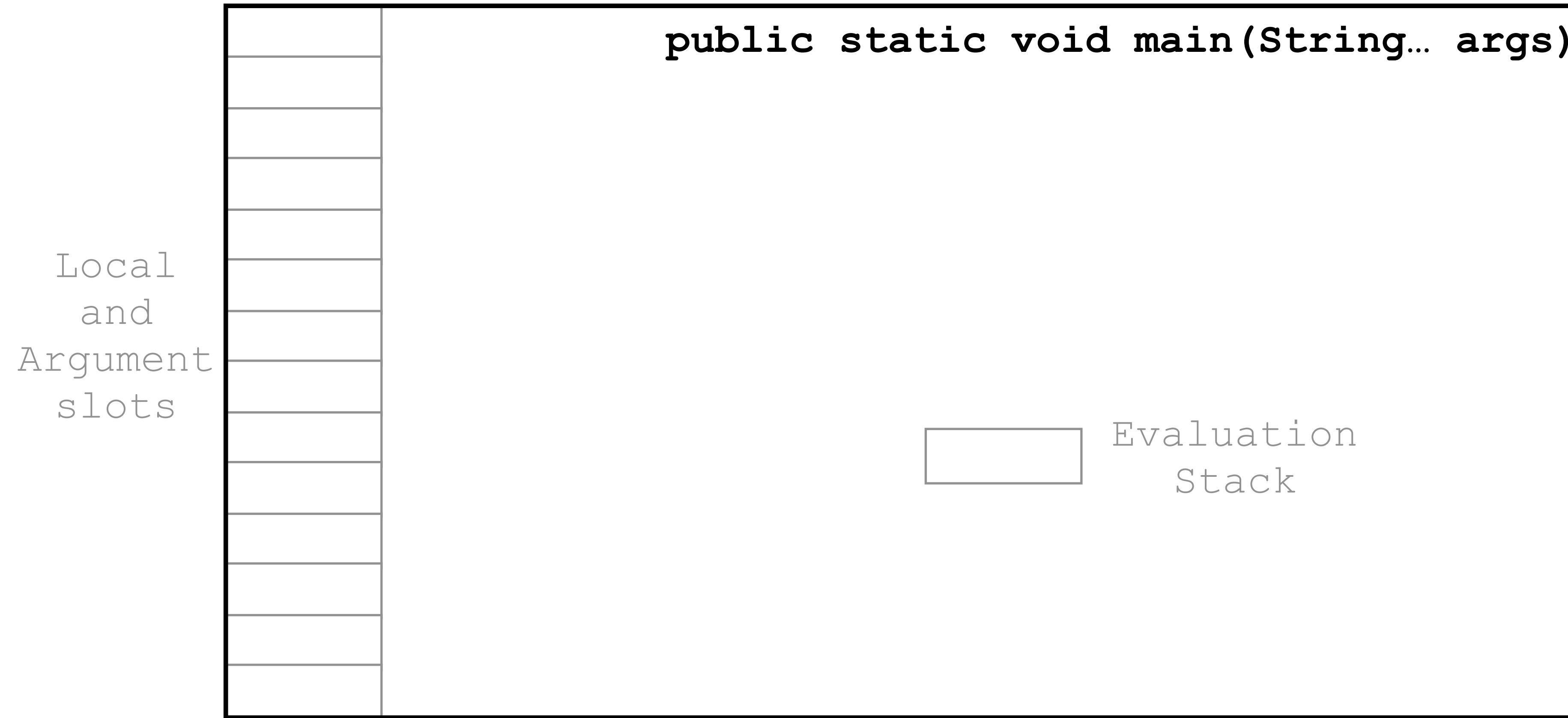
Call Stack

```
Exception in thread "main" java.lang.NullPointerException  
        at Methods.third(Methods.java:16)  
        at Methods.second(Methods.java:11)  
        at Methods.first(Methods.java:7)  
        at Methods.main(Methods.java:3)
```

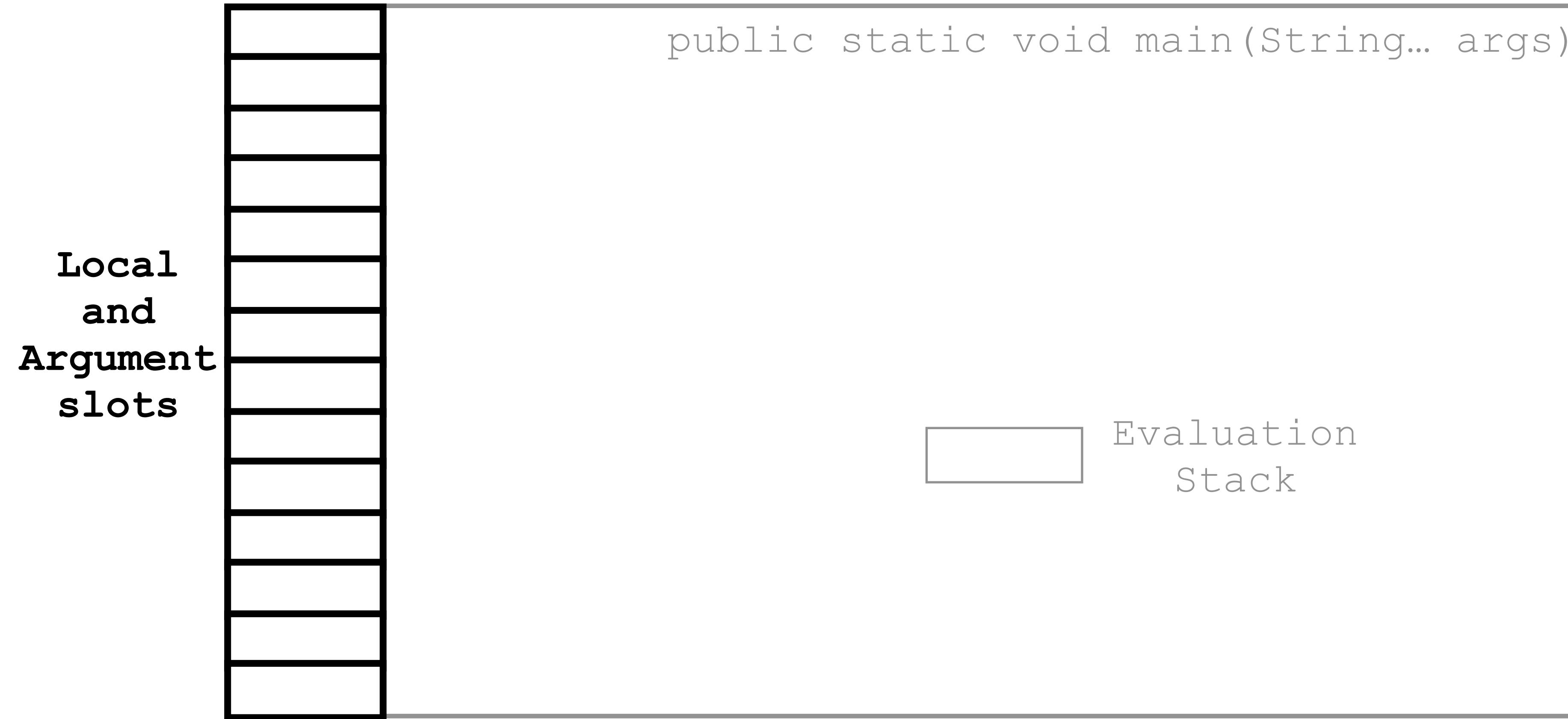
Call Frame



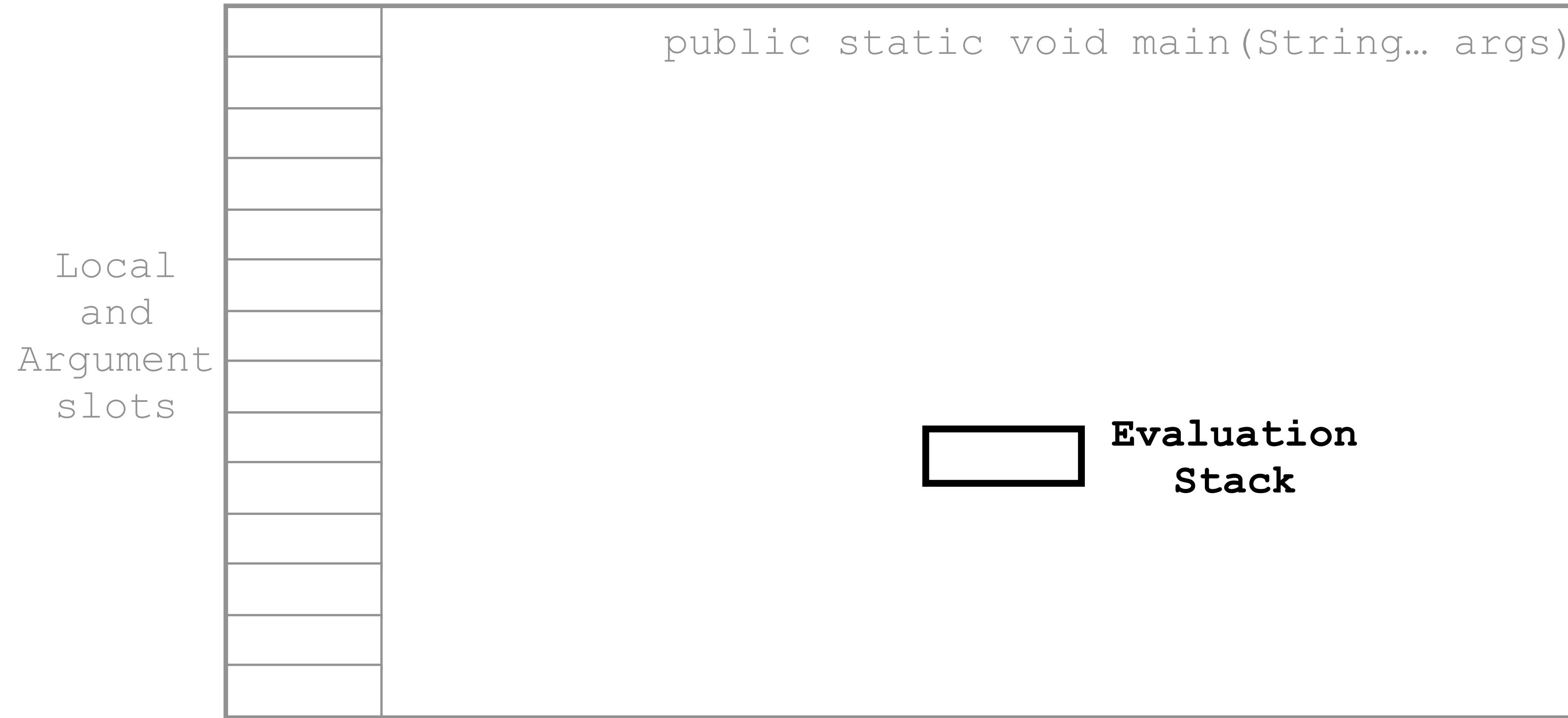
Call Frame



Call Frame



Call Frame





The JVM is a stack machine

Just Enough Bytecode...

Our First Bytecodes

```
public class Add {  
    public int add(int x, int y) {  
        return x + y;  
    }  
}
```

Our First Bytecodes

```
$ javap -c Add
```

Compiled from "Add.java"

```
public class Add {  
    public Add();
```

Code:

```
  0: aload_0  
  1: invokespecial #1 // Method java/lang/Object."<init>"  
  4: return
```

```
public int add(int, int);
```

Code:

...



Look for what you recognize!



@jasonrclark

Our First Bytecodes

```
$ javap -c Add
```

```
Compiled from "Add.java"
```

```
public class Add {
```

```
    public Add();
```

```
    Code:
```

```
        0: aload_0
```

```
        1: invokespecial #1 // Method java/lang/Object."<init>"
```

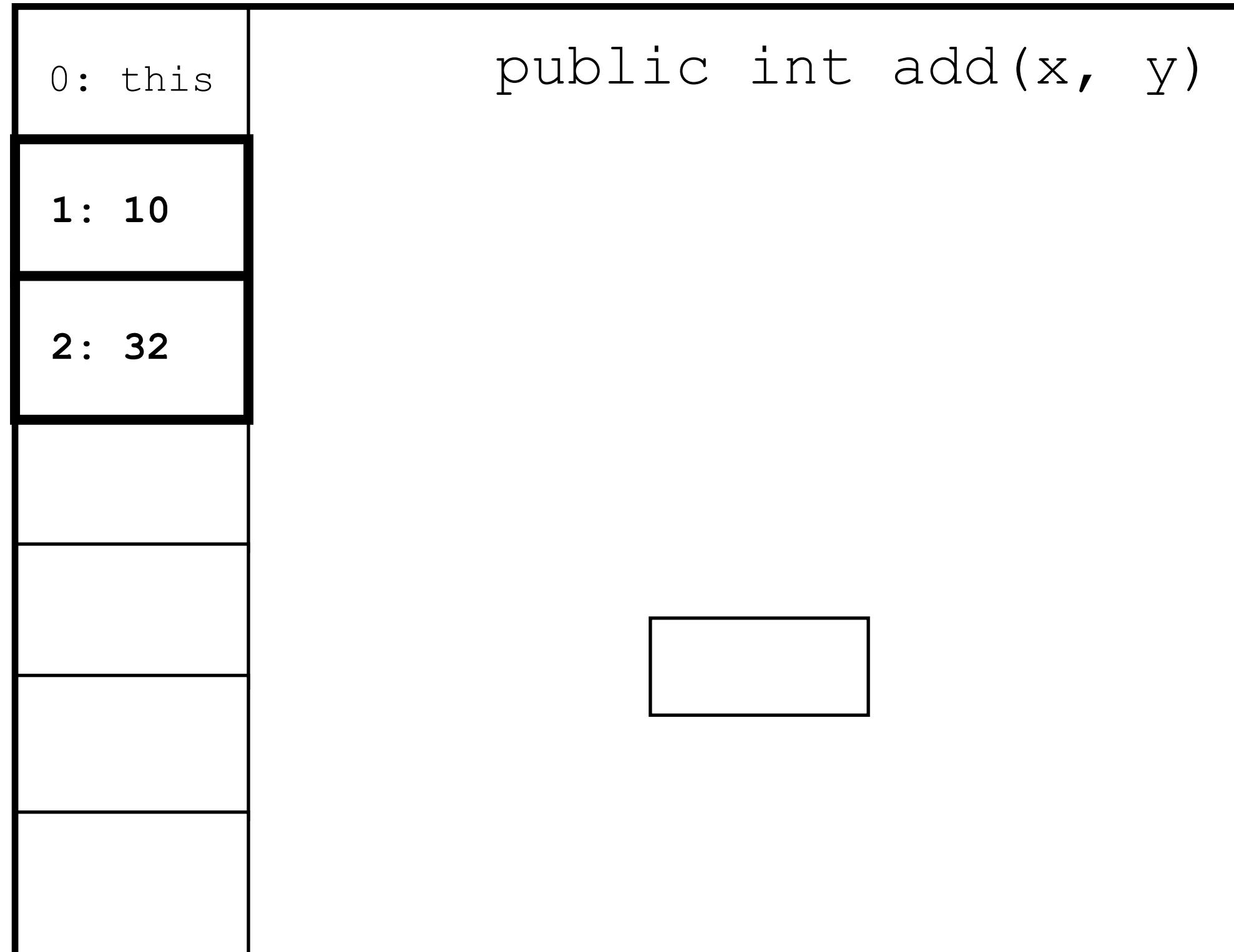
```
        4: return
```

public int add(int, int);

```
    Code:
```

```
    . . .
```

load, add & return



// Let's try add(10, 32);

public int add(int, int);

Code:

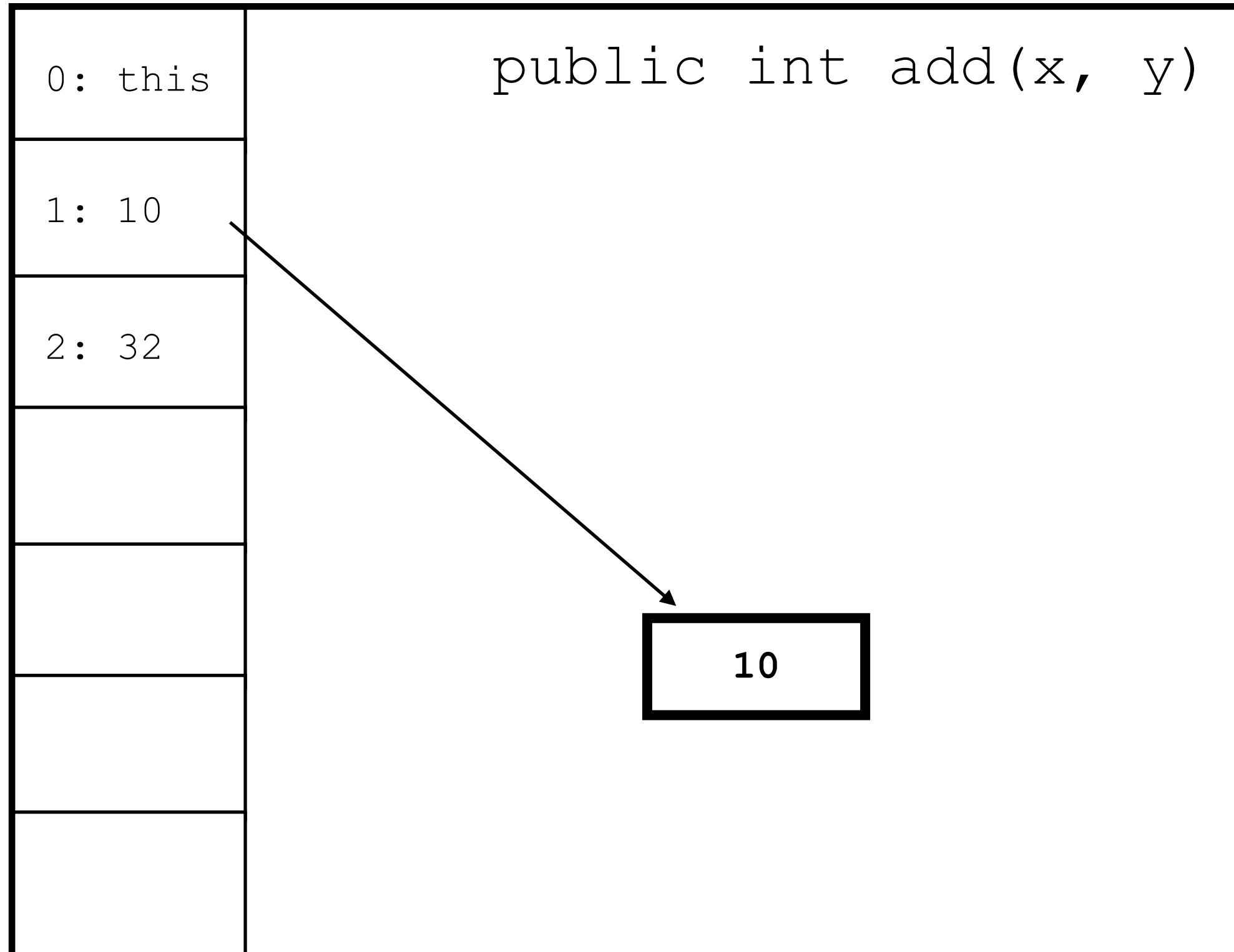
0: iload_1

1: iload_2

2: iadd

3: ireturn

load, add & return



// Let's try add(10, 32);

public int add(int, int);

Code:

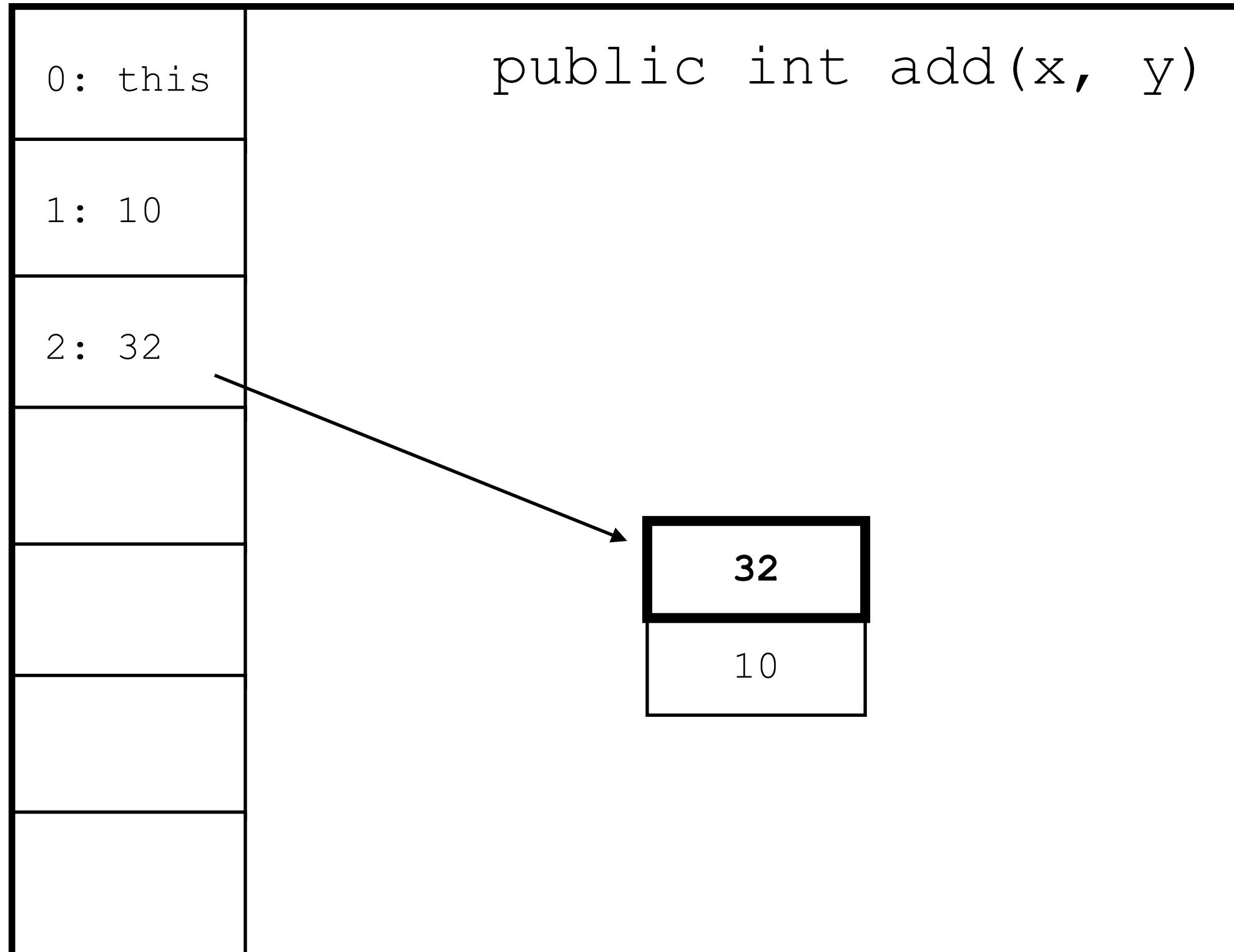
0: **iload_1**

1: **iload_2**

2: **iadd**

3: **ireturn**

load, add & return



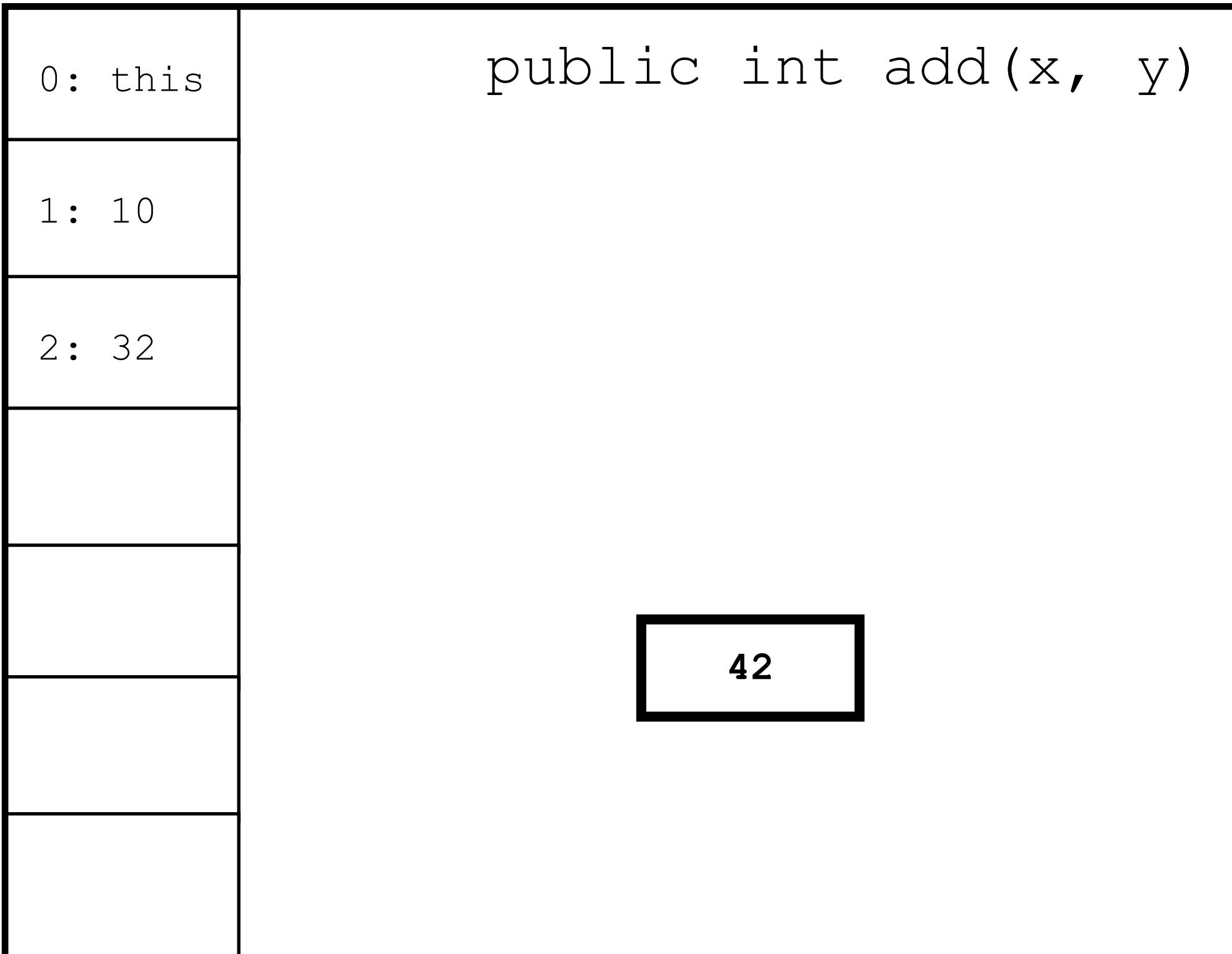
// Let's try add(10, 32);

public int add(int, int);

Code:

0: iload_1
1: **iload_2**
2: iadd
3: ireturn

load, add & return



// Let's try `add(10, 32);`

`public int add(int, int);`

Code:

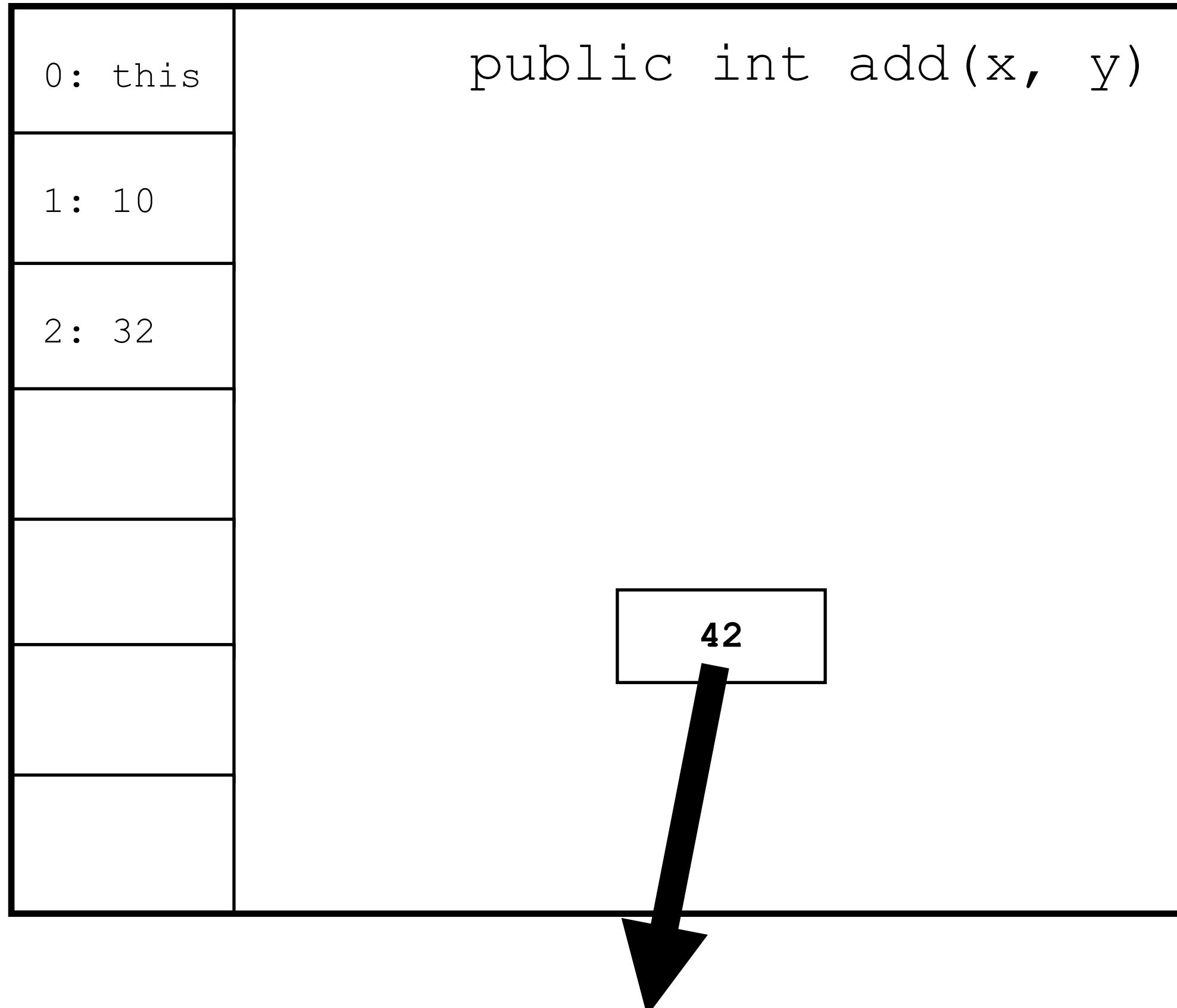
0: `iload_1`

1: `iload_2`

2: `iadd`

3: `ireturn`

load, add & return



```
// Let's try add(10, 32);  
  
public int add(int, int);  
Code:  
0: iload_1  
1: iload_2  
2: iadd  
3: ireturn
```

_store and ldc

```
public void stringingYouAlong() {  
    String s = "Hi";  
}
```

_store and ldc



```
public void stringingYouAlong();
```

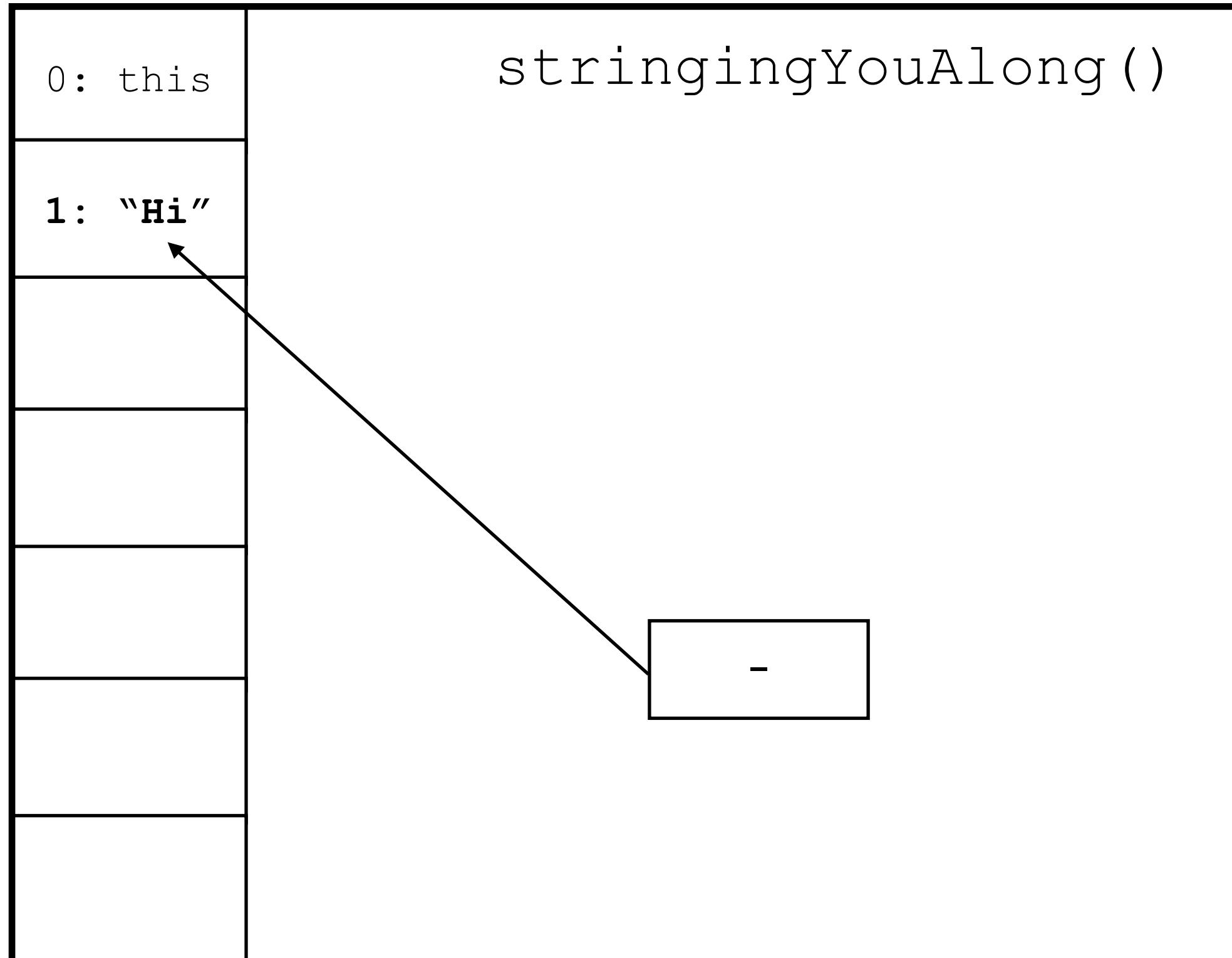
Code:

```
0: ldc #2 // String Hi  
2: astore_1  
3: return
```



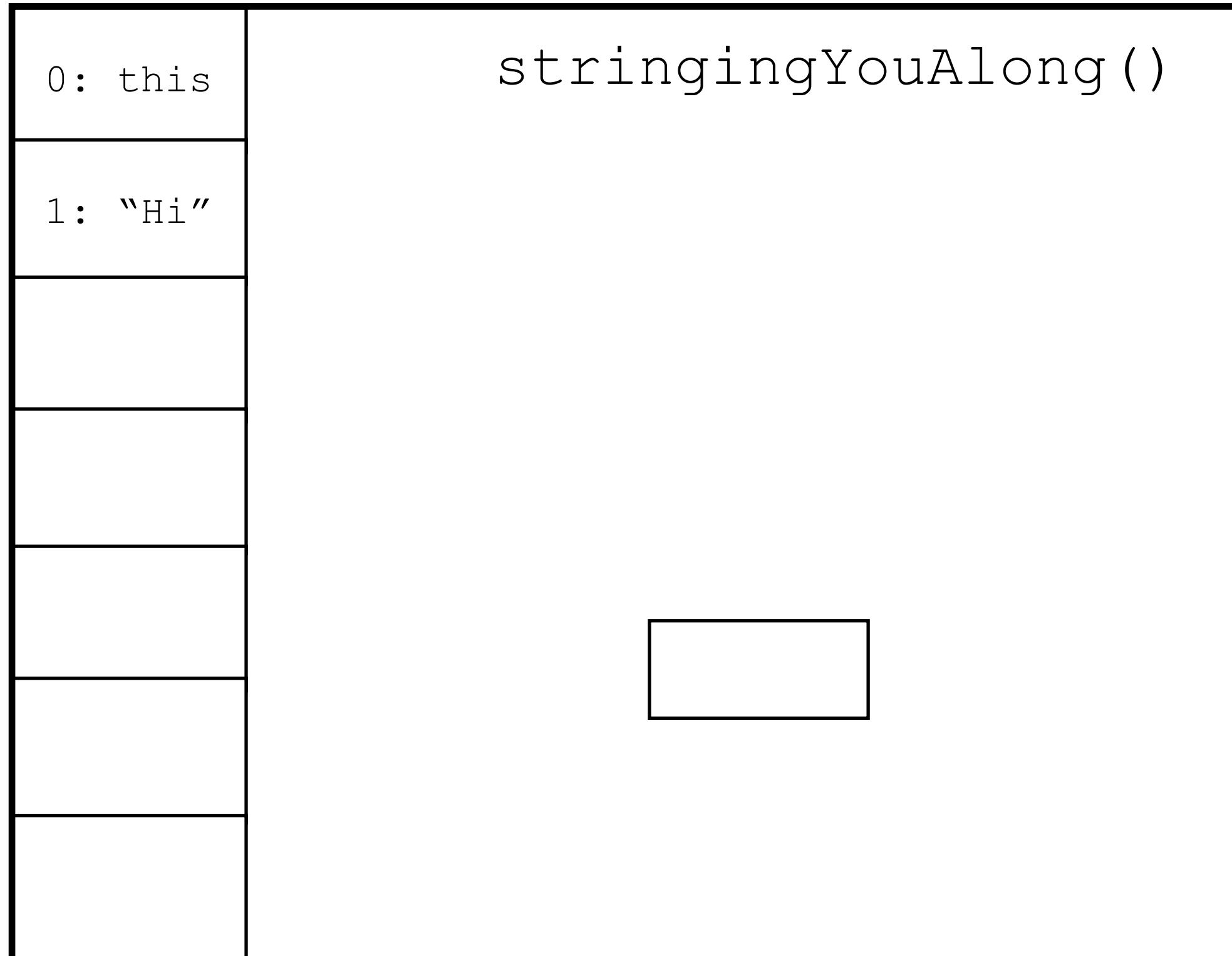
@jasonrclark

_store and ldc



```
public void stringingYouAlong();  
Code:  
0: ldc           #2 // String Hi  
2: astore_1  
3: return
```

_store and ldc



```
public void stringingYouAlong();  
Code:  
0: ldc           #2 // String Hi  
2: astore_1  
3: return
```

store and ldc

```
$ javap -c -v ...
```

Constant pool:

```
#1 = Methodref    #4.#12           // java/lang/Object."<init>": ()V
```

```
#2 = String      #13            // Hi
```

...

```
#13 = Utf8       Hi
```

new Object()

```
public void newObject() {  
    Object o = new Object();  
}
```

```
public void newObject();  
0: new           #2    // class java/lang/Object  
3: dup  
4: invokespecial #1   // Method java/lang/Object."<init>": ()V  
7: astore_1  
8: return
```

new Object()

```
public void newObject() {  
    Object o = new Object();  
}
```

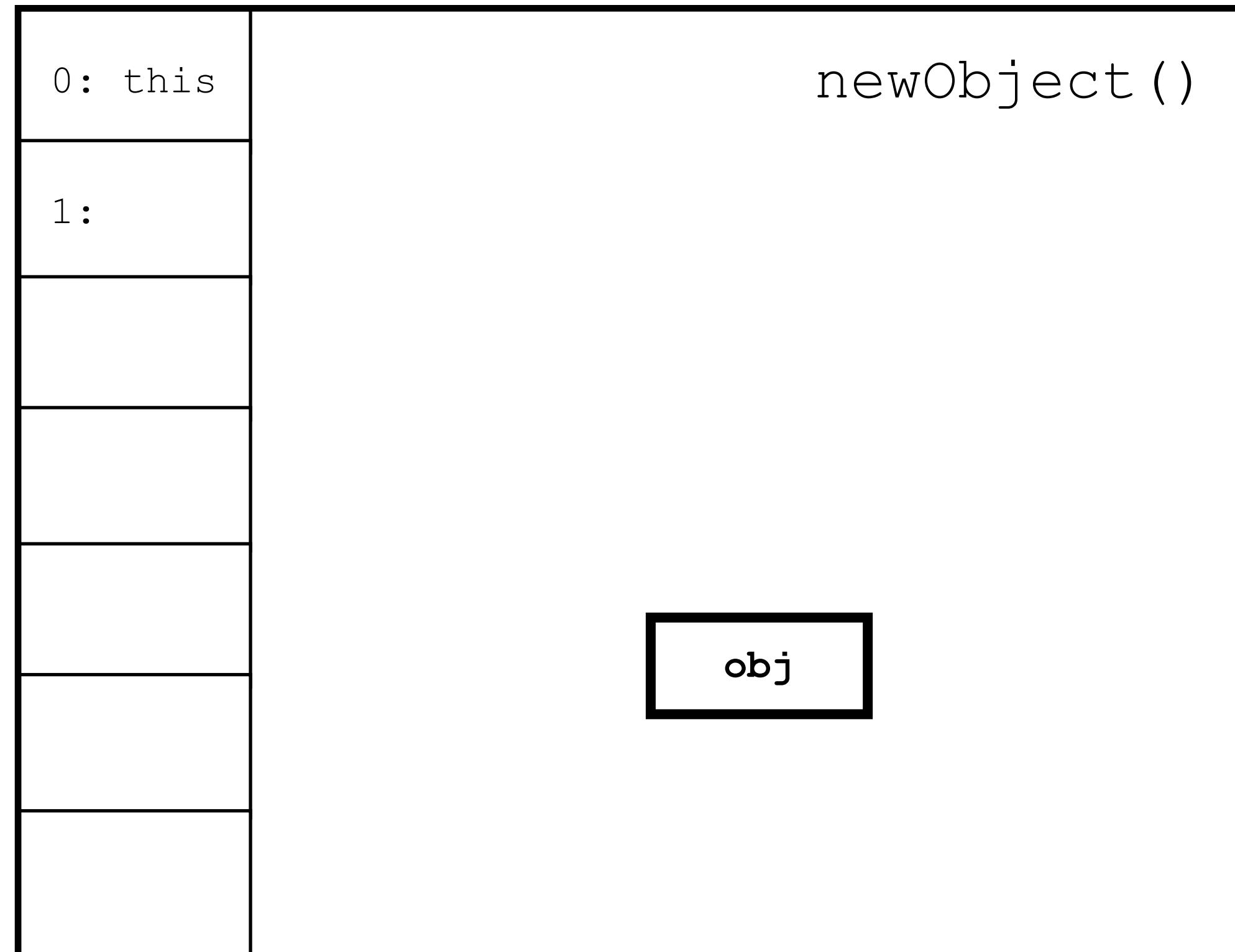
```
public void newObject();  
0: new           #2  // class java/lang/Object  
3: dup  
4: invokespecial #1  // Method java/lang/Object."<init>": ()V  
7: astore_1  
8: return
```

new Object()

```
public void newObject() {  
    Object o = new Object();  
}
```

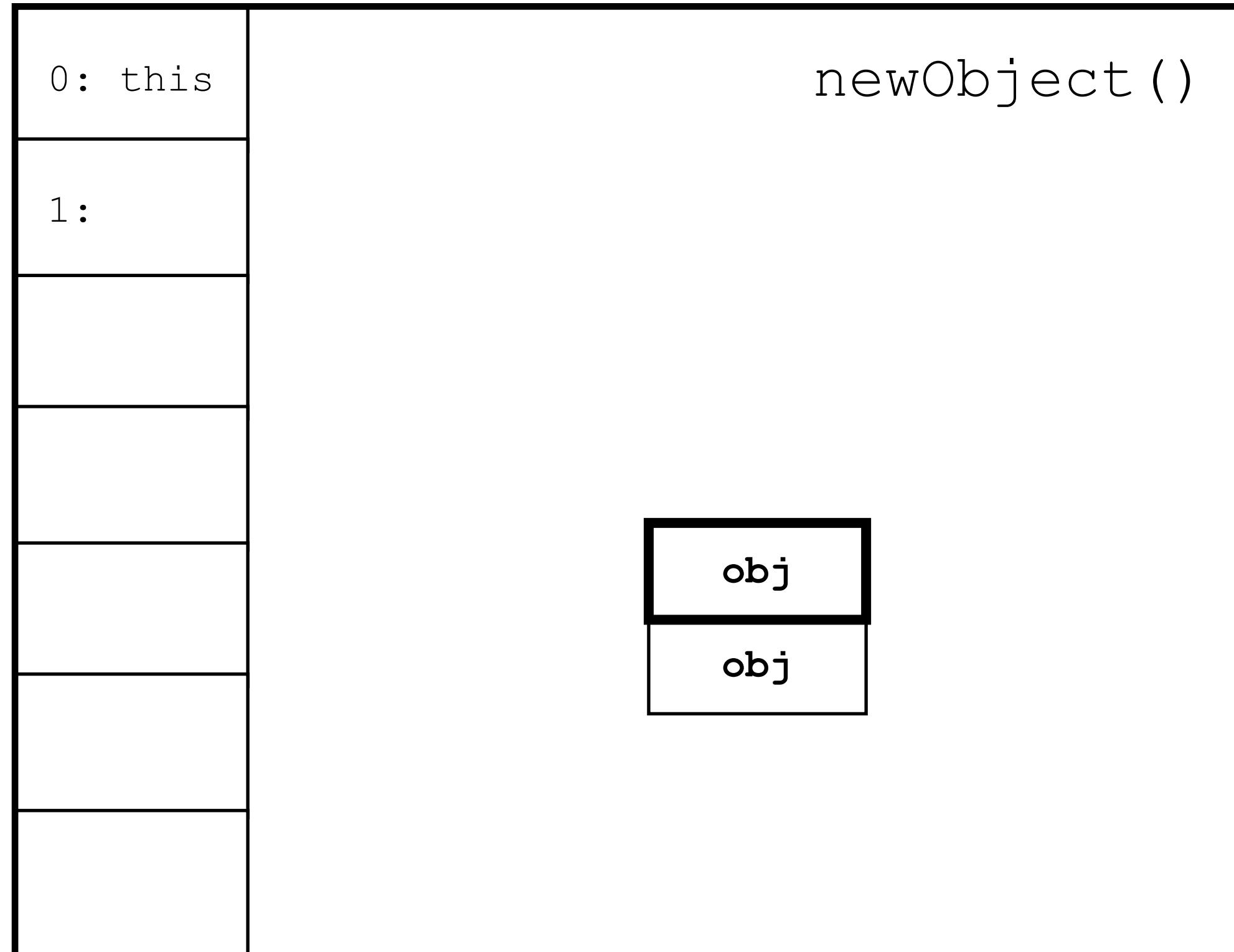
```
public void newObject();  
0: new           #2 // class java/lang/Object  
3: dup  
4: invokespecial #1 // Method java/lang/Object."<init>": ()V  
7: astore_1  
8: return
```

new Object()



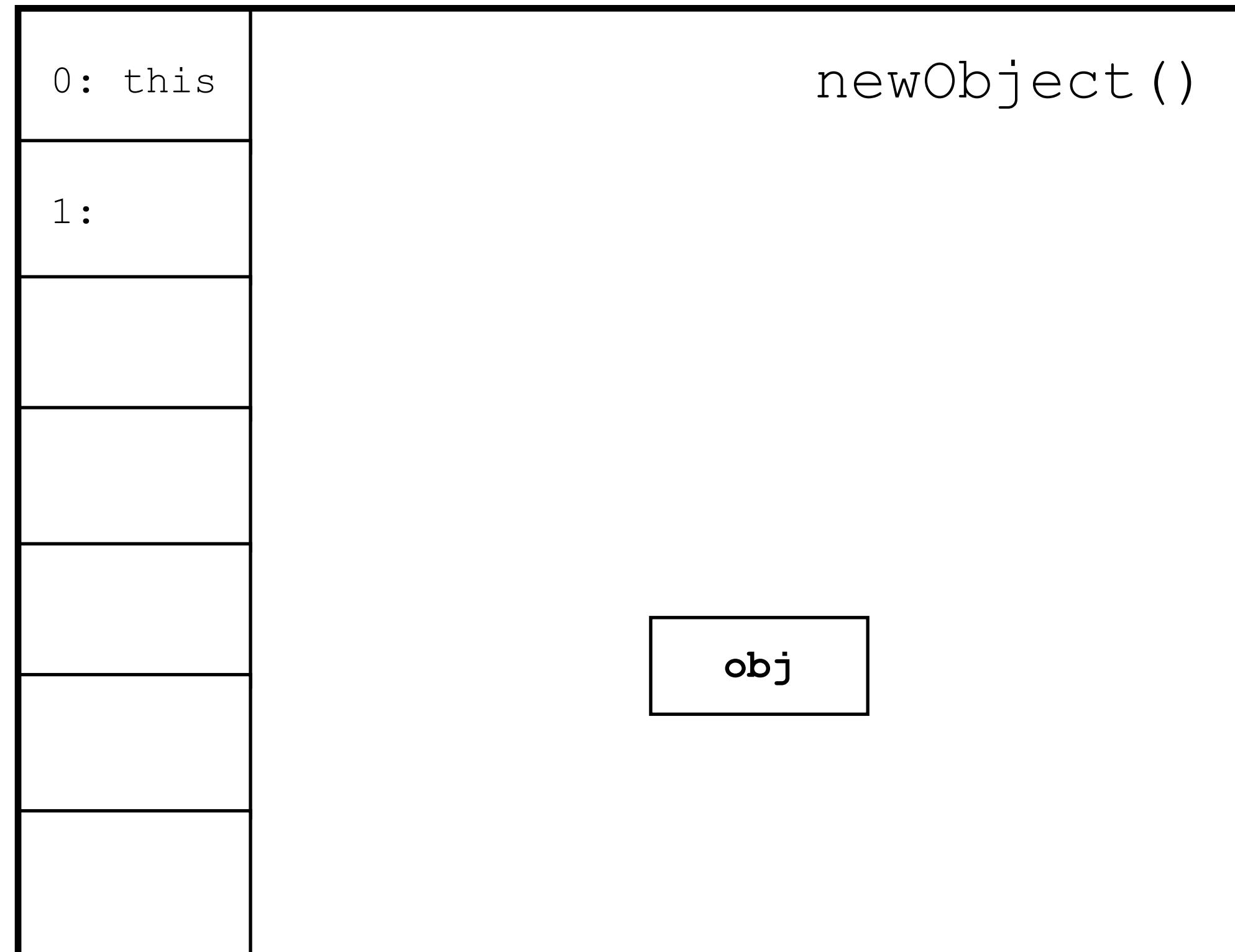
```
public void newObject();  
0: new #2 // Object  
3: dup  
4: invokespecial #1 // <init>  
7: astore_1  
8: return
```

new Object()



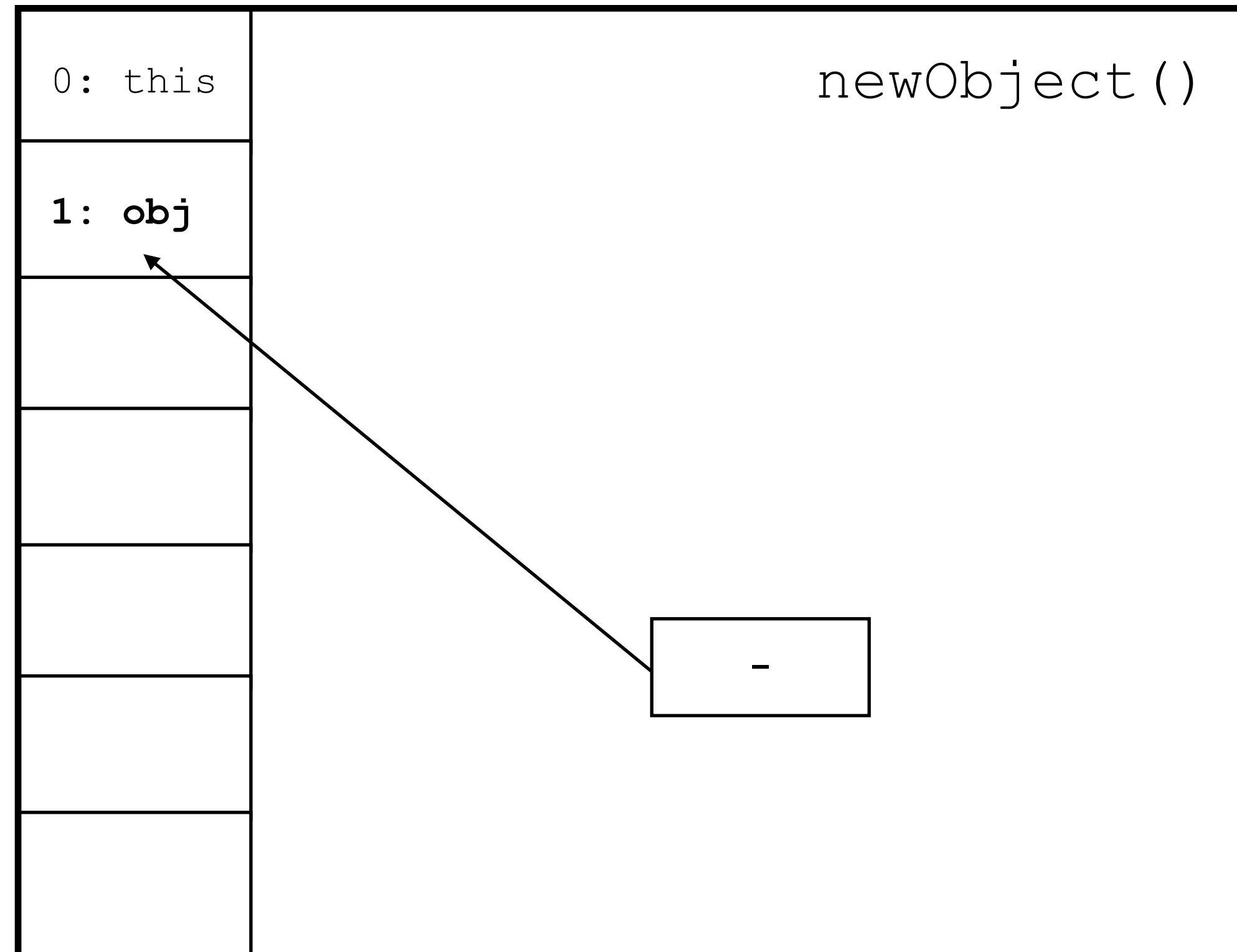
```
public void newObject();  
0: new           #2 // Object  
3: dup  
4: invokespecial #1 // <init>  
7: astore_1  
8: return
```

new Object()



```
public void newObject();  
0: new #2 // Object  
3: dup  
4: invokespecial #1 // <init>  
7: astore_1  
8: return
```

new Object()



```
public void newObject();  
0: new           #2 // Object  
3: dup  
4: invokespecial #1 // <init>  
7: astore_1  
8: return
```

Example: Lost Updates

Account.java

```
class Account {  
    private long balance;  
  
    public Account(long balance) {  
        this.balance = balance;  
    }  
  
    public void transfer(long transferAmount, Account other) {  
        if (balance >= transferAmount) {  
            balance -= transferAmount;  
            other.balance += transferAmount;  
        }  
    }  
}
```



Look like good banking code?
Please don't work at my bank

Concurrent transfers

```
var account = new Account(10000);
var otherAccount = new Account(0);
```

```
Runnable run = () -> {
    while (account.getBalance() > 0) {
        account.transfer(1, otherAccount);
    }
};
```

```
var threads = List.of(
    new Thread(run),
    new Thread(run)
);
```

Concurrent transfers

```
$ ./run
```

```
account: 0
```

```
otherAccount: 10108
```

Account.java

```
class Account {  
    private long balance;  
  
    public Account(long balance) {  
        this.balance = balance;  
    }  
  
    public void transfer(long transferAmount, Account other) {  
        if (balance >= transferAmount) {  
            balance -= transferAmount;  
            other.balance += transferAmount;  
        }  
    }  
}
```

Account

```
public void transfer(long, Account);  
    0: aload_0  
    1: getfield      #2           // Field balance:J  
    4: lload_1  
    5: lcmp  
    6: iflt         29  
    9: aload_0  
   10: dup  
   11: getfield      #2           // Field balance:J  
   14: lload_1  
   15: lsub  
   16: putfield      #2           // Field balance:J  
   19: aload_3  
   20: dup  
   21: getfield      #2           // Field balance:J  
   24: lload_1  
   25: ladd  
   26: putfield      #2           // Field balance:J  
   29: return
```

Account

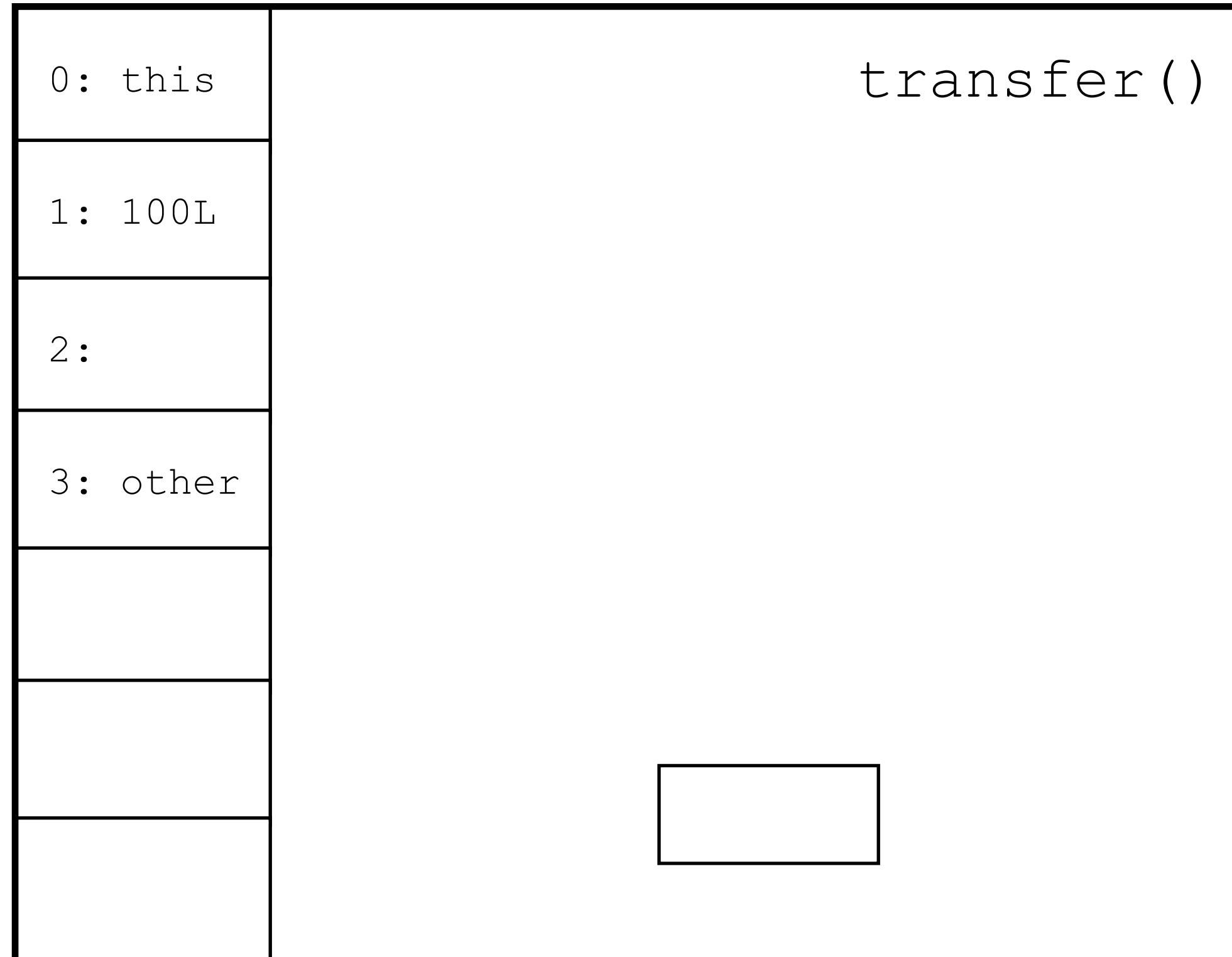
```
public void transfer(long, Account);  
    0: aload_0  
    1: getfield      #2           // Field balance:J  
    4: lload_1  
    5: lcmp  
    6: iflt       29  
    9: aload_0  
   10: dup  
   11: getfield      #2           // Field balance:J  
   14: lload_1  
   15: lsub  
   16: putfield      #2           // Field balance:J  
   19: aload_3  
   20: dup  
   21: getfield      #2           // Field balance:J  
   24: lload_1  
   25: ladd  
   26: putfield      #2           // Field balance:J  
   29: return
```



Account

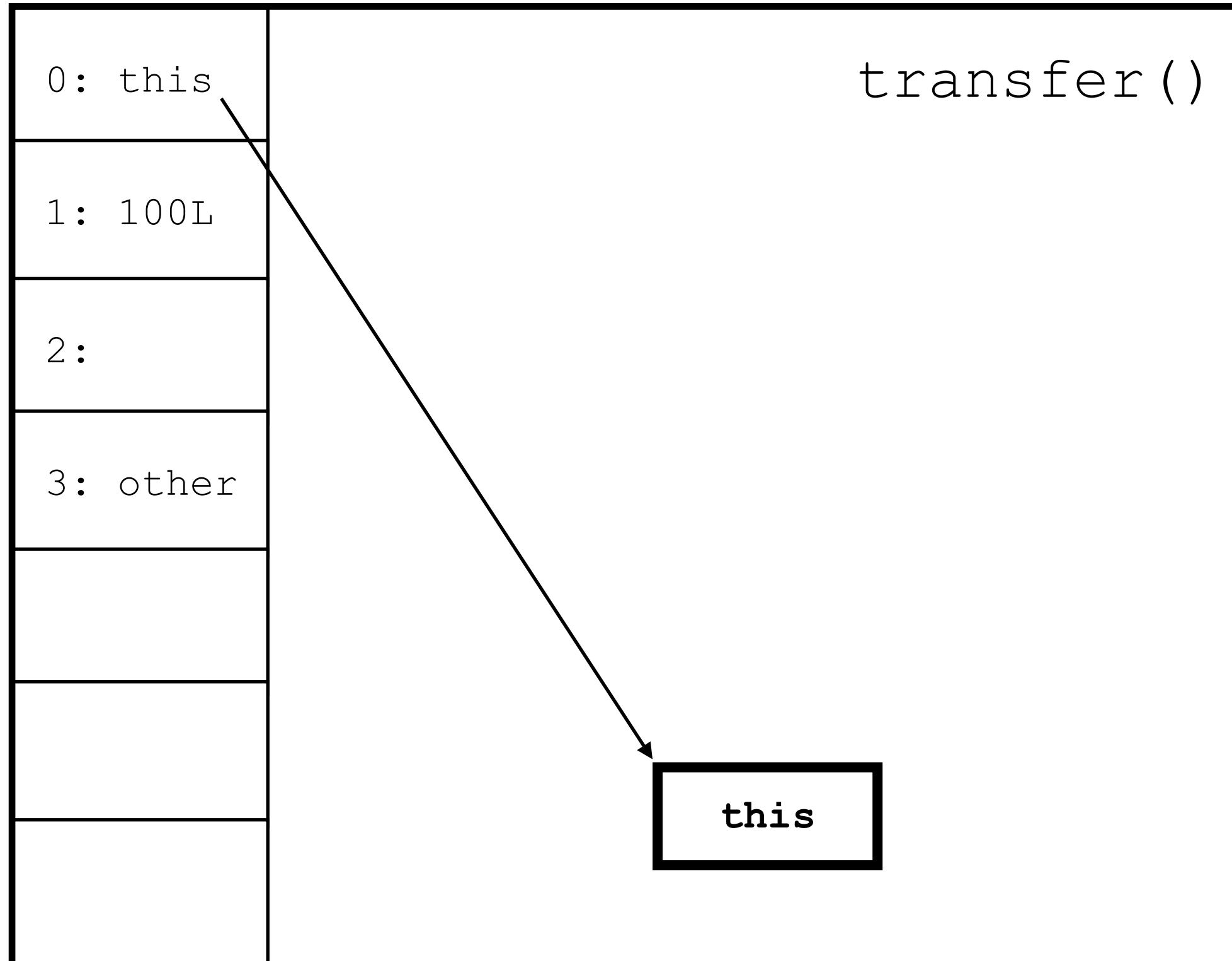
```
public void transfer(long, Account);  
    0: aload_0  
    1: getfield      #2                  // Field balance:J  
    4: lload_1  
5: lcmp  
6: iflt          29  
    9: aload_0  
   10: dup  
   11: getfield      #2                  // Field balance:J  
   14: lload_1  
   15: lsub  
   16: putfield      #2                  // Field balance:J  
   19: aload_3  
   20: dup  
   21: getfield      #2                  // Field balance:J  
   24: lload_1  
   25: ladd  
   26: putfield      #2                  // Field balance:J  
29: return
```

Account



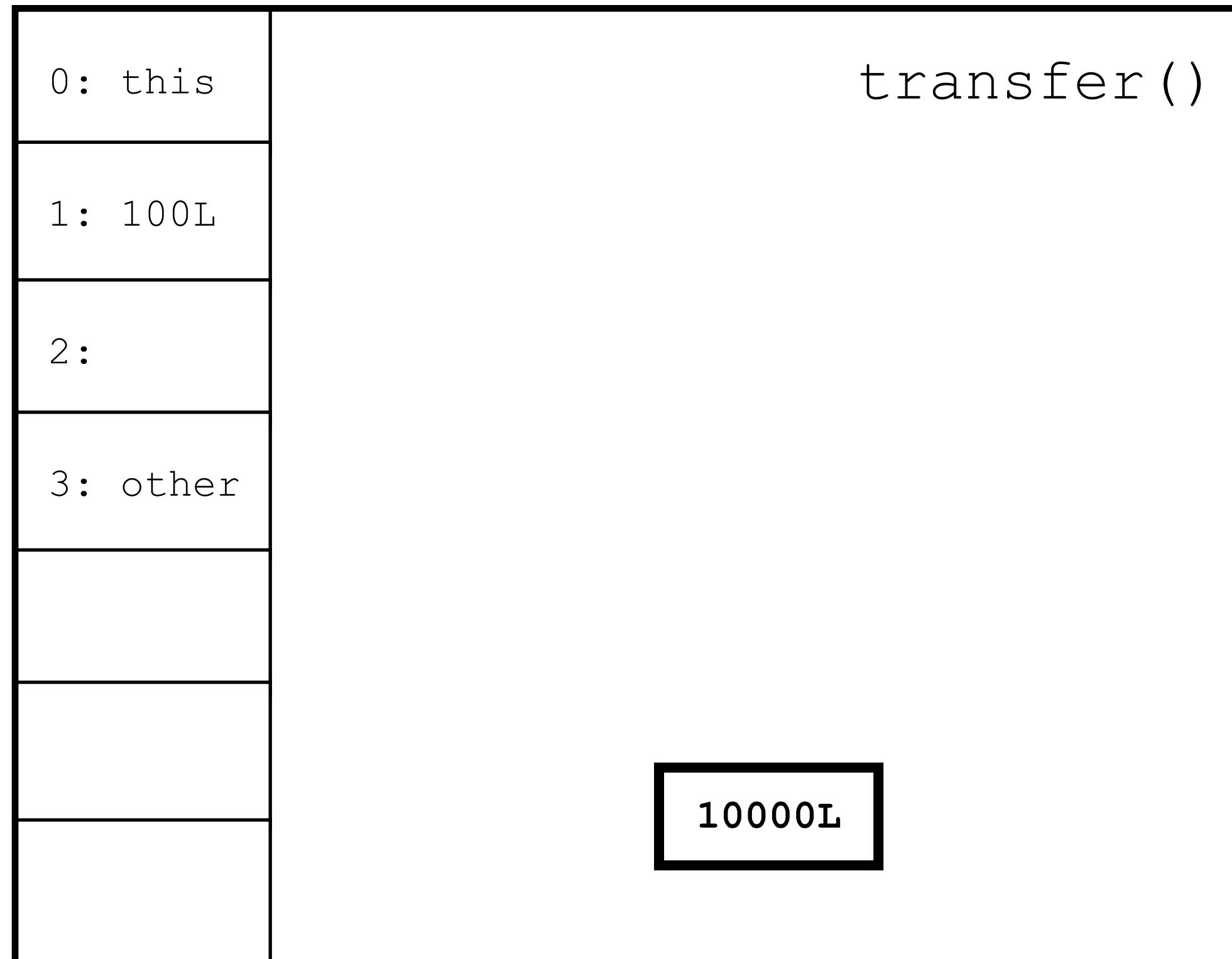
```
public void transfer(long, Account);  
          0: aload_0  
          1: getfield      #2 // Field  
          4: lload_1  
          5: lcmp  
          6: iflt         29  
          9: aload_0  
         10: dup  
         11: getfield      #2 // Field  
         14: lload_1  
         15: lsub  
         16: putfield      #2 // Field  
         19: aload_3  
         20: dup  
         21: getfield      #2 // Field  
         24: lload_1  
         25: ladd  
         26: putfield      #2 // Field  
         29: return
```

Account



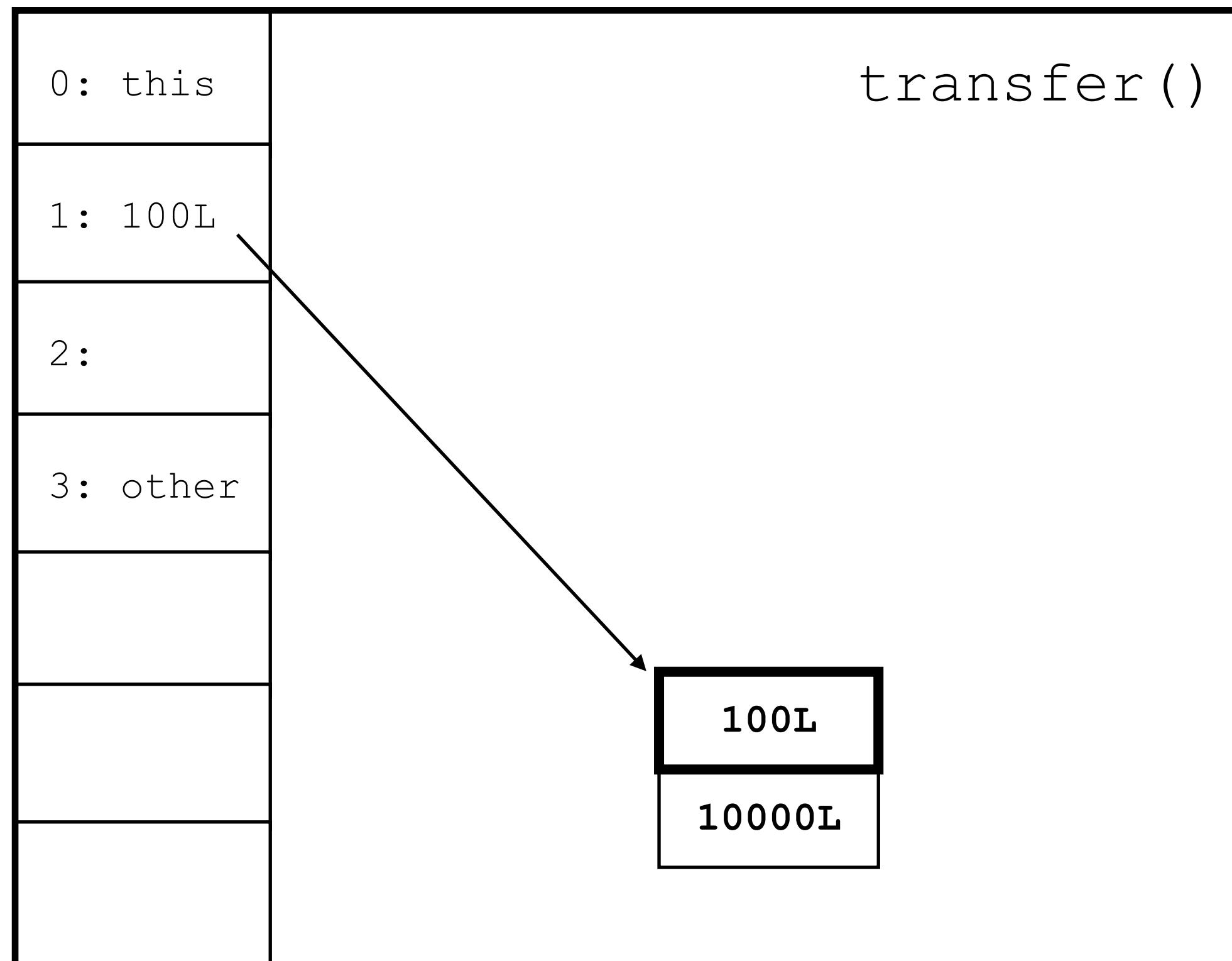
```
public void transfer(long, Account);  
    0: aload_0                                #2 // Field  
    1: getfield      #2 // Field  
    4: lload_1  
    5: lcmp  
    6: iflt         29  
    9: aload_0  
   10: dup  
   11: getfield      #2 // Field  
   14: lload_1  
   15: lsub  
   16: putfield      #2 // Field  
   19: aload_3  
   20: dup  
   21: getfield      #2 // Field  
   24: lload_1  
   25: ladd  
   26: putfield      #2 // Field  
   29: return
```

Account



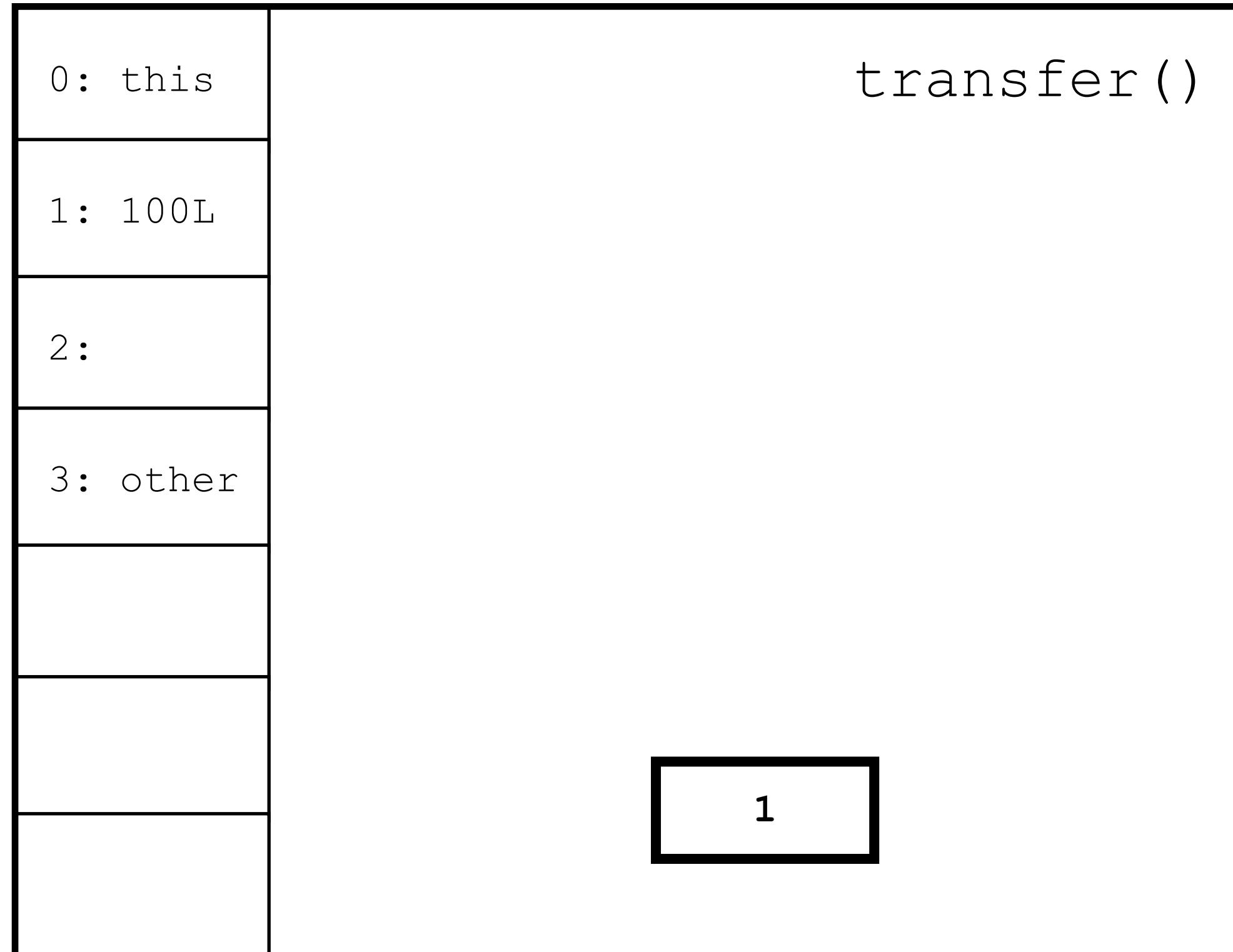
```
public void transfer(long, Account);  
    0: aload_0  
    1: getfield           #2 // Field  
    4: lload_1  
    5: lcmp  
    6: iflt             29  
    9: aload_0  
   10: dup  
   11: getfield         #2 // Field  
   14: lload_1  
   15: lsub  
   16: putfield         #2 // Field  
   19: aload_3  
   20: dup  
   21: getfield         #2 // Field  
   24: lload_1  
   25: ladd  
   26: putfield         #2 // Field  
   29: return
```

Account



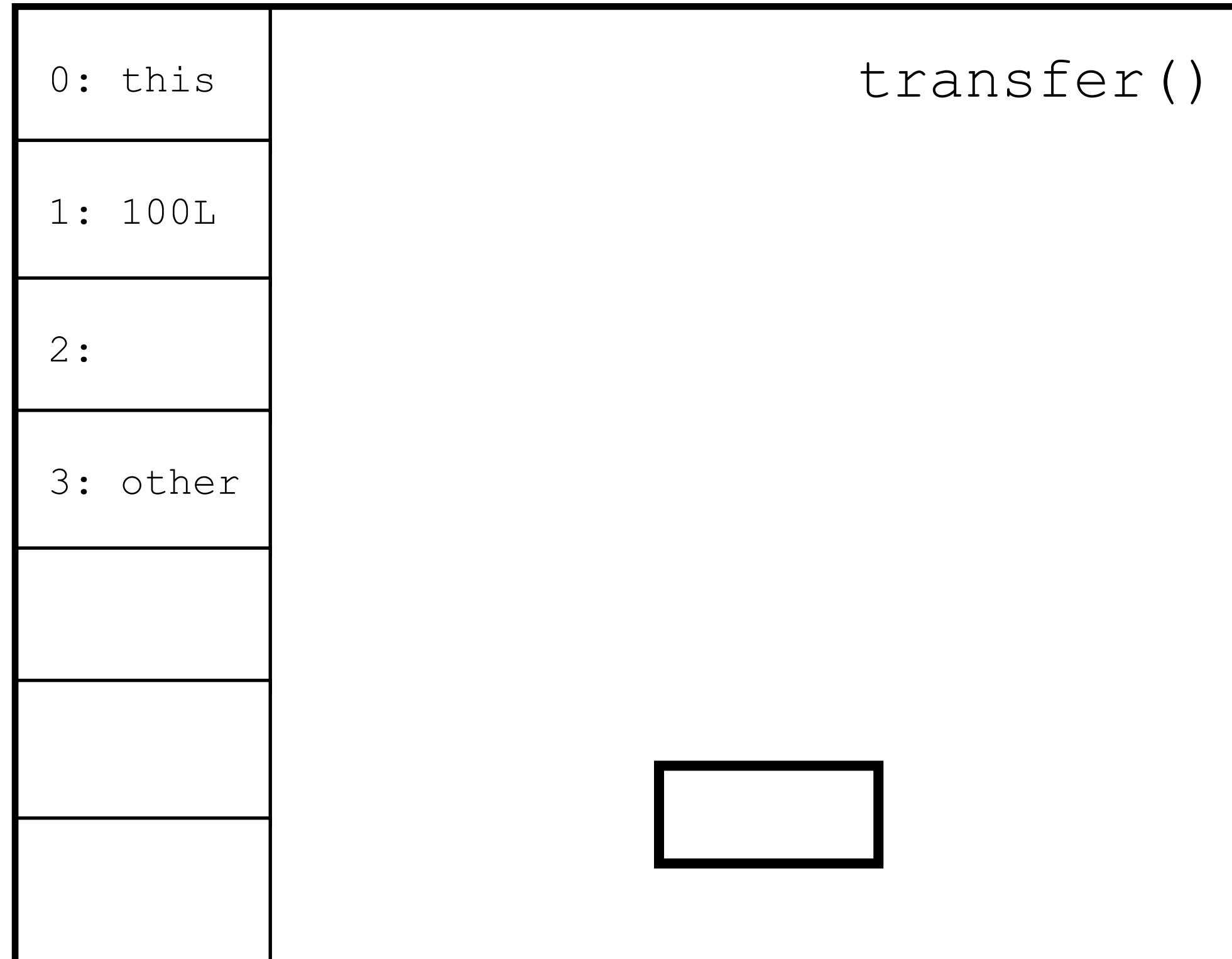
```
public void transfer(long, Account);  
    0: aload_0  
    1:  getfield           #2 // Field  
    4:  lload_1  
    5:  lcmp  
    6:  iflt  
    9:  aload_0  
   10: dup  
   11: getfield           #2 // Field  
   14: lload_1  
   15: lsub  
   16: putfield           #2 // Field  
   19: aload_3  
   20: dup  
   21: getfield           #2 // Field  
   24: lload_1  
   25: ladd  
   26: putfield           #2 // Field  
   29: return
```

Account



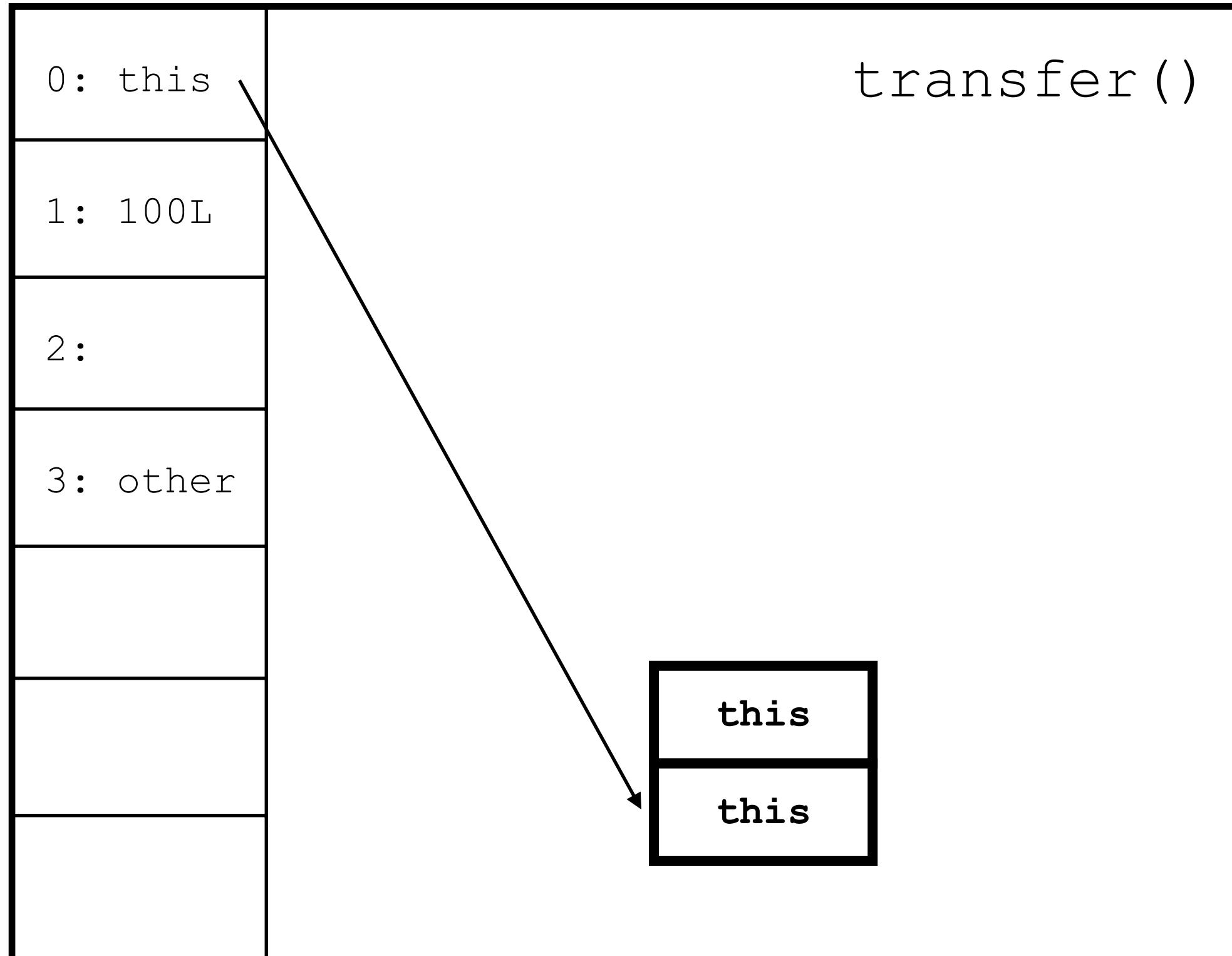
```
public void transfer(long, Account);  
    0: aload_0  
    1: getfield           #2 // Field  
    4: lload_1  
    5: lcmp  
    6: iflt               29  
    9: aload_0  
   10: dup  
   11: getfield           #2 // Field  
   14: lload_1  
   15: lsub  
   16: putfield           #2 // Field  
   19: aload_3  
   20: dup  
   21: getfield           #2 // Field  
   24: lload_1  
   25: ladd  
   26: putfield           #2 // Field  
   29: return
```

Account



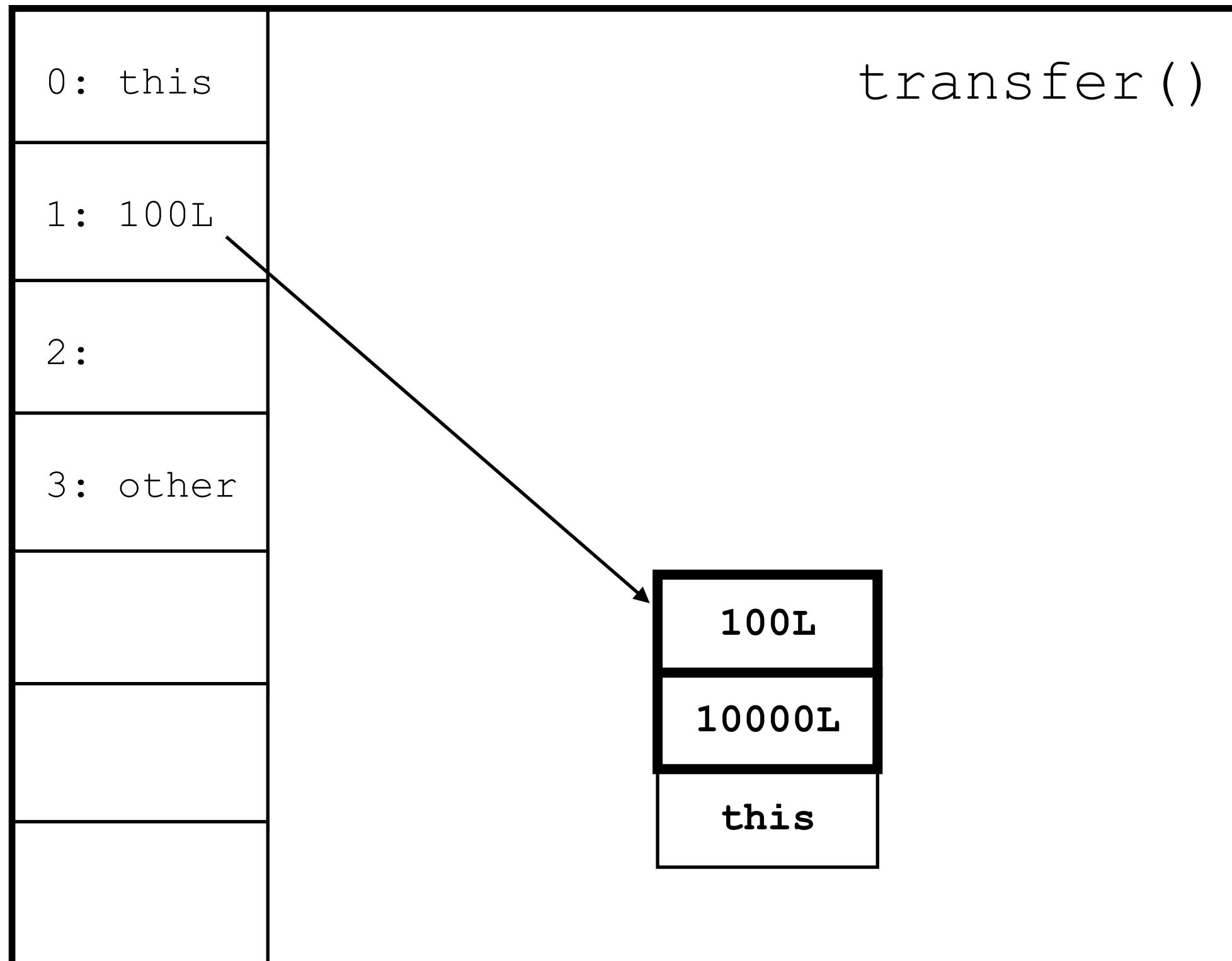
```
public void transfer(long, Account);  
    0: aload_0  
    1: getfield           #2 // Field  
    4: lload_1  
    5: lcmp  
    6: iflt               29  
    9: aload_0  
   10: dup  
   11: getfield           #2 // Field  
   14: lload_1  
   15: lsub  
   16: putfield           #2 // Field  
   19: aload_3  
   20: dup  
   21: getfield           #2 // Field  
   24: lload_1  
   25: ladd  
   26: putfield           #2 // Field  
29: return
```

Account



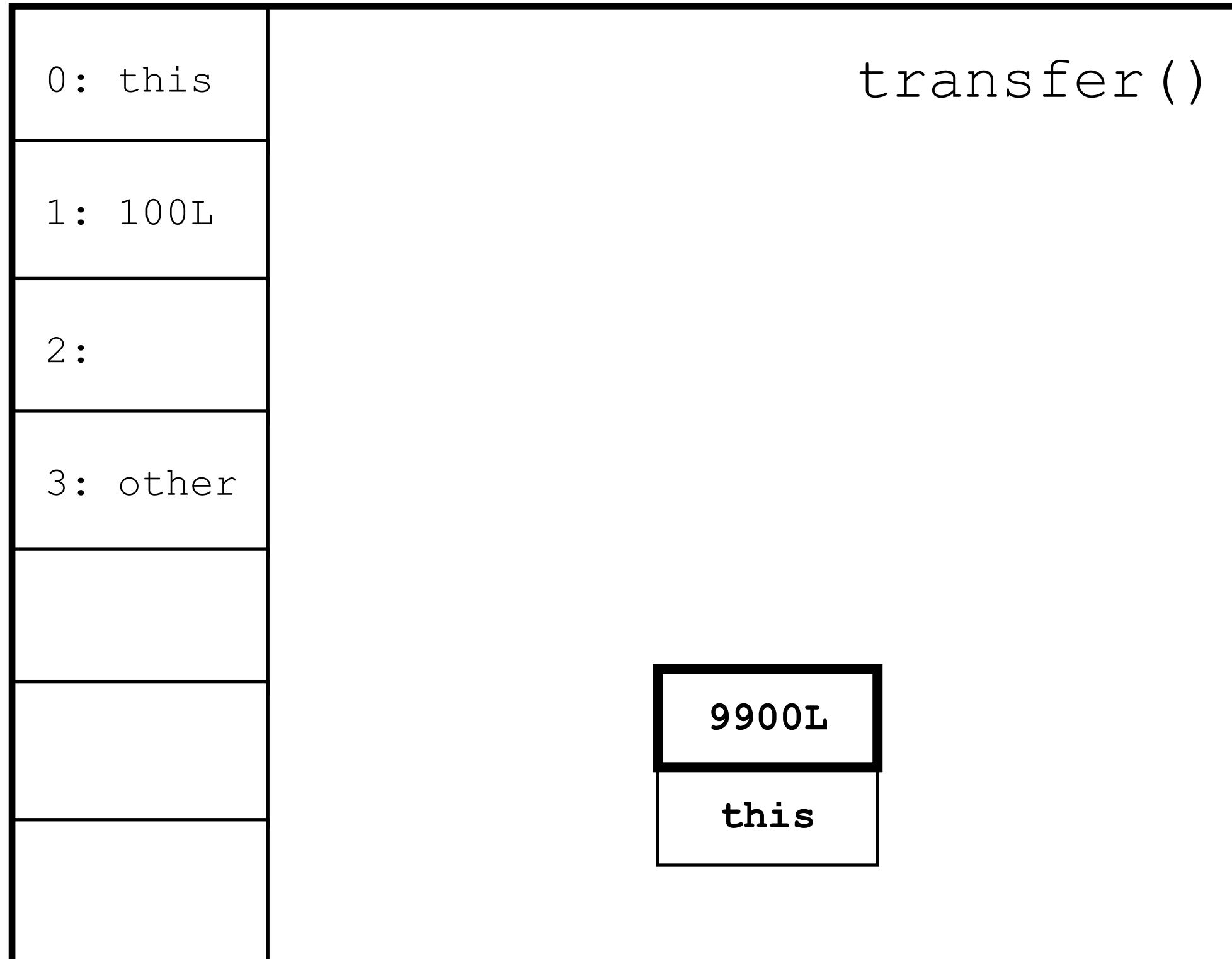
```
public void transfer(long, Account);  
    0: aload_0  
    1: getfield      #2 // Field  
    4: lload_1  
    5: lcmp  
    6: iflt  
    9: aload_0  
   10: dup  
   11: getfield      #2 // Field  
   14: lload_1  
   15: lsub  
   16: putfield      #2 // Field  
   19: aload_3  
   20: dup  
   21: getfield      #2 // Field  
   24: lload_1  
   25: ladd  
   26: putfield      #2 // Field  
   29: return
```

Account



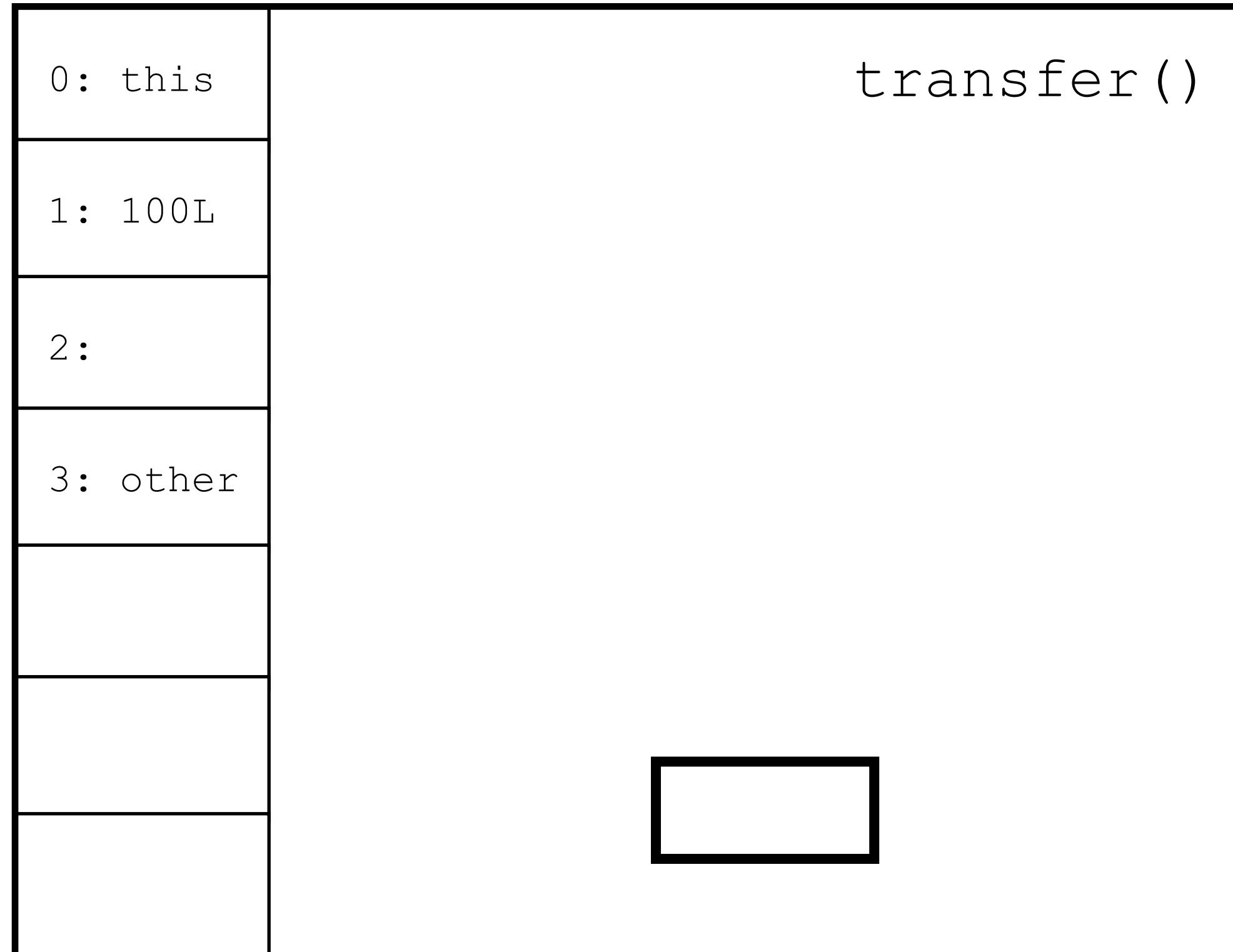
```
public void transfer(long, Account);  
    0: aload_0  
    1: getfield           #2 // Field  
    4: lload_1  
    5: lcmp  
    6: iflt  
    9: aload_0  
   10: dup  
   11: getfield           #2 // Field  
   14: lload_1  
   15: lsub  
   16: putfield          #2 // Field  
   19: aload_3  
   20: dup  
   21: getfield           #2 // Field  
   24: lload_1  
   25: ladd  
   26: putfield          #2 // Field  
   29: return
```

Account



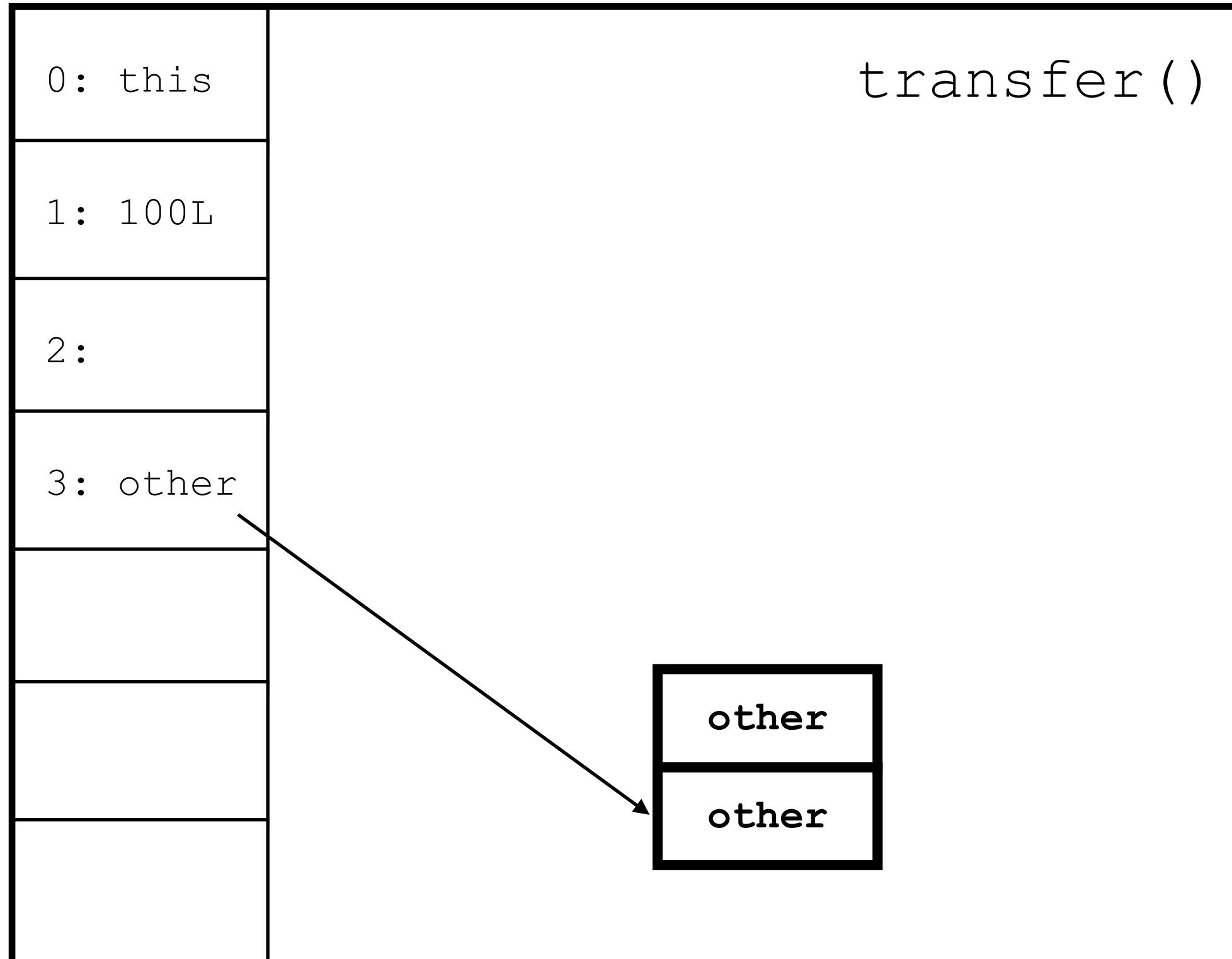
```
public void transfer(long, Account);  
    0:  aload_0  
    1:  getfield           #2 // Field  
    4:  lload_1  
    5:  lcmp  
    6:  iflt               29  
    9:  aload_0  
   10: dup  
   11: getfield           #2 // Field  
   14: lload_1  
  15: lsub  
   16: putfield           #2 // Field  
   19:  aload_3  
   20: dup  
   21: getfield           #2 // Field  
   24: lload_1  
   25: ladd  
   26: putfield           #2 // Field  
   29: return
```

Account



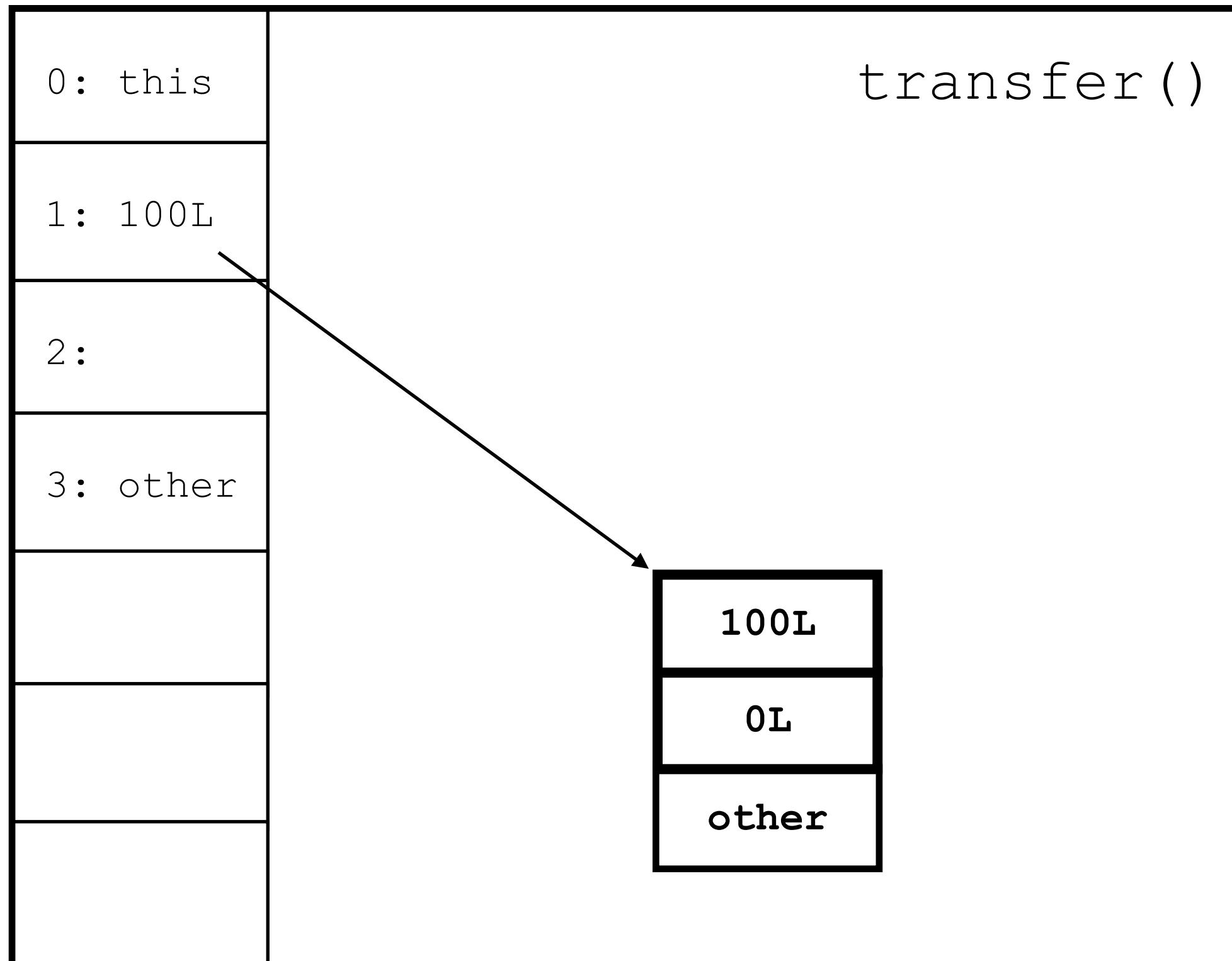
```
public void transfer(long, Account);  
    0: aload_0  
    1: getfield      #2 // Field  
    4: lload_1  
    5: lcmp  
    6: iflt         29  
    9: aload_0  
   10: dup  
   11: getfield      #2 // Field  
   14: lload_1  
   15: lsub  
16: putfield      #2 // Field  
   19: aload_3  
   20: dup  
   21: getfield      #2 // Field  
   24: lload_1  
   25: ladd  
   26: putfield      #2 // Field  
   29: return
```

Account



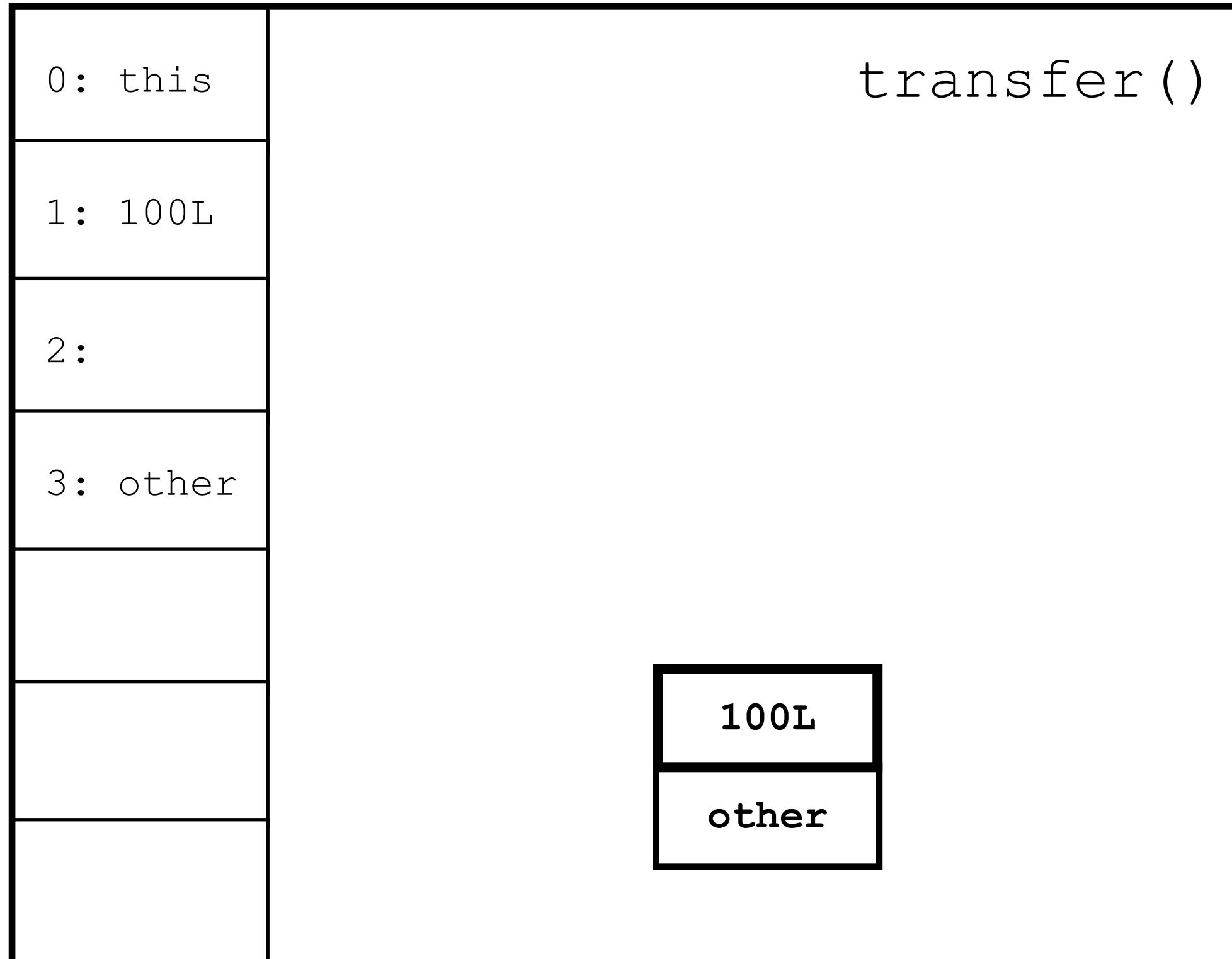
```
public void transfer(long, Account);  
    0: aload_0  
    1: getfield           #2 // Field  
    4: lload_1  
    5: lcmp  
    6: iflt               29  
    9: aload_0  
   10: dup  
   11: getfield           #2 // Field  
   14: lload_1  
   15: lsub  
   16: putfield           #2 // Field  
   19: aload_3  
   20: dup  
   21: getfield           #2 // Field  
   24: lload_1  
   25: ladd  
   26: putfield           #2 // Field  
   29: return
```

Account



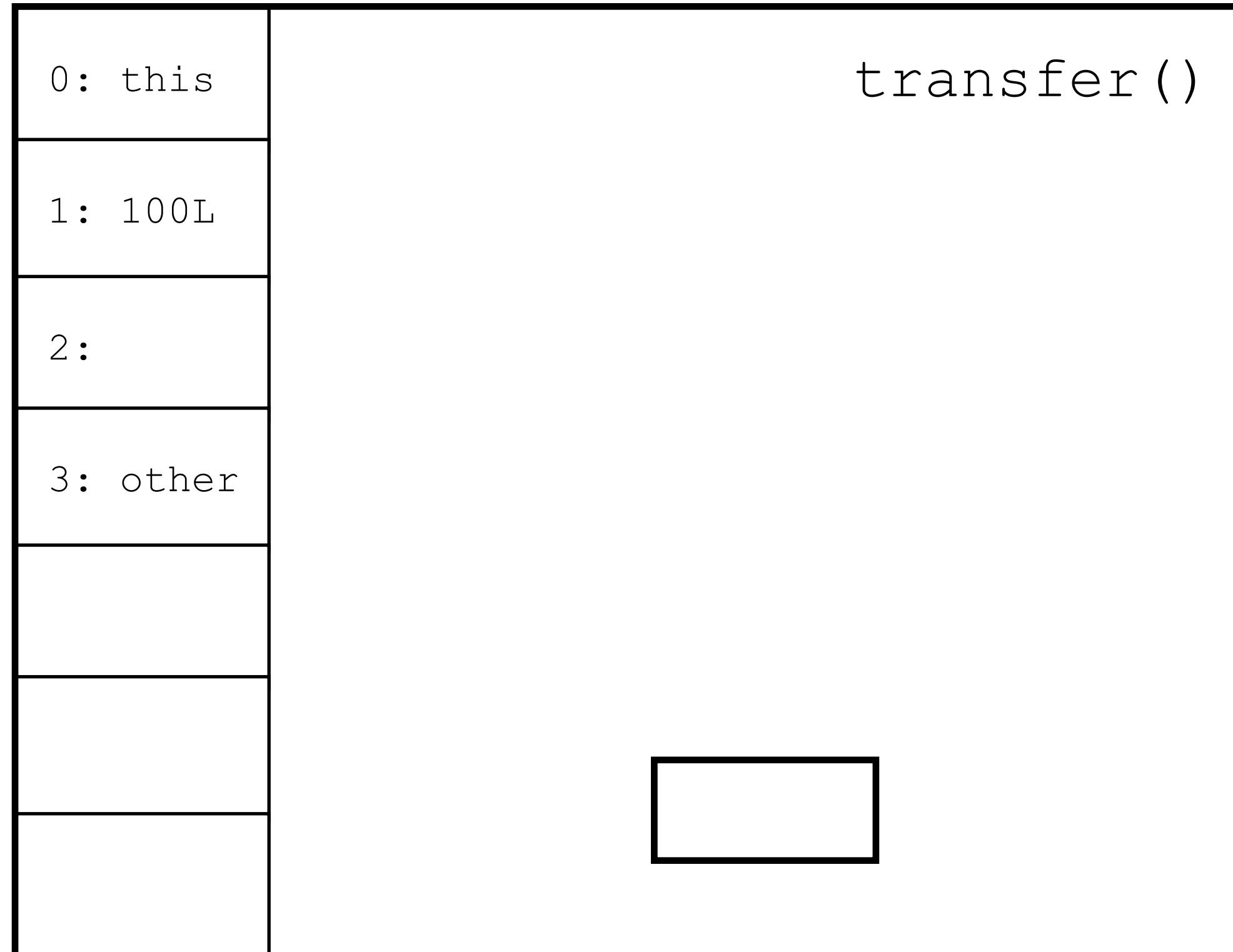
```
public void transfer(long, Account);  
    0:  aload_0  
    1:  getfield           #2 // Field  
    4:  lload_1  
    5:  lcmp  
    6:  iflt  
    9:  aload_0  
   10: dup  
   11: getfield           #2 // Field  
   14: lload_1  
   15: lsub  
   16: putfield           #2 // Field  
   19:  aload_3  
   20: dup  
   21: getfield           #2 // Field  
   24: lload_1  
   25: ladd  
   26: putfield           #2 // Field  
   29: return
```

Account



```
public void transfer(long, Account);  
    0: aload_0  
    1: getfield           #2 // Field  
    4: lload_1  
    5: lcmp  
    6: iflt               29  
    9: aload_0  
   10: dup  
   11: getfield           #2 // Field  
   14: lload_1  
   15: lsub  
   16: putfield           #2 // Field  
   19: aload_3  
   20: dup  
   21: getfield           #2 // Field  
   24: lload_1  
  25: ladd  
   26: putfield           #2 // Field  
   29: return
```

Account



```
public void transfer(long, Account);  
    0: aload_0  
    1: getfield      #2 // Field  
    4: lload_1  
    5: lcmp  
    6: iflt         29  
    9: aload_0  
   10: dup  
   11: getfield      #2 // Field  
   14: lload_1  
   15: lsub  
   16: putfield      #2 // Field  
   19: aload_3  
   20: dup  
   21: getfield      #2 // Field  
   24: lload_1  
   25: ladd  
26: putfield      #2 // Field  
29: return
```

Account

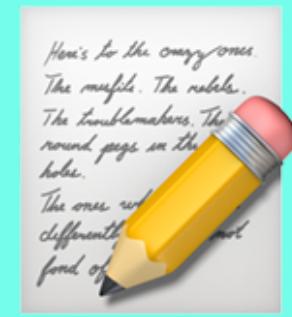
```
public void transfer(long, Account);  
    0: aload_0  
    1: getfield      #2 // Field  
    4: lload_1  
    5: lcmp  
    6: iflt           29  
    9: aload_0  
   10: dup  
   11: getfield      #2 // Field  
   14: lload_1  
   15: lsub  
   16: putfield      #2 // Field  
   19: aload_3  
   20: dup  
   21: getfield      #2 // Field  
   24: lload_1  
   25: ladd  
   26: putfield      #2 // Field  
   29: return
```

```
public void transfer(long, Account);  
    0: aload_0  
    1: getfield      #2 // Field  
    4: lload_1  
    5: lcmp  
    6: iflt           29  
    9: aload_0  
   10: dup  
   11: getfield      #2 // Field  
   14: lload_1  
   15: lsub  
   16: putfield      #2 // Field  
   19: aload_3  
   20: dup  
   21: getfield      #2 // Field  
   24: lload_1  
   25: ladd  
   26: putfield      #2 // Field  
   29: return
```

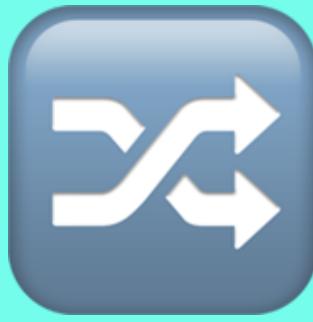
Account

```
public void transfer(long, Account);  
    0: aload_0  
    1: getfield      #2 // Field  
    4: lload_1  
    5: lcmp  
    6: iflt           29  
    9: aload_0  
   10: dup  
   11: getfield      #2 // Field  
   14: lload_1  
   15: lsub  
   16: putfield      #2 // Field  
   19: aload_3  
   20: dup  
   21: getfield      #2 // Field  
   24: lload_1  
   25: ladd  
   26: putfield      #2 // Field  
   29: return
```

```
public void transfer(long, Account);  
    0: aload_0  
    1: getfield      #2 // Field  
    4: lload_1  
    5: lcmp  
    6: iflt           29  
    9: aload_0  
   10: dup  
   11: getfield      #2 // Field  
   14: lload_1  
   15: lsub  
   16: putfield      #2 // Field  
   19: aload_3  
   20: dup  
   21: getfield      #2 // Field  
   24: lload_1  
   25: ladd  
   26: putfield      #2 // Field  
   29: return
```



Our evaluation stack holds intermediate values



Bytecodes on different threads interleave

Example: Tail Recursion

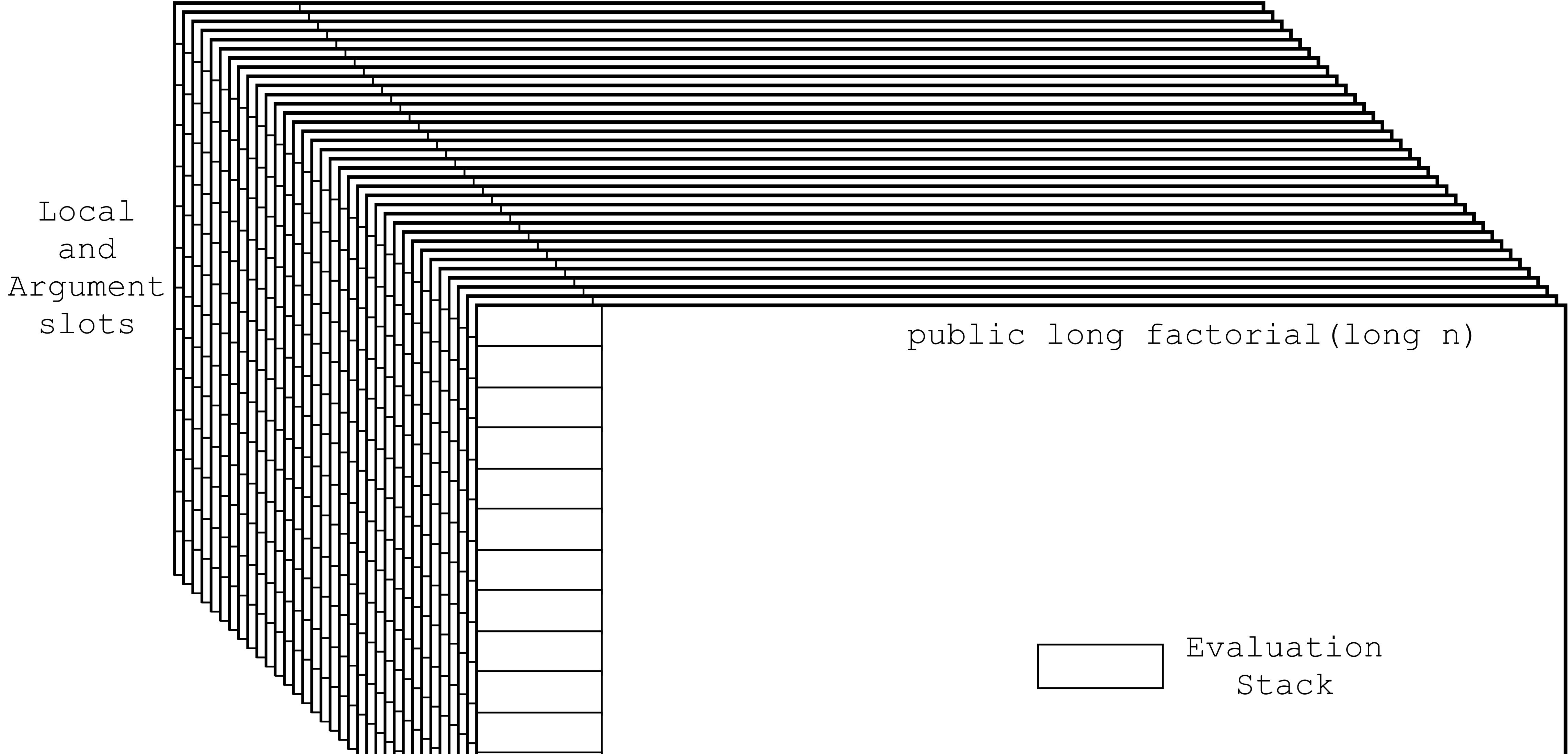
Recursion

```
public long factorial(long n) {  
    if (n <= 0) {  
        return 1;  
    }  
  
    return n * factorial(n - 1);  
}
```

Recursion

```
Exception in thread "main" java.lang.StackOverflowError  
at Overflowing.overflow(Overflowing.java:7)  
at Overflowing.overflow(Overflowing.java:7)
```

Recursion







**Tail recursion: when the last
step in a method recurses**

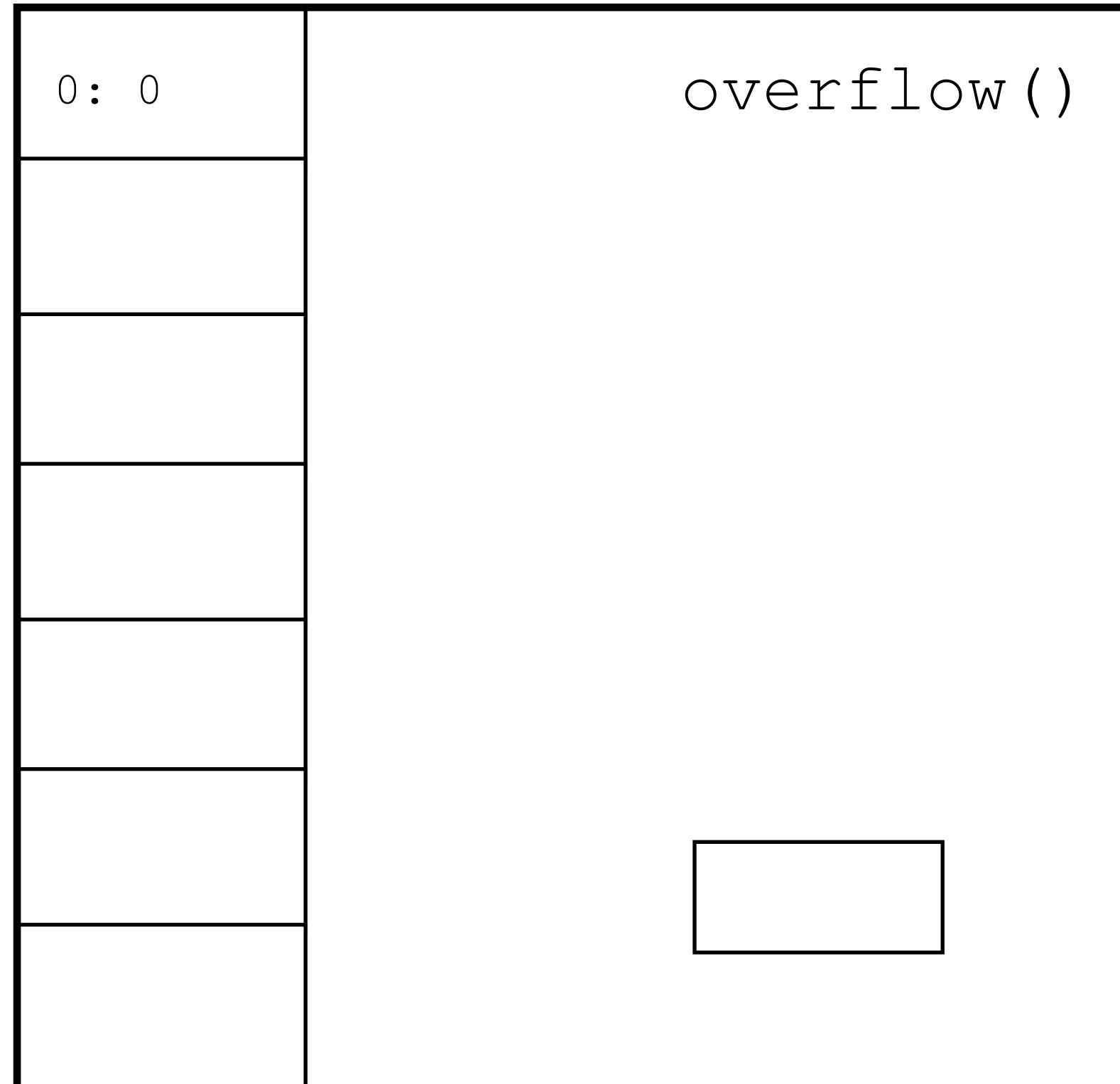
Not a Tail Call

```
public long factorial(long n) {  
    if (n <= 0) {  
        return 1;  
    }  
  
    return n * factorial(n - 1);  
}
```

“Fail” Call Recursion

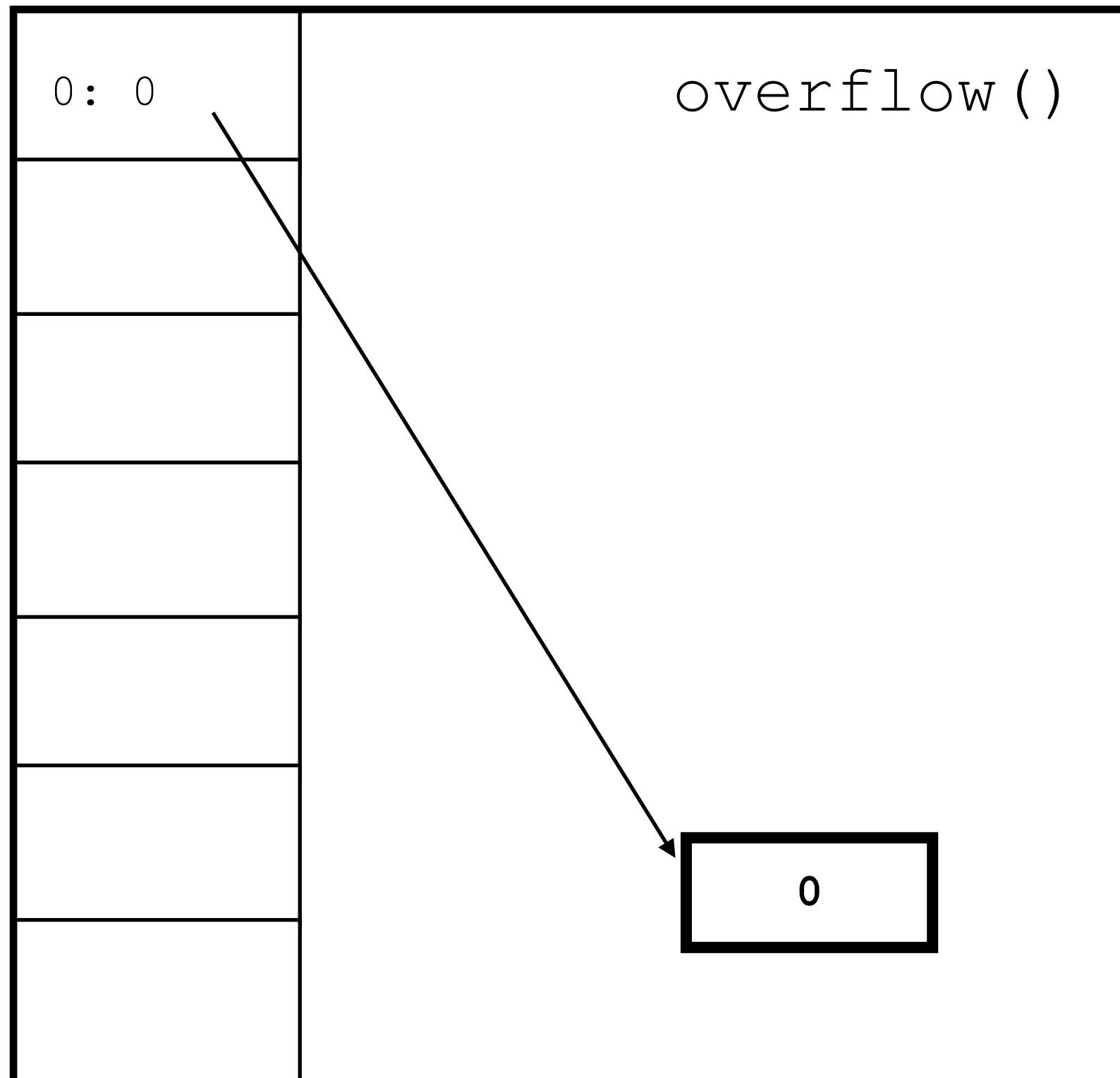
```
public class Overflowing {  
    public static void overflow(int current) {  
        overflow(current + 1);  
    }  
}
```

“Fail” Call Recursion



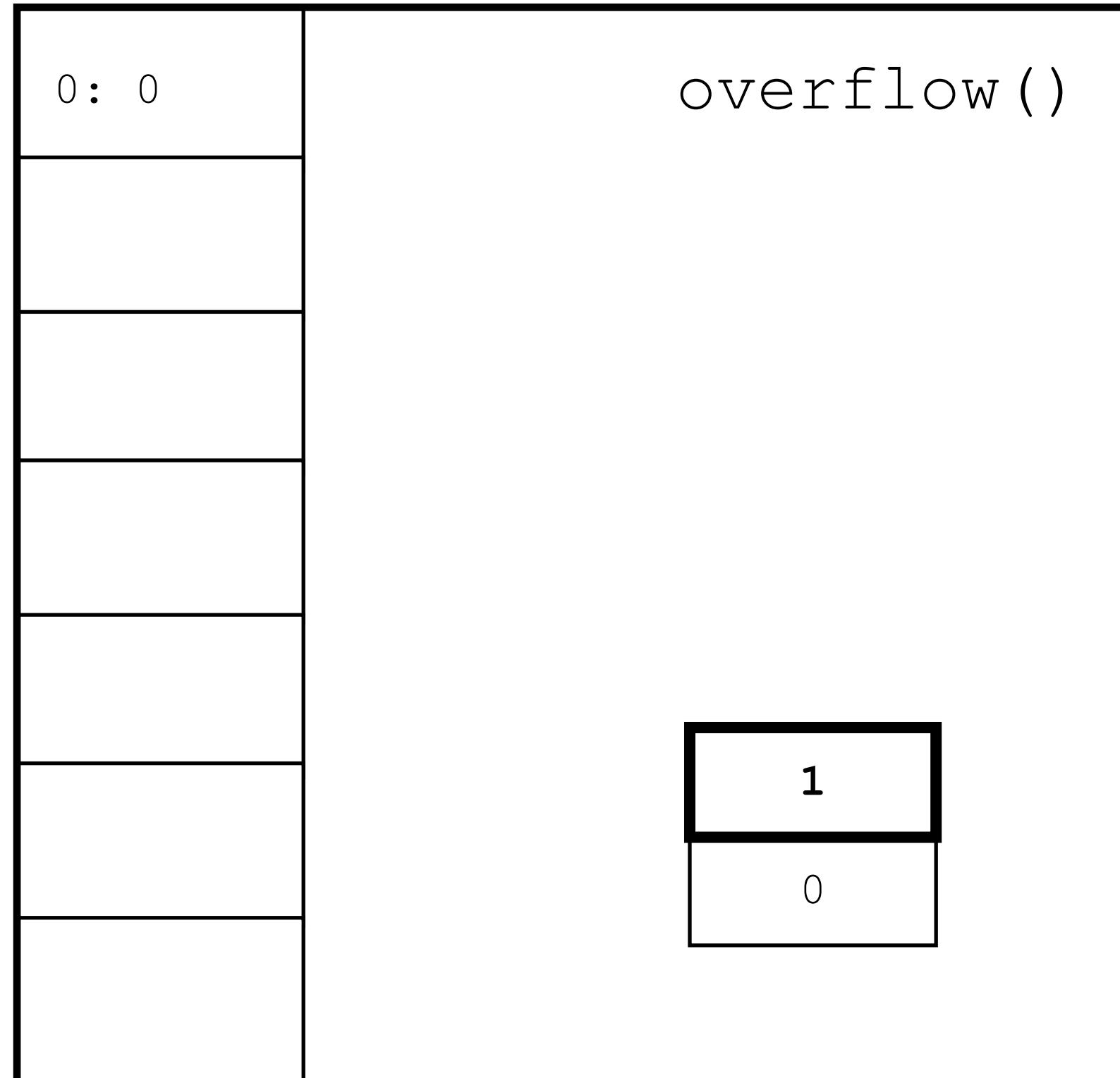
```
public static void overflow(int);  
0: iload_0  
1: iconst_1  
2: iadd  
3: invokestatic #2 //overflow: (I)V  
6: return
```

“Fail” Call Recursion



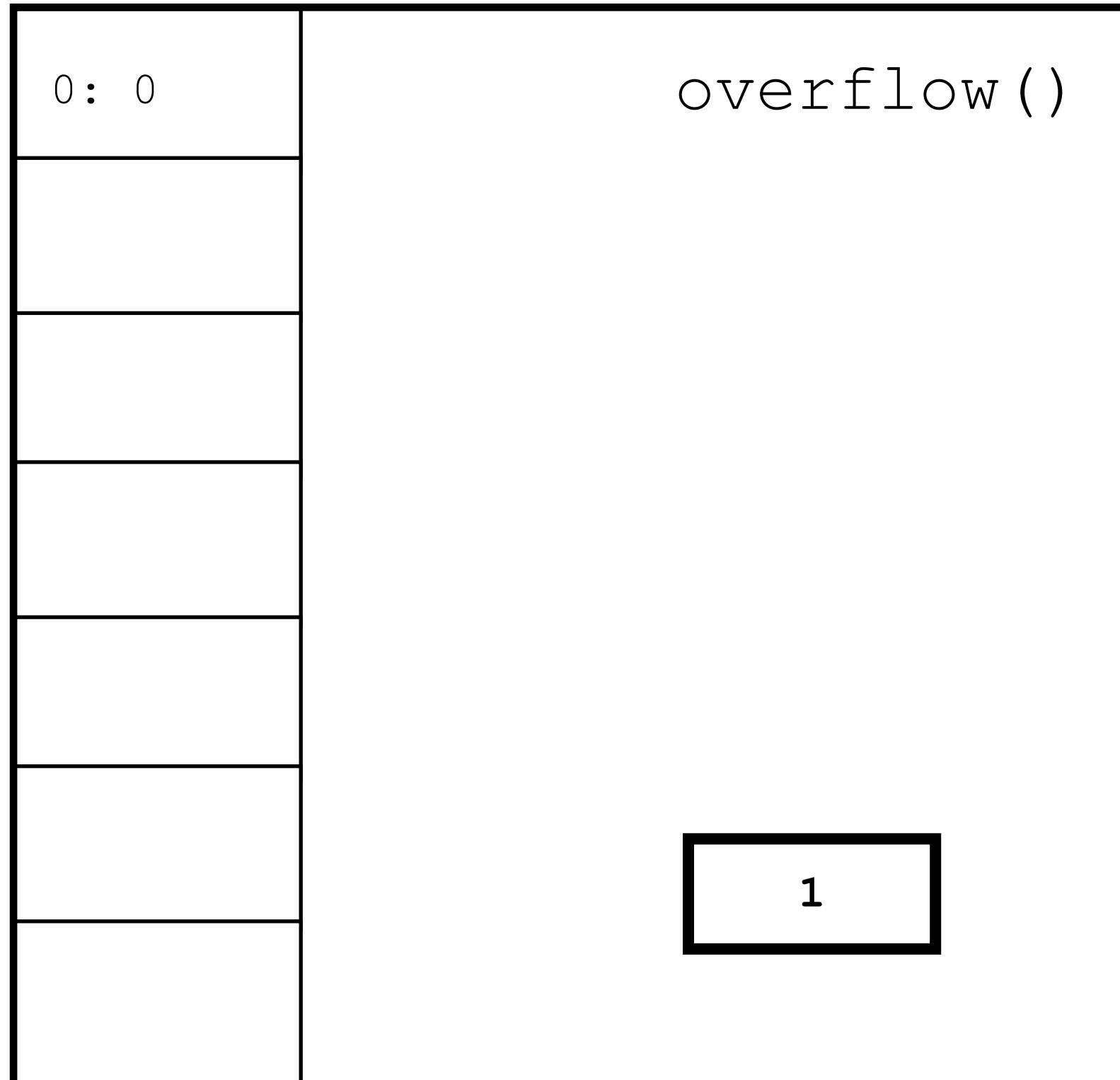
```
public static void overflow(int);  
0: iload_0  
1: iconst_1  
2: iadd  
3: invokestatic #2 //overflow: (I)V  
6: return
```

“Fail” Call Recursion



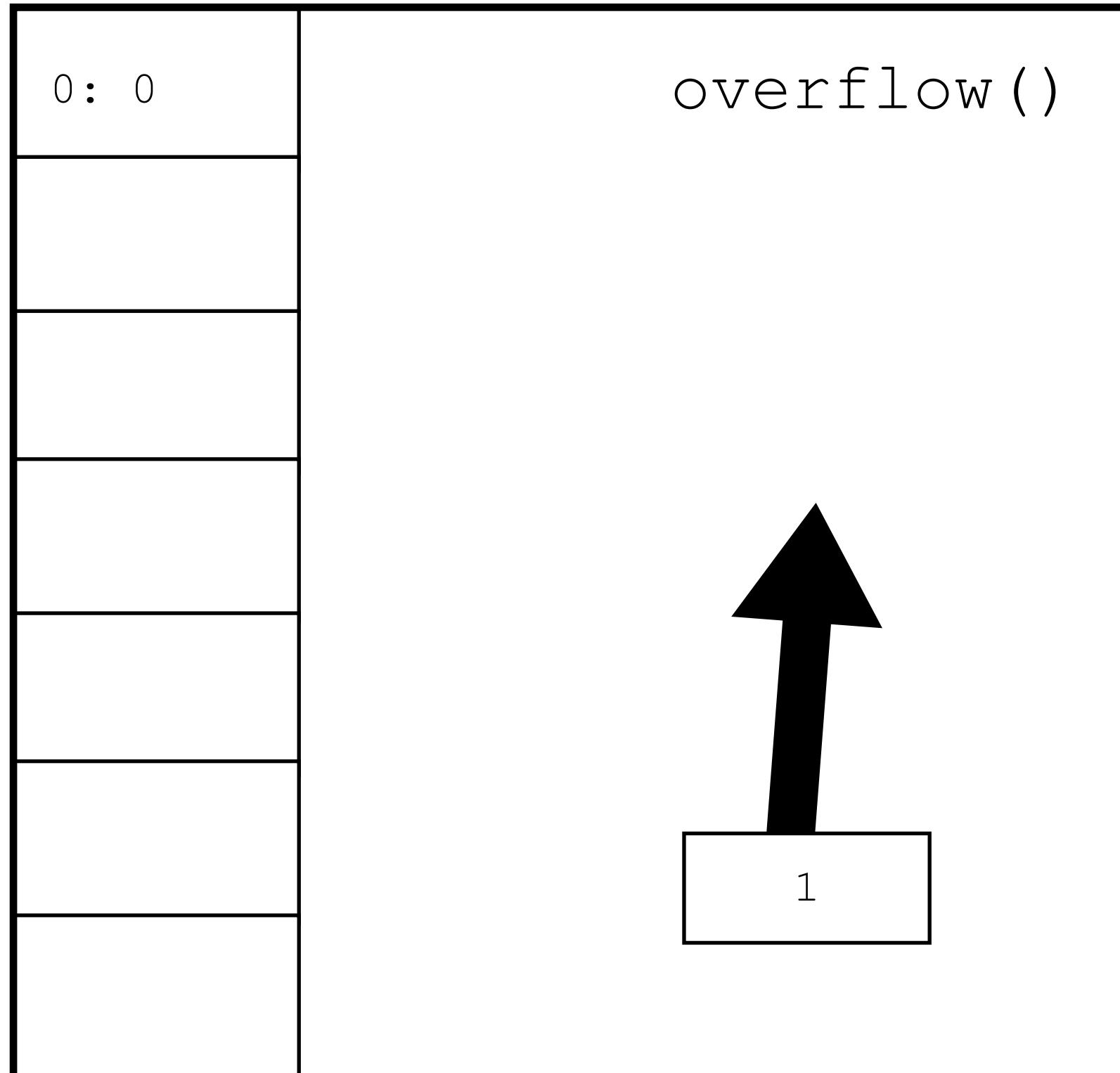
```
public static void overflow(int);  
0: iload_0  
1: iconst_1  
2: iadd  
3: invokestatic #2 //overflow: (I)V  
6: return
```

“Fail” Call Recursion

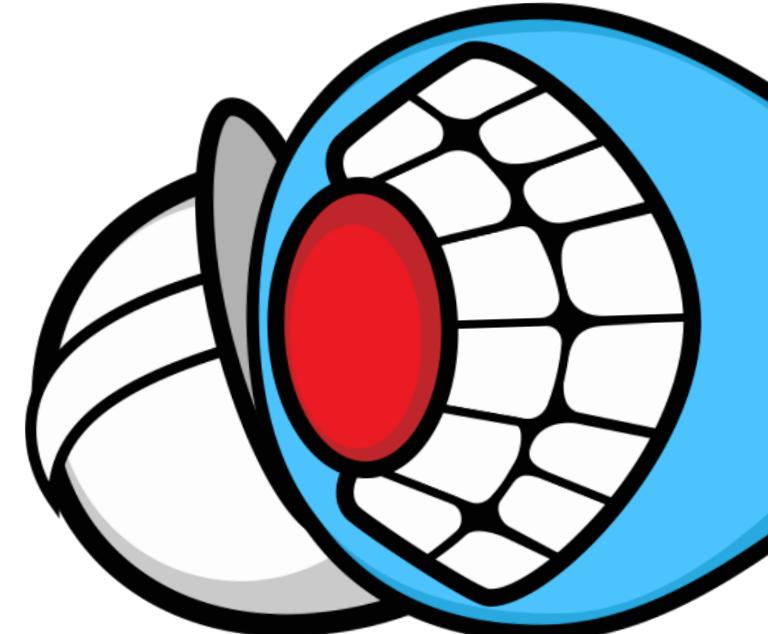


```
public static void overflow(int);  
0: iload_0  
1: iconst_1  
2: iadd  
3: invokestatic #2 //overflow: (I)V  
6: return
```

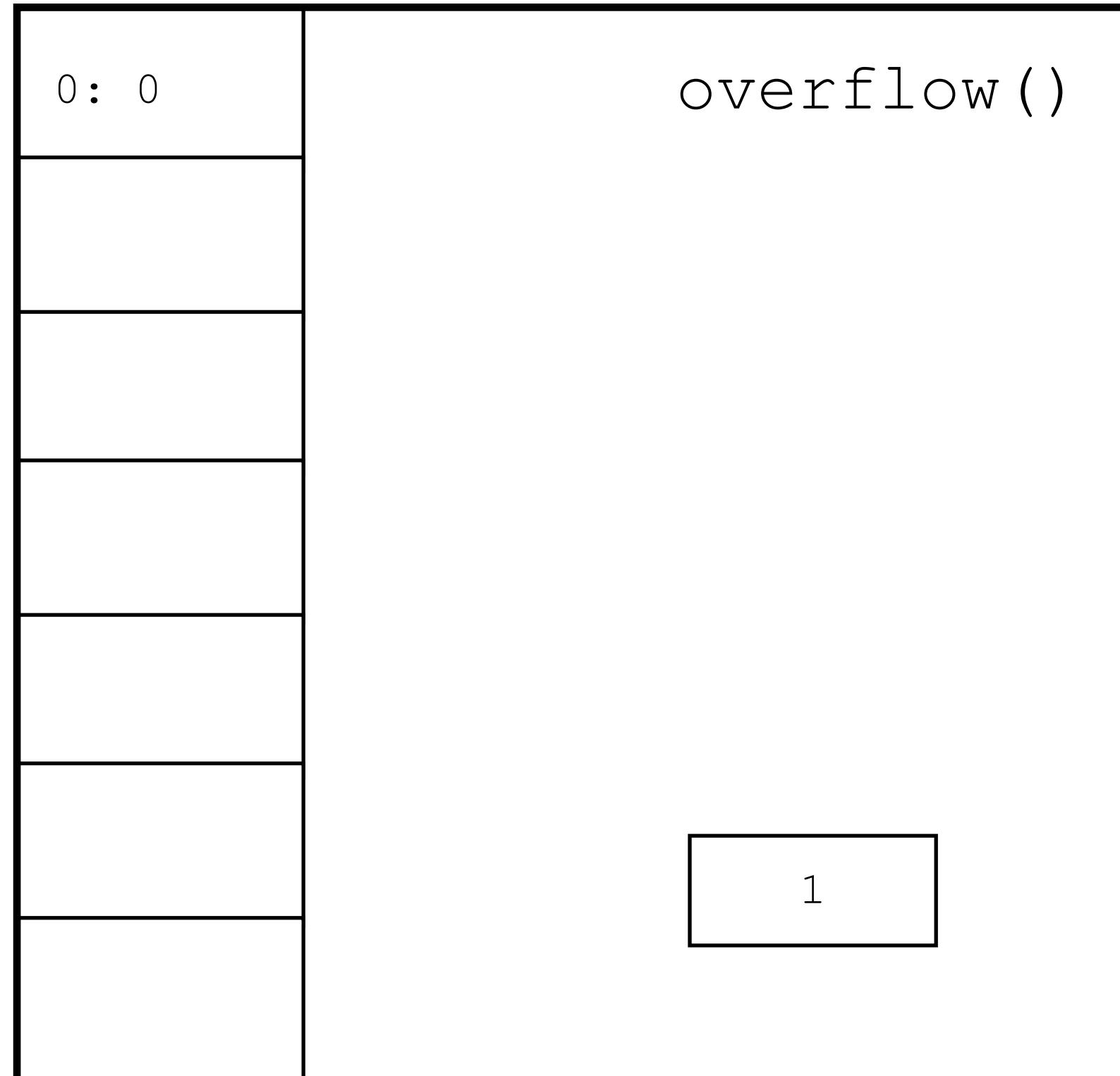
“Fail” Call Recursion



```
public static void overflow(int);  
0: iload_0  
1: iconst_1  
2: iadd  
3: invokestatic #2 //overflow: (I)V  
6: return
```



“Fail” Call Recursion



```
public static void overflow(int);  
0: iload_0  
1: iconst_1  
2: iadd  
3: invokestatic #2 //overflow: (I)V  
6: return
```



Java



Kotlin

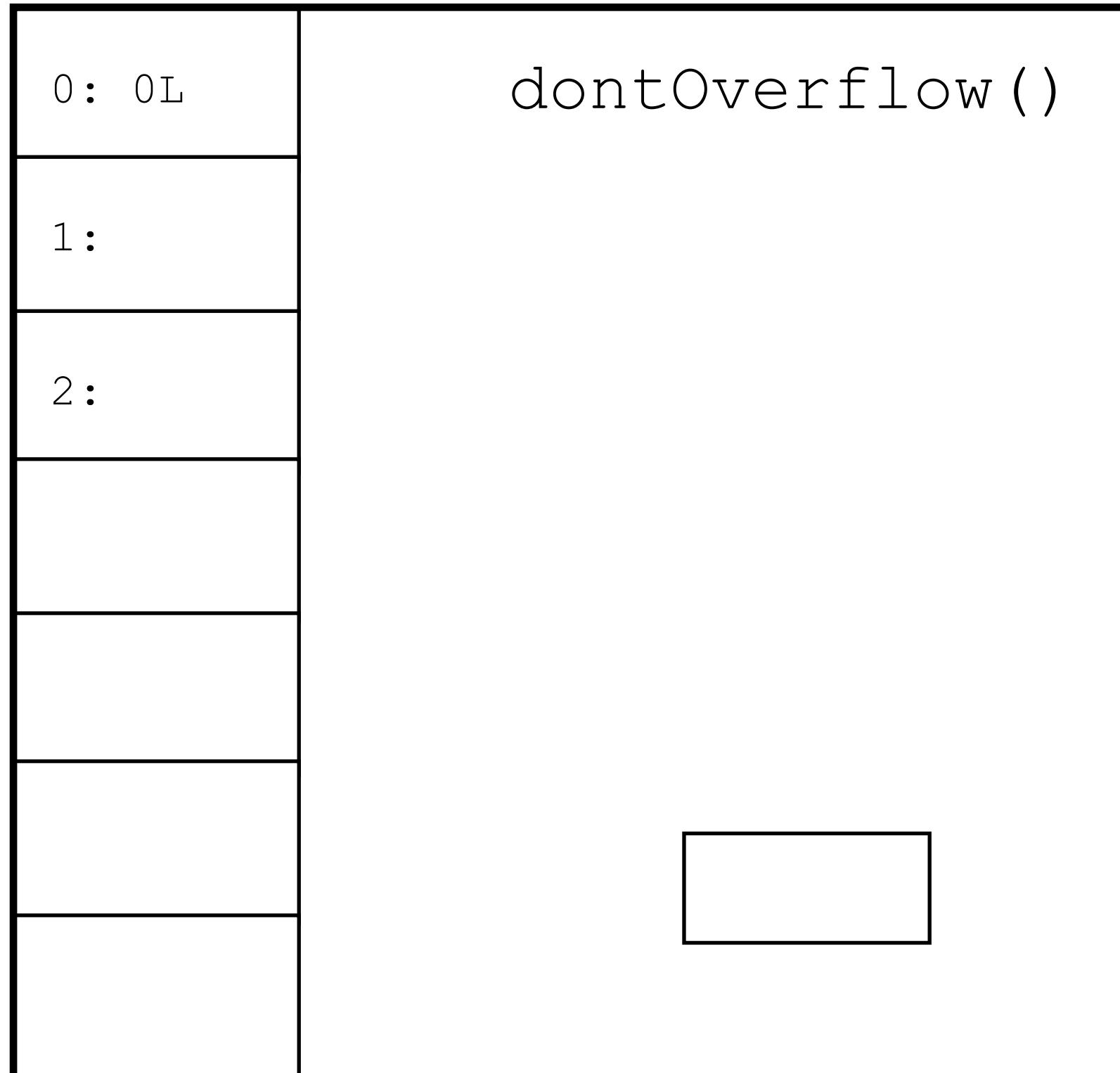


@jasonrclark

Tail Call Recursion

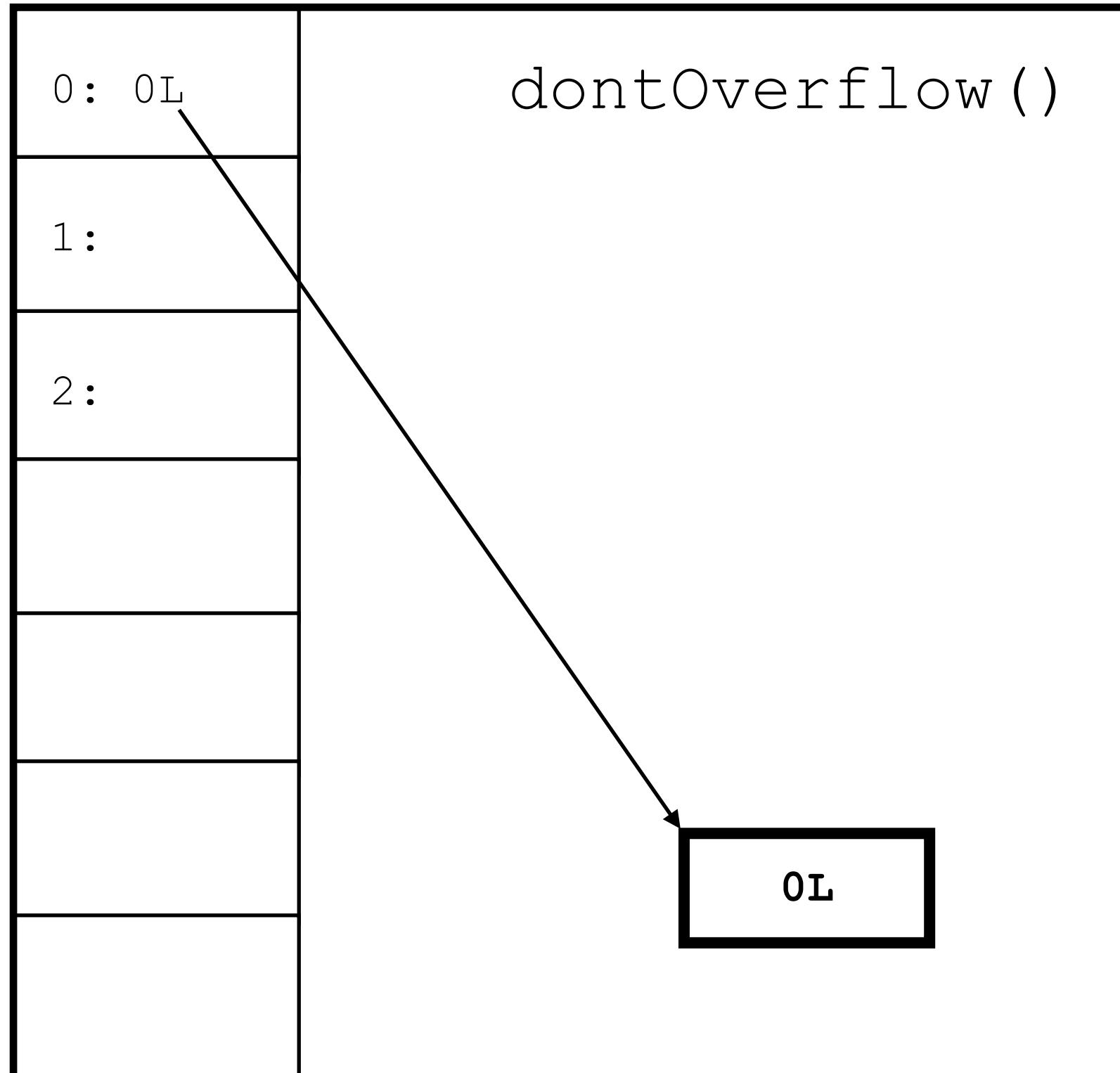
```
tailrec fun dontOverflow(n: Long) {  
    dontOverflow(n + 1)  
}
```

Tail Call Recursion



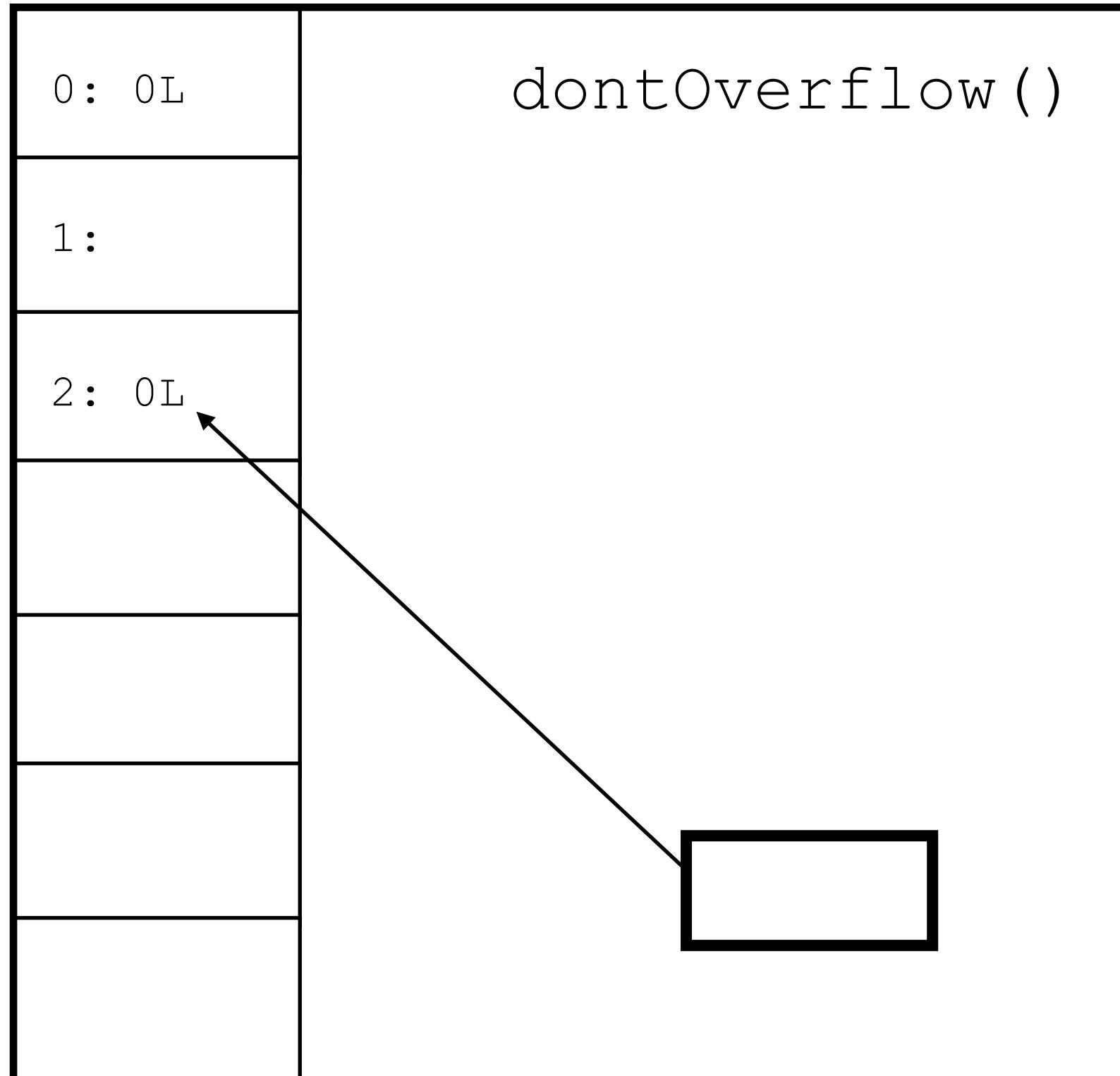
```
public static void dontOverflow(long);  
0: lload_0  
1: lstore_2  
2: lload_2  
3: lconst_1  
4: ladd  
5: lstore_2  
6: goto 2
```

Tail Call Recursion



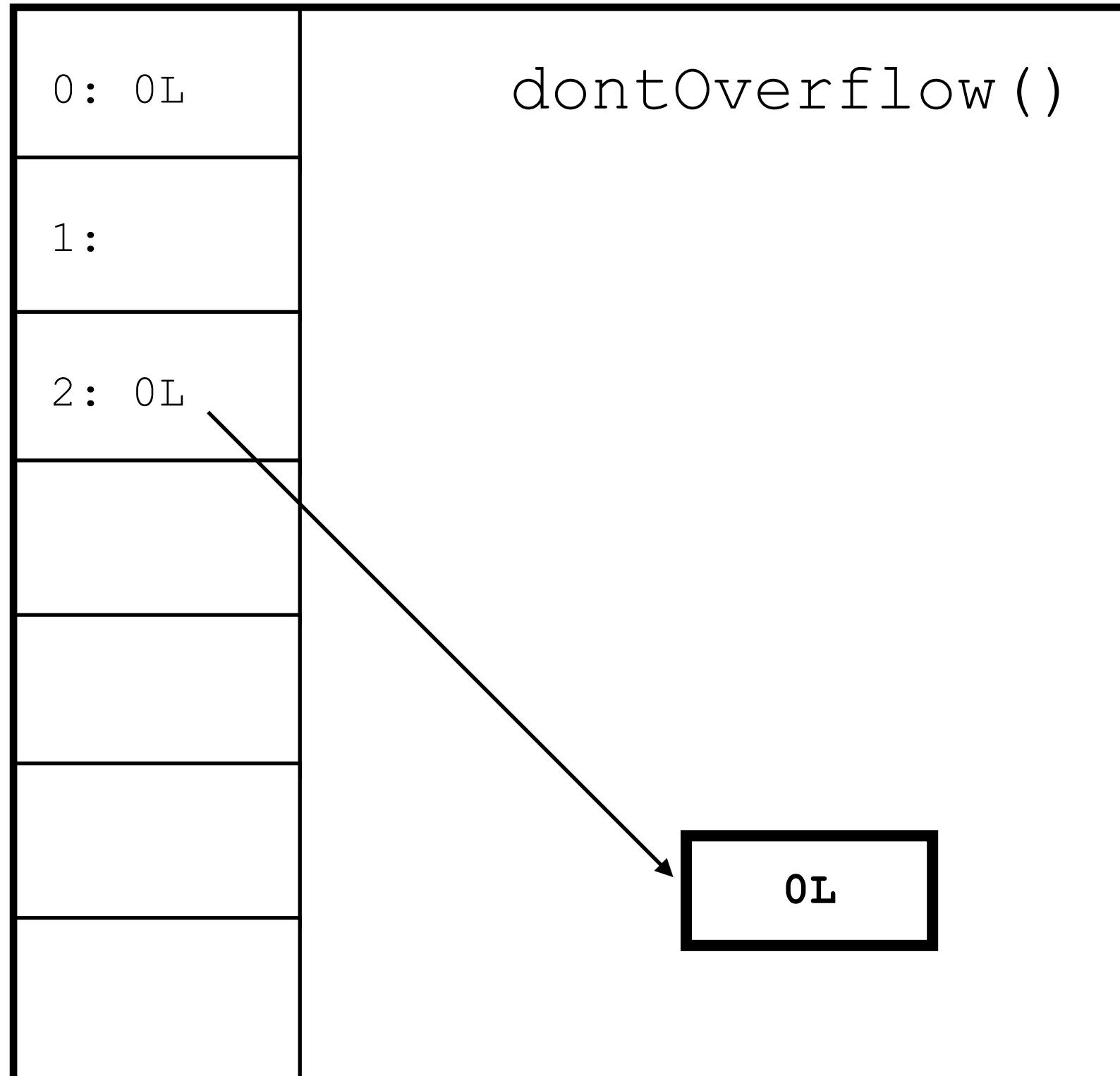
```
public static void dontOverflow(long);  
0: lload_0  
1: lstore_2  
2: lload_2  
3: lconst_1  
4: ladd  
5: lstore_2  
6: goto 2
```

Tail Call Recursion



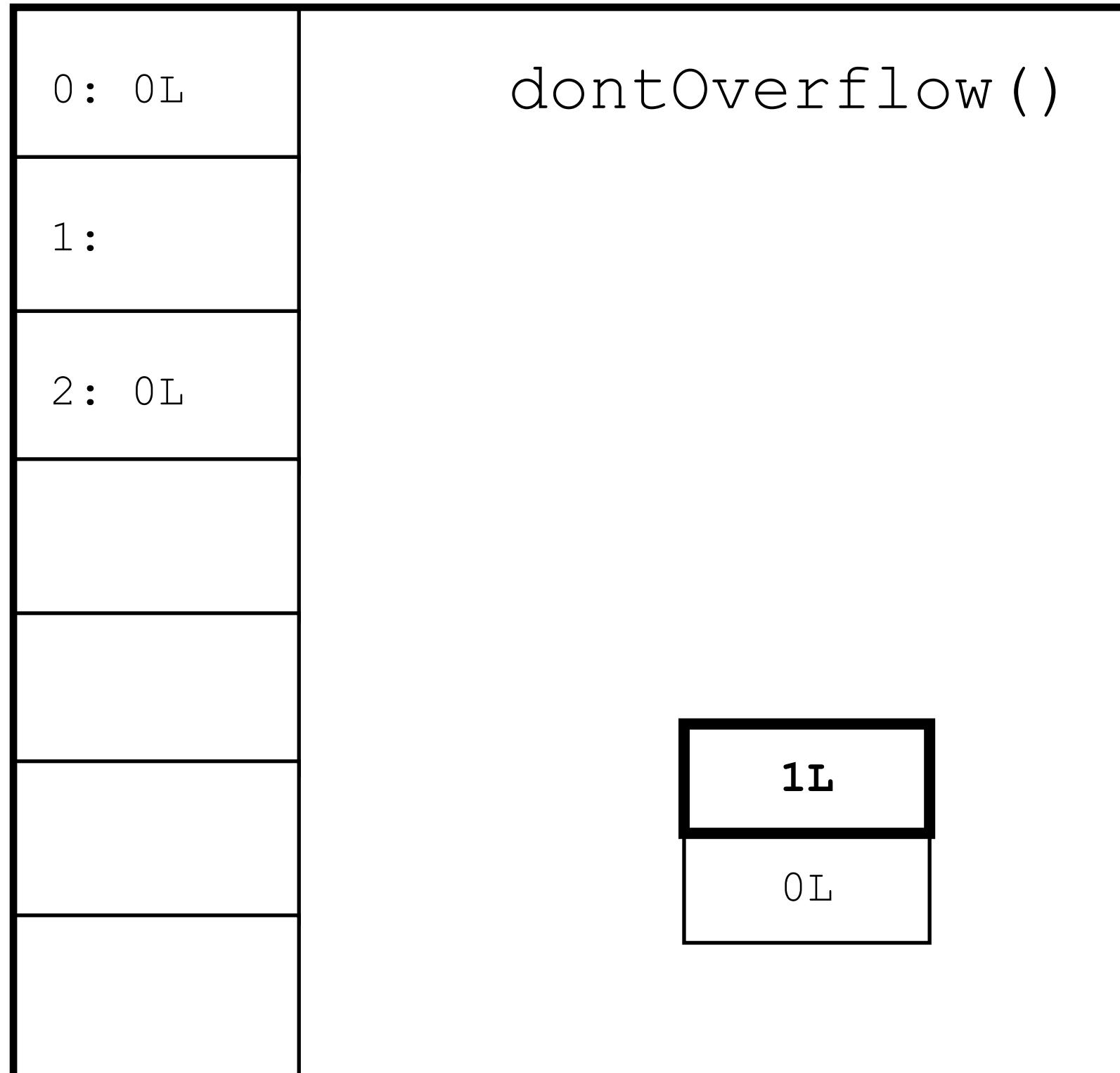
```
public static void dontOverflow(long);  
0: lload_0  
1: lstore_2  
2: lload_2  
3: lconst_1  
4: ladd  
5: lstore_2  
6: goto 2
```

Tail Call Recursion



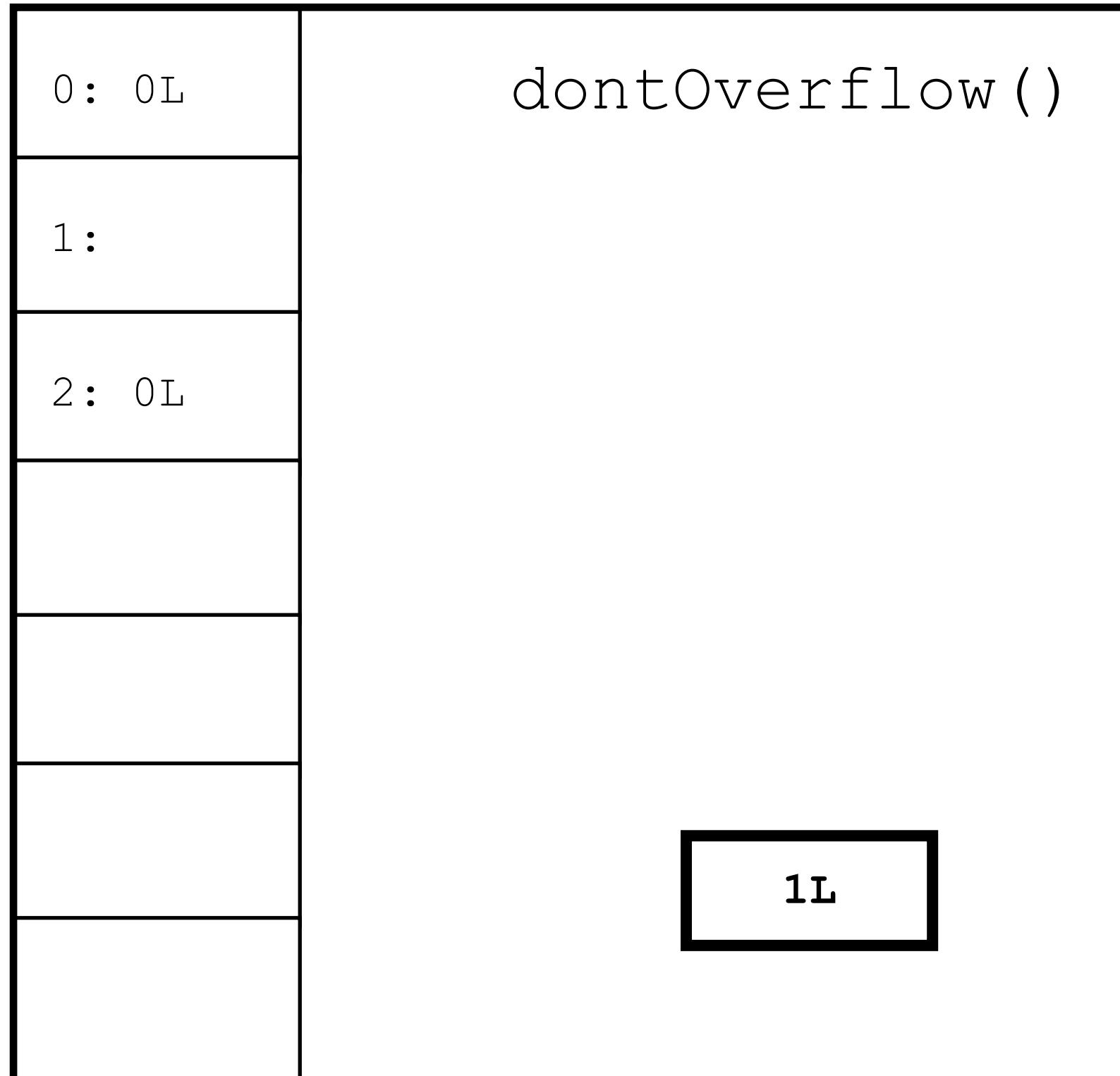
```
public static void dontOverflow(long);  
    0: lload_0  
    1: lstore_2  
2: lload_2  
    3: lconst_1  
    4: ladd  
    5: lstore_2  
    6: goto 2
```

Tail Call Recursion



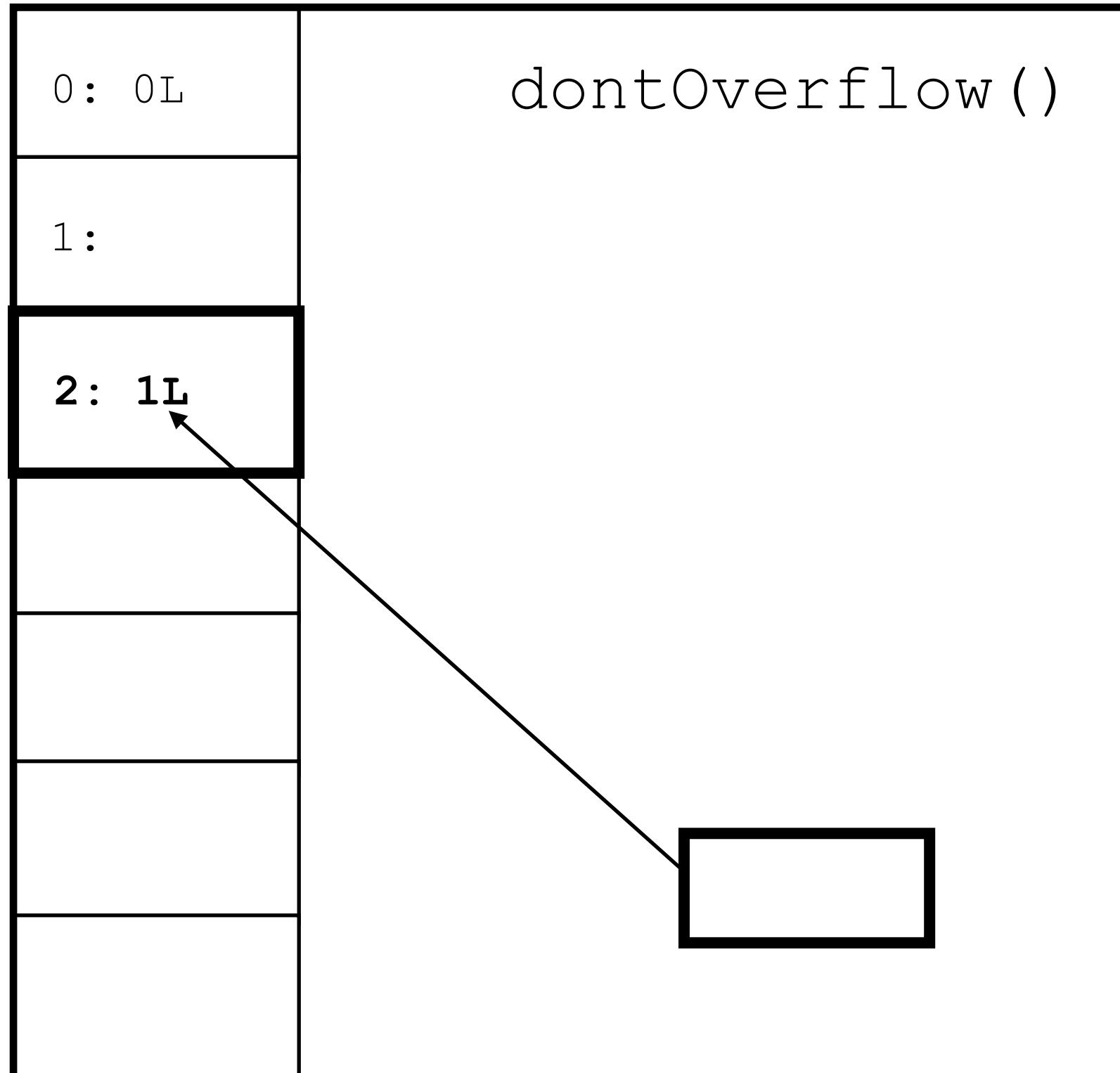
```
public static void dontOverflow(long);  
 0: lload_0  
 1: lstore_2  
 2: lload_2  
 3: lconst_1  
 4: ladd  
 5: lstore_2  
 6: goto 2
```

Tail Call Recursion



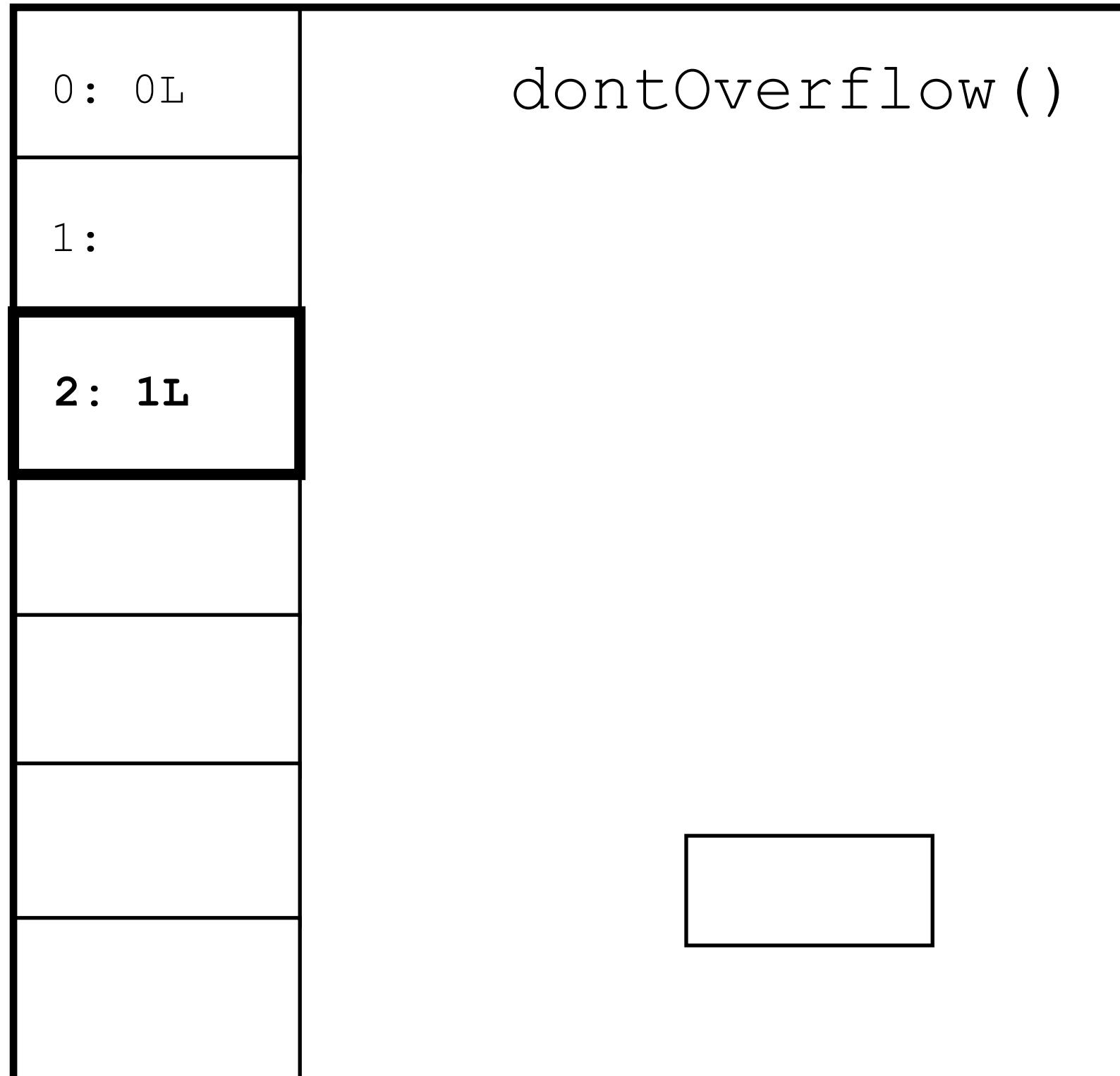
```
public static void dontOverflow(long);  
0: lload_0  
1: lstore_2  
2: lload_2  
3: lconst_1  
4: ladd  
5: lstore_2  
6: goto 2
```

Tail Call Recursion



```
public static void dontOverflow(long);  
 0: lload_0  
 1: lstore_2  
 2: lload_2  
 3: lconst_1  
 4: ladd  
5: lstore_2  
 6: goto 2
```

Tail Call Recursion



```
public static void dontOverflow(long);  
    0: lload_0  
    1: lstore_2  
2: lload_2  
    3: lconst_1  
    4: ladd  
    5: lstore_2  
    6: goto 2
```





Infinite loop > overflowing



Recursive function -> tail recursive



Tail recursive -> Loop

Recursion == Looping An emoji of a yellow face with an exploding brain, symbolizing a headache or a complex problem.

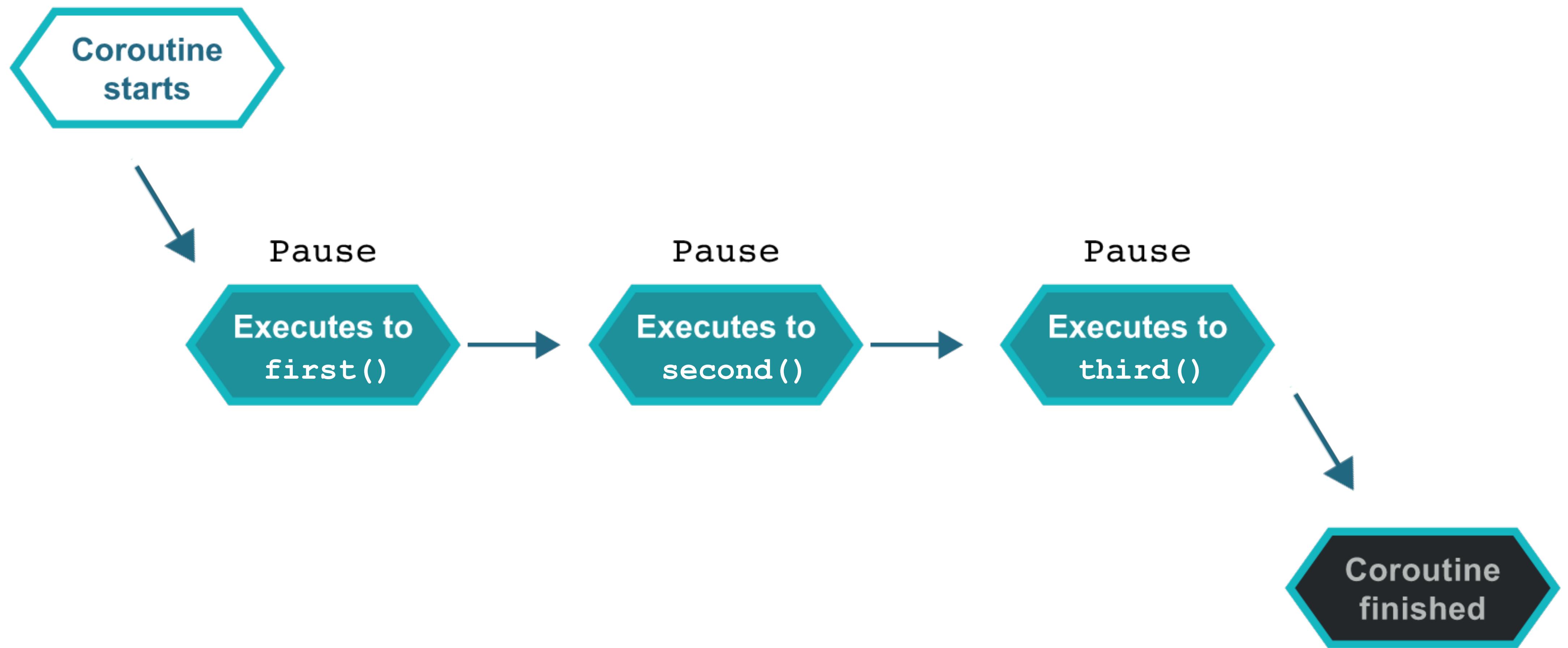
Example: Coroutines

Coroutines

```
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.launch

fun main() {
    GlobalScope.launch {
        first()
        second()
        third()
    }
}

suspend fun first() {
    println("first")
}
```



Coroutines

```
suspend fun first() {  
    println("first")  
}
```

```
suspend fun second() {  
    println("second")  
}
```

```
suspend fun third() {  
    println("third")  
}
```

Coroutines

```
$ kotlinc *.kt
```

```
•  
├ Coroutines.kt  
└ CoroutinesKt$main$1.class  
  └ CoroutinesKt.class
```

Coroutines

```
$ kotlinc *.kt
```

```
•  
├ Coroutines.kt  
└ CoroutinesKt$main$1.class  
  └ CoroutinesKt.class
```

Coroutines

```
// CoroutinesKt.class
public static final void main();
0: getstatic      #12 // Field kotlinx/coroutines/GlobalScope.INSTANCE:...
3: checkcast     #14 // class kotlinx/coroutines/CoroutineScope
6: aconst_null
7: aconst_null
8: new           #16 // class CoroutinesKt$main$1
11: dup
12: aconst_null
13: invokespecial #20 // Method CoroutinesKt$main$1."<init>":...
16: checkcast     #22 // class kotlin/jvm/functions/Function2
19: iconst_3
20: aconst_null
21: invokestatic   #28 // Method kotlinx/coroutines/BuildersKt.launch$default:
                      // (Lkotlinx/coroutines/CoroutineScope;
                      // Lkotlin/coroutines/CoroutineContext;
                      // Lkotlinx/coroutines/CoroutineStart;
                      // Lkotlin/jvm/functions/Function2;
                      // ILjava/lang/Object;)Lkotlinx/coroutines/Job;
```

Coroutines

```
// CoroutinesKt.class
public static final void main();
0: getstatic      #12 // Field kotlinx/coroutines/GlobalScope.INSTANCE:...
3: checkcast      #14 // class kotlinx/coroutines/CoroutineScope
6: aconst_null
7: aconst_null
8: new           #16 // class CoroutinesKt$main$1
11: dup
12: aconst_null
13: invokespecial #20 // Method CoroutinesKt$main$1."<init>":...
16: checkcast      #22 // class kotlin/jvm/functions/Function2
19: iconst_3
20: aconst_null
21: invokestatic   #28 // Method kotlinx/coroutines/BuildersKt.launch$default:
                      // (Lkotlinx/coroutines/CoroutineScope;
                      // Lkotlinx/coroutines/CoroutineContext;
                      // Lkotlinx/coroutines/CoroutineStart;
                      // Lkotlin/jvm/functions/Function2;
                      // ILjava/lang/Object;)Lkotlinx/coroutines/Job;
```

Coroutines

```
// CoroutinesKt.class
public static final void main();
0: getstatic      #12 // Field kotlinx/coroutines/GlobalScope.INSTANCE:...
3: checkcast     #14 // class kotlinx/coroutines/CoroutineScope
6: aconst_null
7: aconst_null
8: new           #16 // class CoroutinesKt$main$1
11: dup
12: aconst_null
13: invokespecial #20 // Method CoroutinesKt$main$1."<init>":...
16: checkcast     #22 // class kotlin/jvm/functions/Function2
19: iconst_3
20: aconst_null
21: invokestatic   #28 // Method kotlinx/coroutines/BuildersKt.launch$default:
                      // (Lkotlinx/coroutines/CoroutineScope;
                      // Lkotlin/coroutines/CoroutineContext;
                      // Lkotlinx/coroutines/CoroutineStart;
                      // Lkotlin/jvm/functions/Function2;
                      // ILjava/lang/Object;)Lkotlinx/coroutines/Job;
```

Coroutines

```
// CoroutinesKt.class
public static final void main();
0: getstatic      #12 // Field kotlinx/coroutines/GlobalScope.INSTANCE:...
3: checkcast     #14 // class kotlinx/coroutines/CoroutineScope
6: aconst_null
7: aconst_null
8: new           #16 // class CoroutinesKt$main$1
11: dup
12: aconst_null
13: invokespecial #20 // Method CoroutinesKt$main$1."<init>":...
16: checkcast     #22 // class kotlin/jvm/functions/Function2
19: iconst_3
20: aconst_null
21: invokestatic #28 // Method kotlinx/coroutines/BuildersKt.launch$default:
    // (Lkotlinx/coroutines/CoroutineScope;
    // Lkotlin/coroutines/CoroutineContext;
    // Lkotlinx/coroutines/CoroutineStart;
    // Lkotlin/jvm/functions/Function2;
    // ILjava/lang/Object;)Lkotlinx/coroutines/Job;
```

Coroutines

```
// CoroutinesKt.class
public static final void main();
0: getstatic      #12 // Field kotlinx/coroutines/GlobalScope.INSTANCE:...
3: checkcast     #14 // class kotlinx/coroutines/CoroutineScope
6: aconst_null
7: aconst_null
8: new           #16 // class CoroutinesKt$main$1
11: dup
12: aconst_null
13: invokespecial #20 // Method CoroutinesKt$main$1."<init>":...
16: checkcast     #22 // class kotlin/jvm/functions/Function2
19: iconst_3
20: aconst_null
21: invokestatic   #28 // Method kotlinx/coroutines/BuildersKt.launch$default:
                      // (Lkotlinx/coroutines/CoroutineScope;
                      // Lkotlin/coroutines/CoroutineContext;
                      // Lkotlinx/coroutines/CoroutineStart;
                      // Lkotlin/jvm/functions/Function2;
                      // ILjava/lang/Object;)Lkotlinx/coroutines/Job;
```

!? What's in that generated class?



@jasonrclark

Coroutines

```
// CoroutinesKt$main$1
public final java.lang.Object invoke(
    kotlinx.coroutines.CoroutineScope,
    kotlin.coroutines.Continuation<? super kotlin.Unit>);

0: aload_0
1: aload_1
2: aload_2
3: invokevirtual #91      // Method create:(Ljava/lang/Object;...;
6: checkcast      #2      // class CoroutinesKt$main$1
9: getstatic       #69     // Field kotlin/Unit.INSTANCE:Lkotlin/Unit;
12: invokevirtual #93     // Method invokeSuspend:...;
15: areturn
```

Coroutines

```
// CoroutinesKt$main$1
public final java.lang.Object invokeSuspend(java.lang.Object);
0: invokestatic #39 // Method kotlin/coroutines/intrinsics/...
3: astore_2
4: aload_0
5: getfield #43 // Field label:I
8: tableswitch { // 0 to 3
    0: 40
    1: 63
    2: 88
    3: 113
    default: 123
}
...
...
```

Coroutines

```
// CoroutinesKt$main$1
public final java.lang.Object invokeSuspend(java.lang.Object);
    0: invokestatic #39 // Method kotlin/coroutines/intrinsics/...
    3: astore_2
    4: aload_0
    5: getfield      #43 // Field label:I
8: tableswitch { // 0 to 3
    0: 40
    1: 63
    2: 88
    3: 113
default: 123
}
```

Coroutines

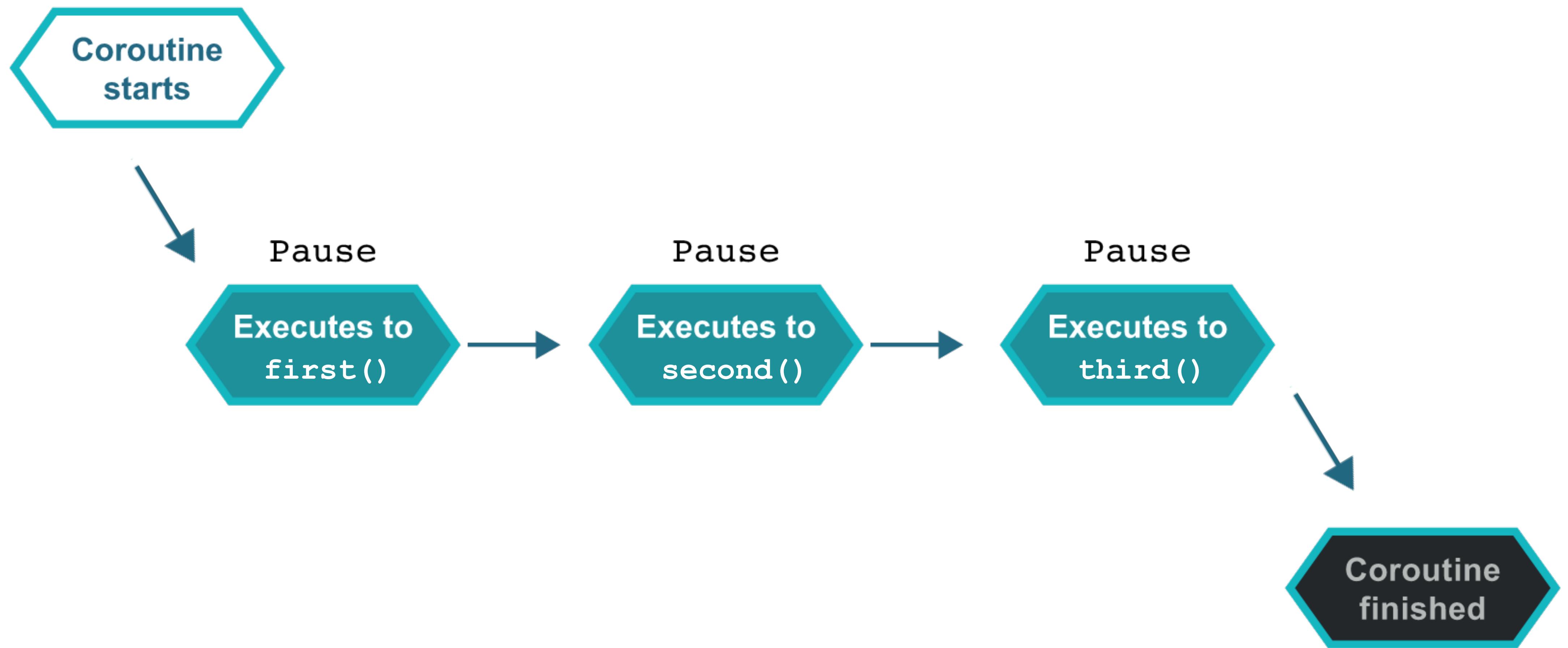
```
// CoroutinesKt$main$1
public final java.lang.Object invokeSuspend(java.lang.Object);
...
40: aload_1
41: invokestatic #49    // Method kotlin/ResultKt.throwOnFailure:...
44: aload_0
45: checkcast      #51    // class kotlin/coroutines/Continuation
48: aload_0
49: iconst_1
50: putfield       #43    // Field label:I
53: invokestatic #57    // Method CoroutinesKt.first:...;
56: dup
57: aload_2
58: if_acmpne      68
61: aload_2
62: areturn
```

Coroutines

```
// CoroutinesKt$main$1
public final java.lang.Object invokeSuspend(java.lang.Object);
...
40:  aload_1
41:  invokestatic #49    // Method kotlin/ResultKt.throwOnFailure:...
44:  aload_0
45:  checkcast     #51    // class kotlin/coroutines/Continuation
48:  aload_0
49:  iconst_1
50:  putfield      #43    // Field label:I
53:  invokestatic #57    // Method CoroutinesKt.first:...
56:  dup
57:  aload_2
58:  if_acmpne      68
61:  aload_2
62:  areturn
```

Coroutines

```
// CoroutinesKt$main$1
public final java.lang.Object invokeSuspend(java.lang.Object);
...
40:  aload_1
41:  invokestatic #49    // Method kotlin/ResultKt.throwOnFailure:...
44:  aload_0
45:  checkcast     #51    // class kotlin/coroutines/Continuation
48:  aload_0
49:  iconst_1
50:  putfield      #43    // Field label:I
53:  invokestatic #57    // Method CoroutinesKt.first:...;
56:  dup
57:  aload_2
58:  if_acmpne   68
61:  aload_2
62:  areturn
```





Bytecode reveals how Kotlin can pause/resume

Wrap-up

- **Tools for digging**
 - How the JVM runs
 - Just enough bytecode to be dangerous
- **Examples**
 - Losing concurrent updates
 - Tail recursion
 - Coroutines