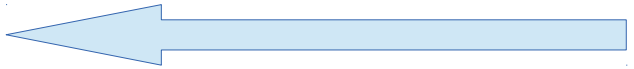
Let's see how data is created
in the computer's memory
as code is executes

```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

STACK

HEAP

```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```



The first line in this program is the beginning of a class definition. No data is created on memory yet. The computer just knows that we are creating a new complex type.

HEAP

STACK

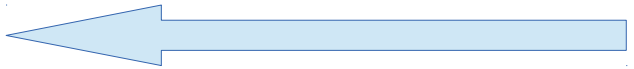
```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

Second, third, and fourth line just complete the definition of class Point. No data is created in memory until an object (an instance) of class Point is declared and initialised.

HEAP

STACK

```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

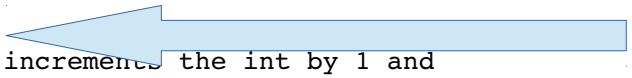


Second, third, and fourth line just complete the definition of class Point. No data is created in memory until an object (an instance) of class Point is declared and initialised.

HEAP

STACK

```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

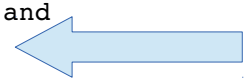


Second, third, and fourth line just complete the definition of class Point. No data is created in memory until an object (an instance) of class Point is declared and initialised.

HEAP

STACK

```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```



Comments are just ignored.

HEAP

STACK

```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

HEAP

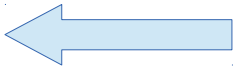
Here we start defining a new method to use it later. No code is executed (and no data created) yet.

In order to execute the code inside a method, we need to call it.

STACK

```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

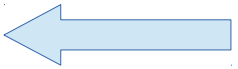
HEAP



All this code is defined now, but
Is not executed at the moment.

STACK

```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```



All this code is defined now, but
Is not executed at the moment.

HEAP

STACK

```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

HEAP

All this code is defined now, but
Is not executed at the moment.

STACK

```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

HEAP

All this code is defined now, but
Is not executed at the moment.

STACK

```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

HEAP

All this code is defined now, but
Is not executed at the moment.

STACK

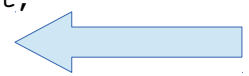
```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

HEAP

All this code is defined now, but
Is not executed at the moment.

STACK

```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

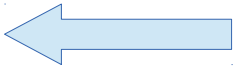


All this code is defined now, but
Is not executed at the moment.

HEAP

STACK

```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```



The execution of the program starts here

HEAP

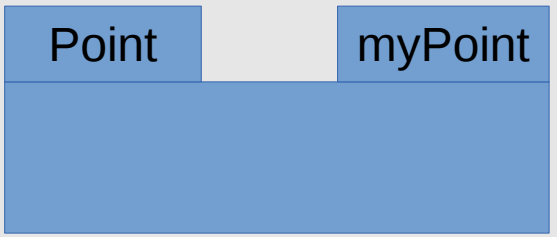
STACK

```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

HEAP

First of all, a variable of type (class) Point is declared...

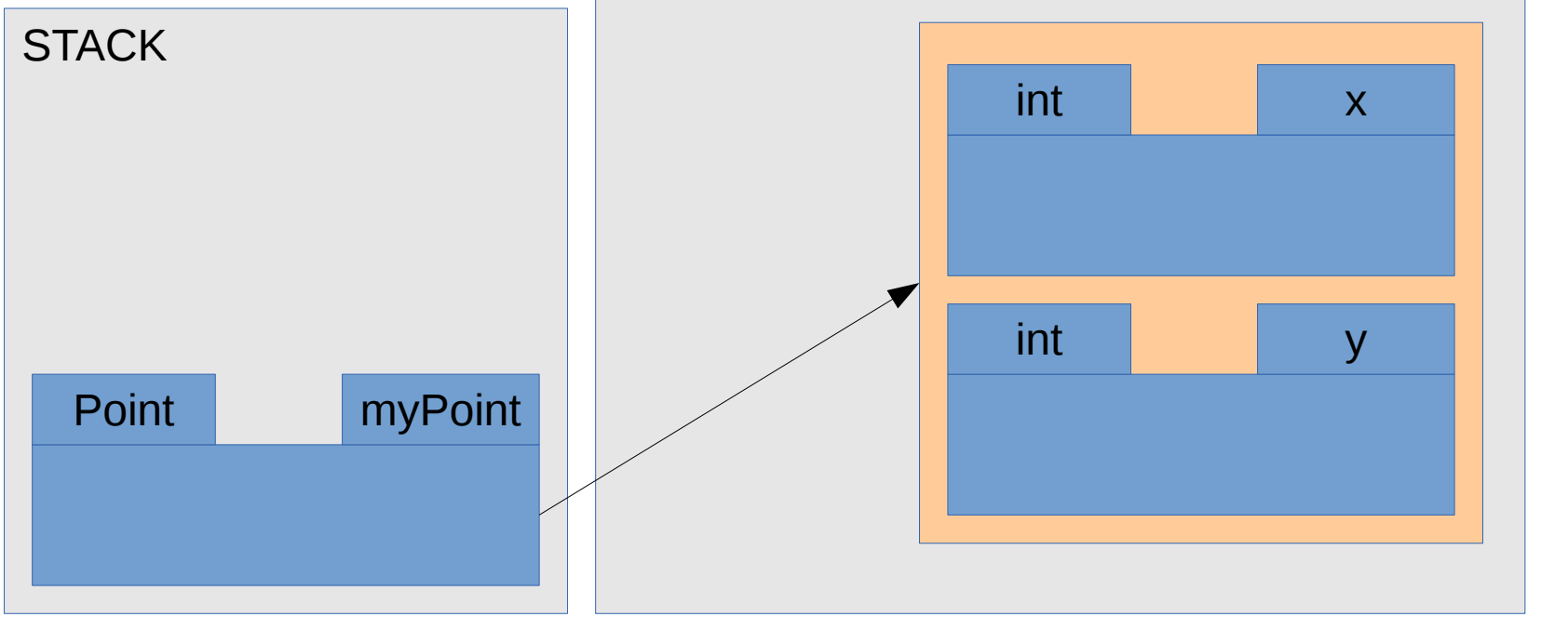
STACK



```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

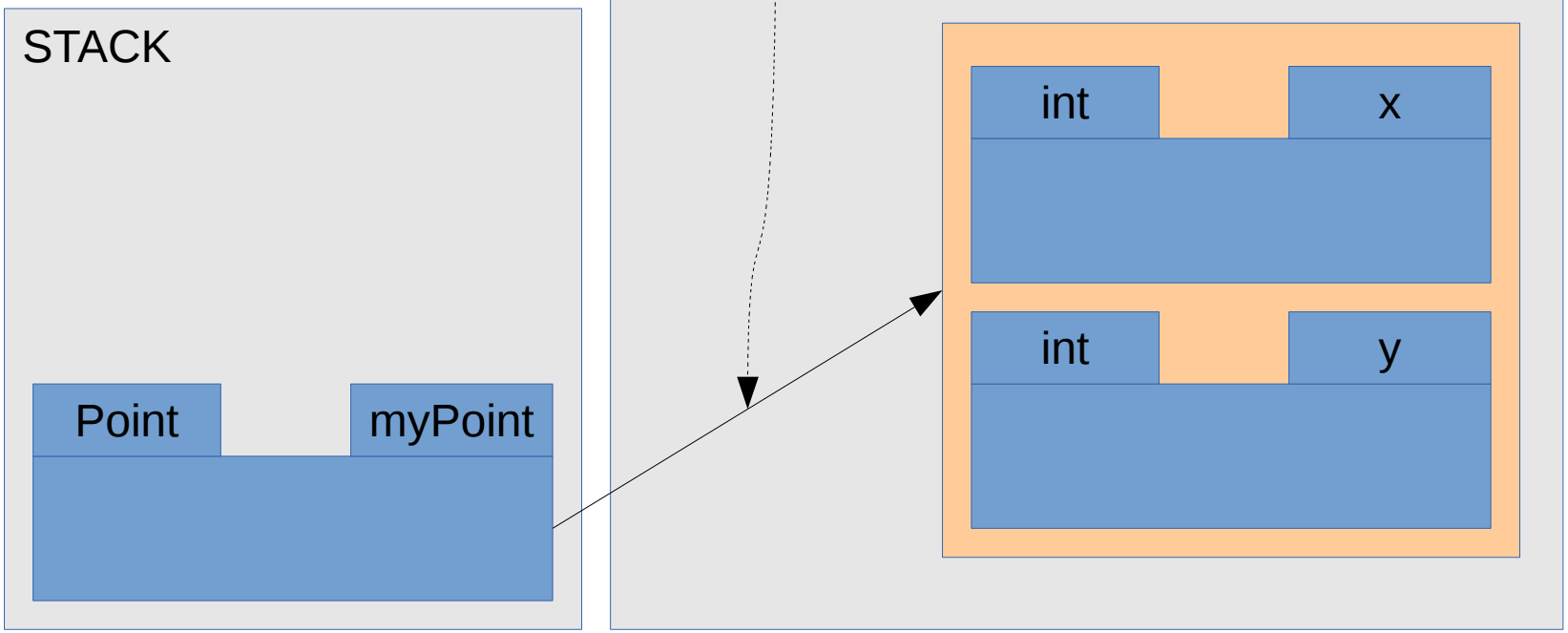
...and its value is set to a new object of type Point (using keyword **new**).

Declaration and initialisation can happen in two different lines or just in one as in this example,

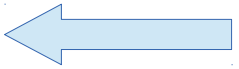


```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

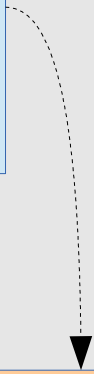
First, note that the value of variable *myPoint* is just a pointer to an object in the heap.

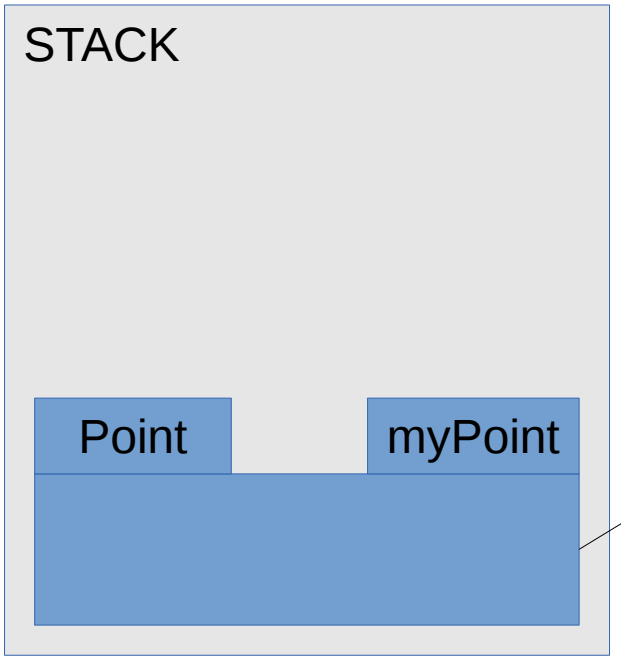


```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```



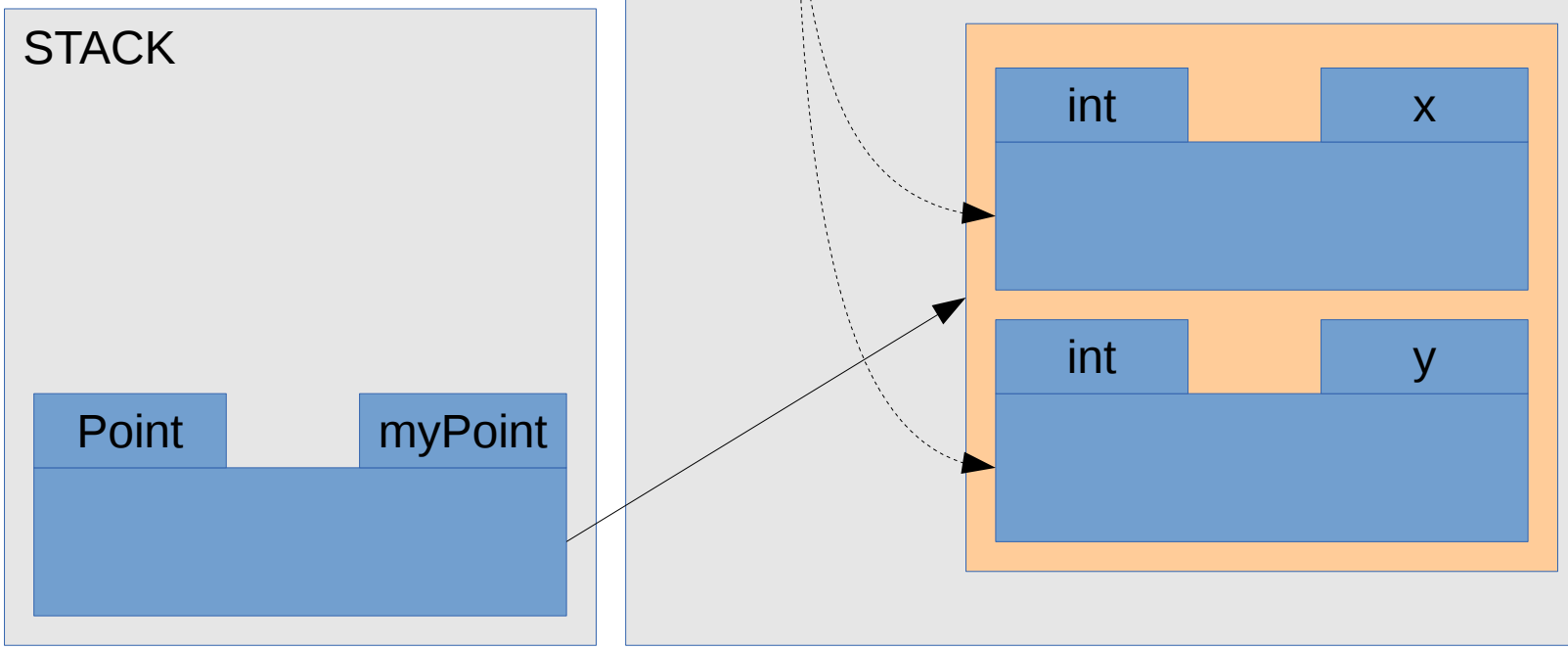
Second, note that the computer knows that the object contains two fields (an **int** called *x*, and another called *y*) because this was defined at the beginning of the program.





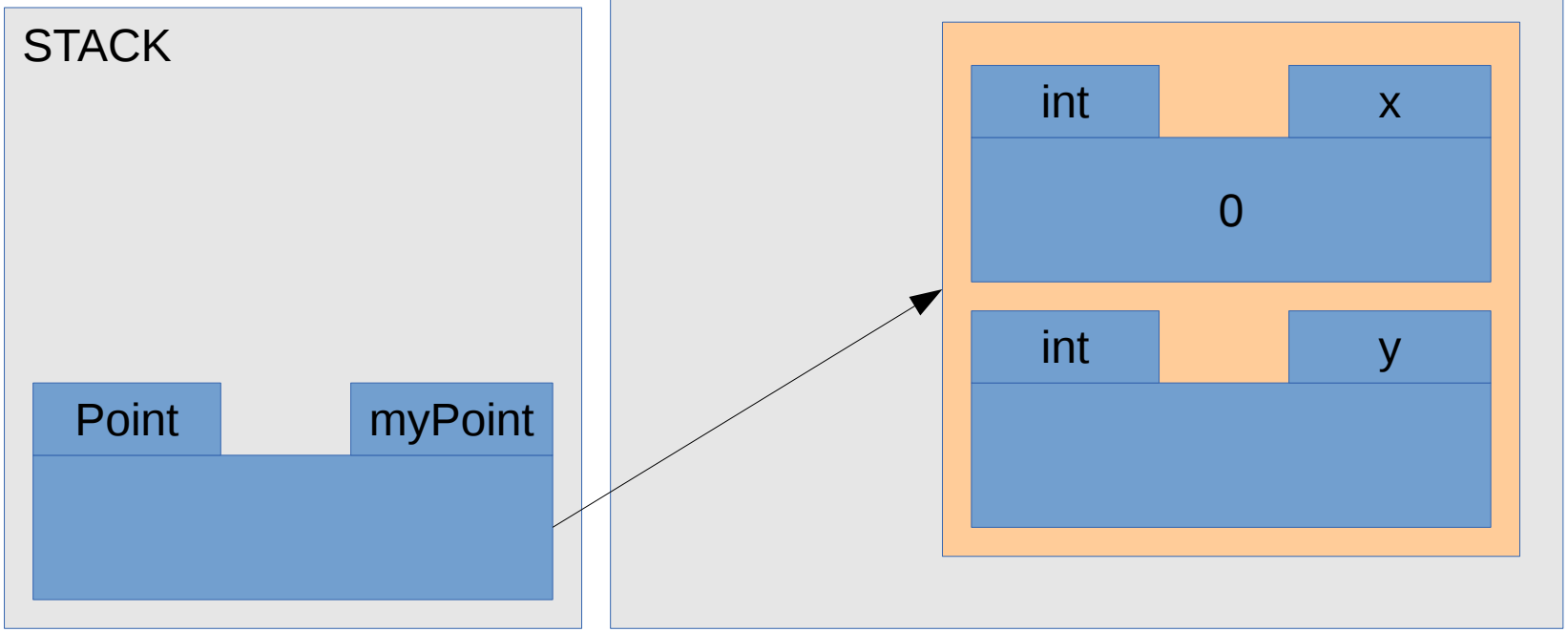
```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

Third, note that the fields do not have any value in them (until they are initialised).

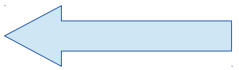


```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

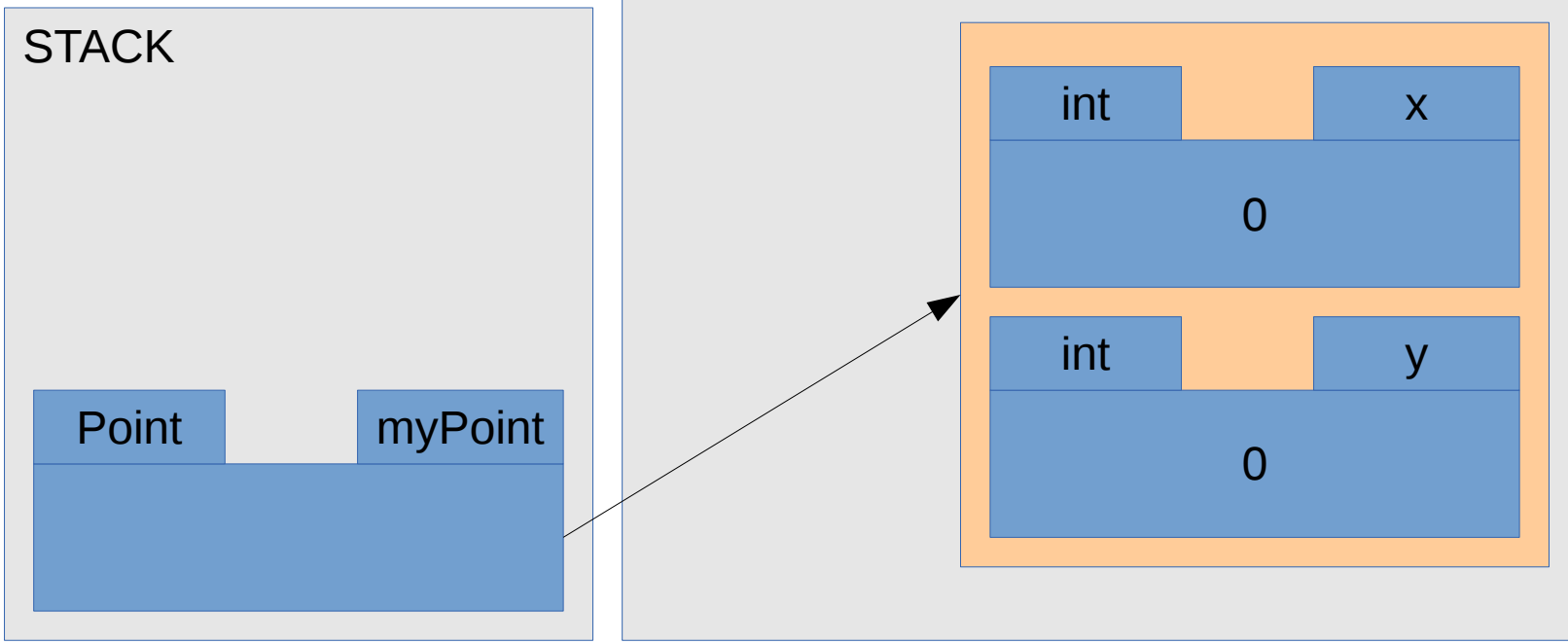
Field *x* is initialised to 0.



```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

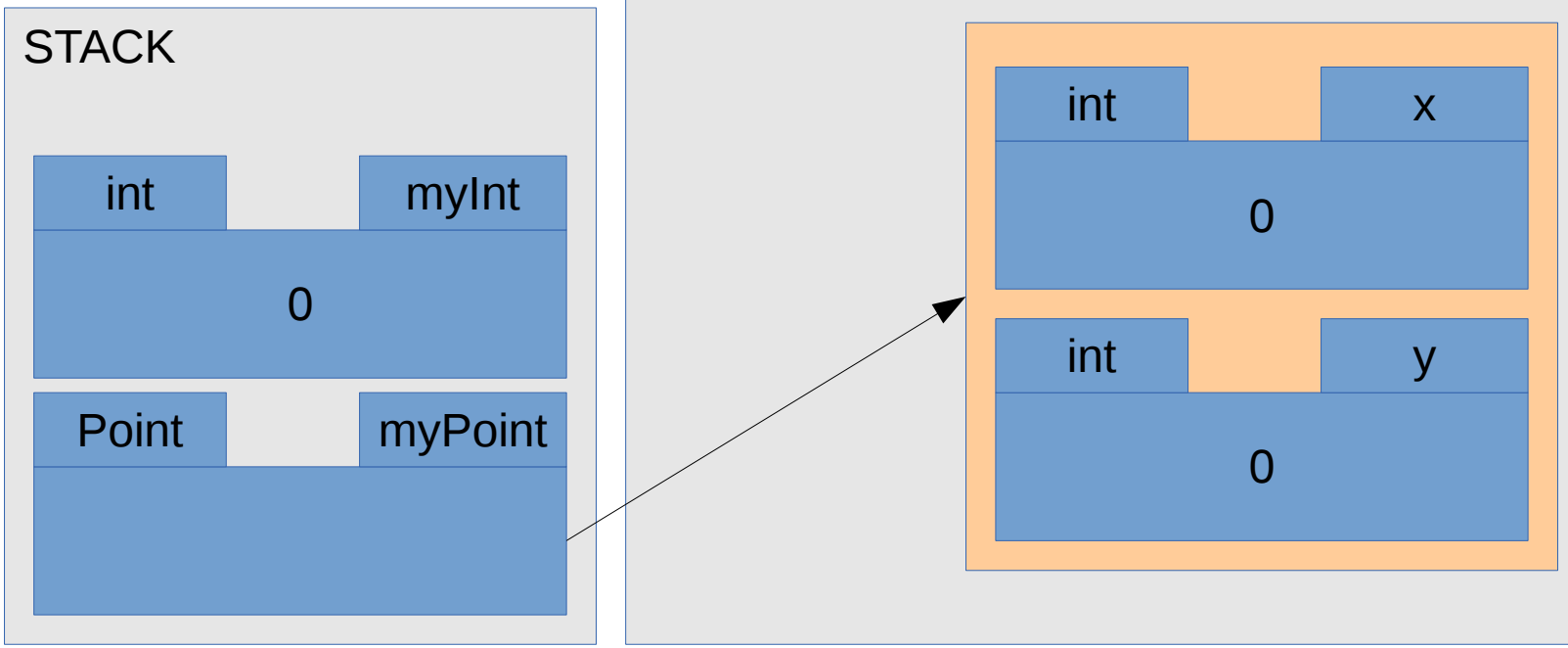


Field *y* is initialised to 0.



```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

An **int** is declared with name *myInt* and initialised to 0

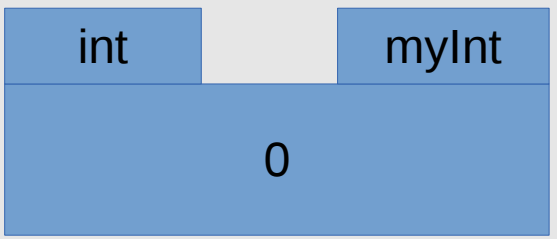


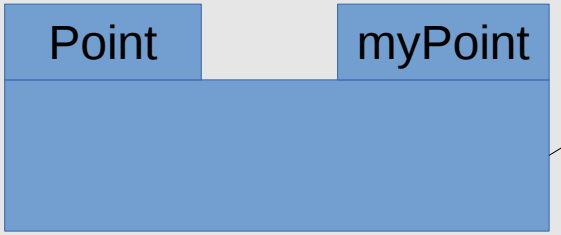
```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

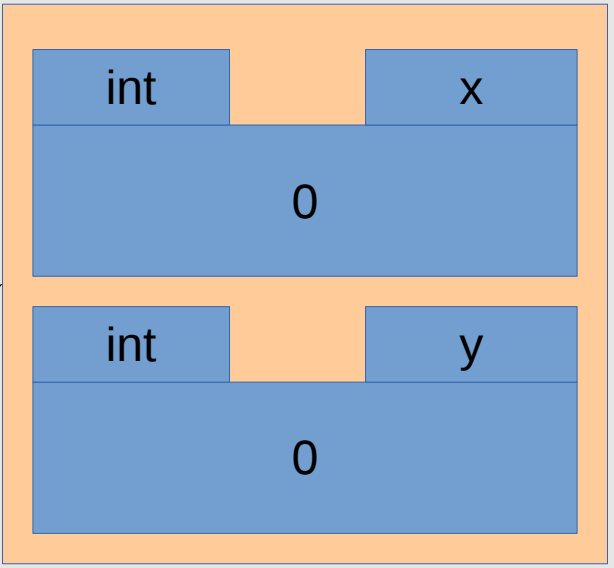
HEAP

Then we print several things

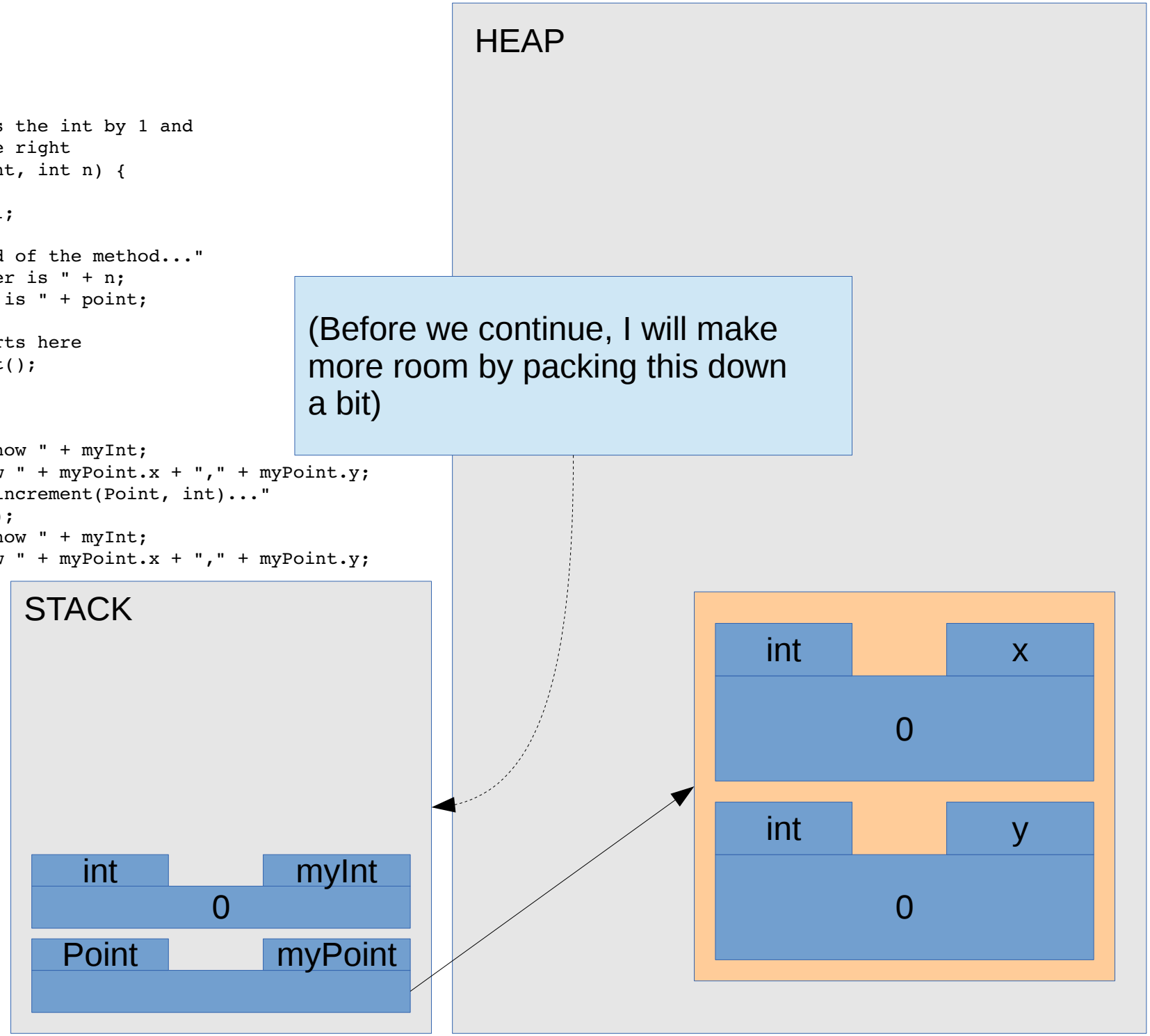
STACK





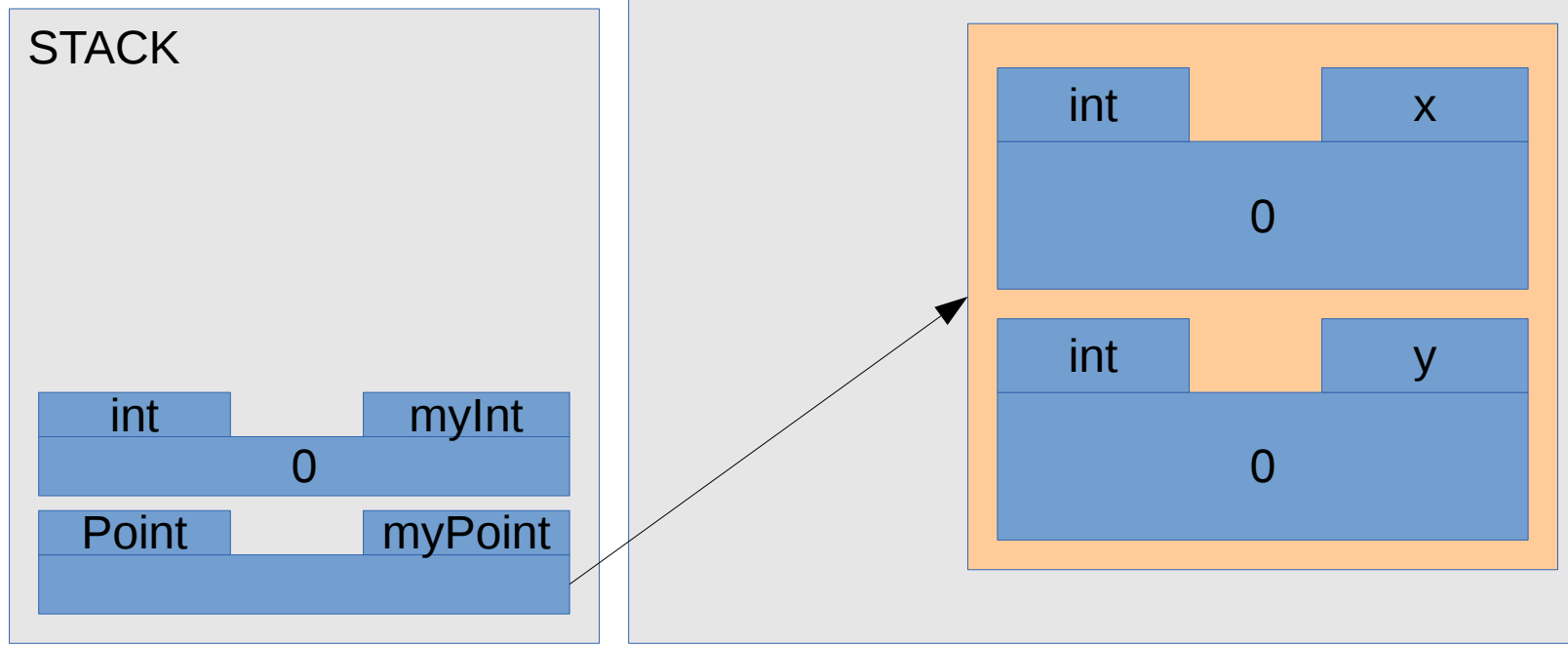


```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```



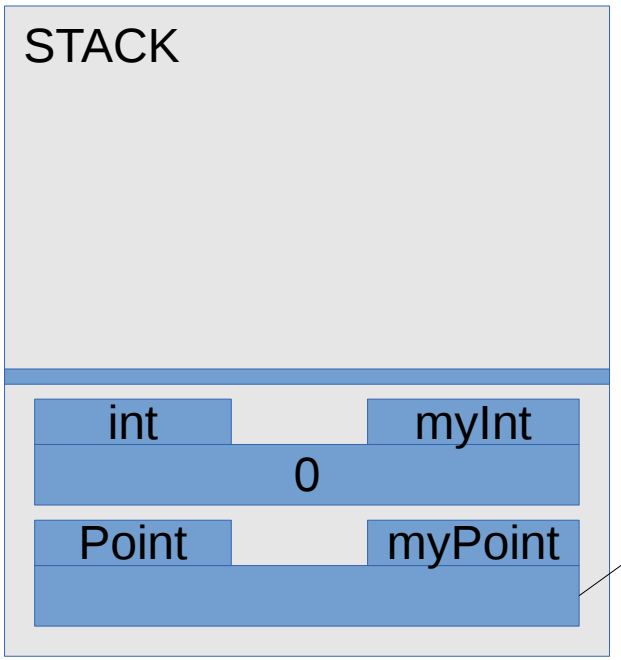
```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

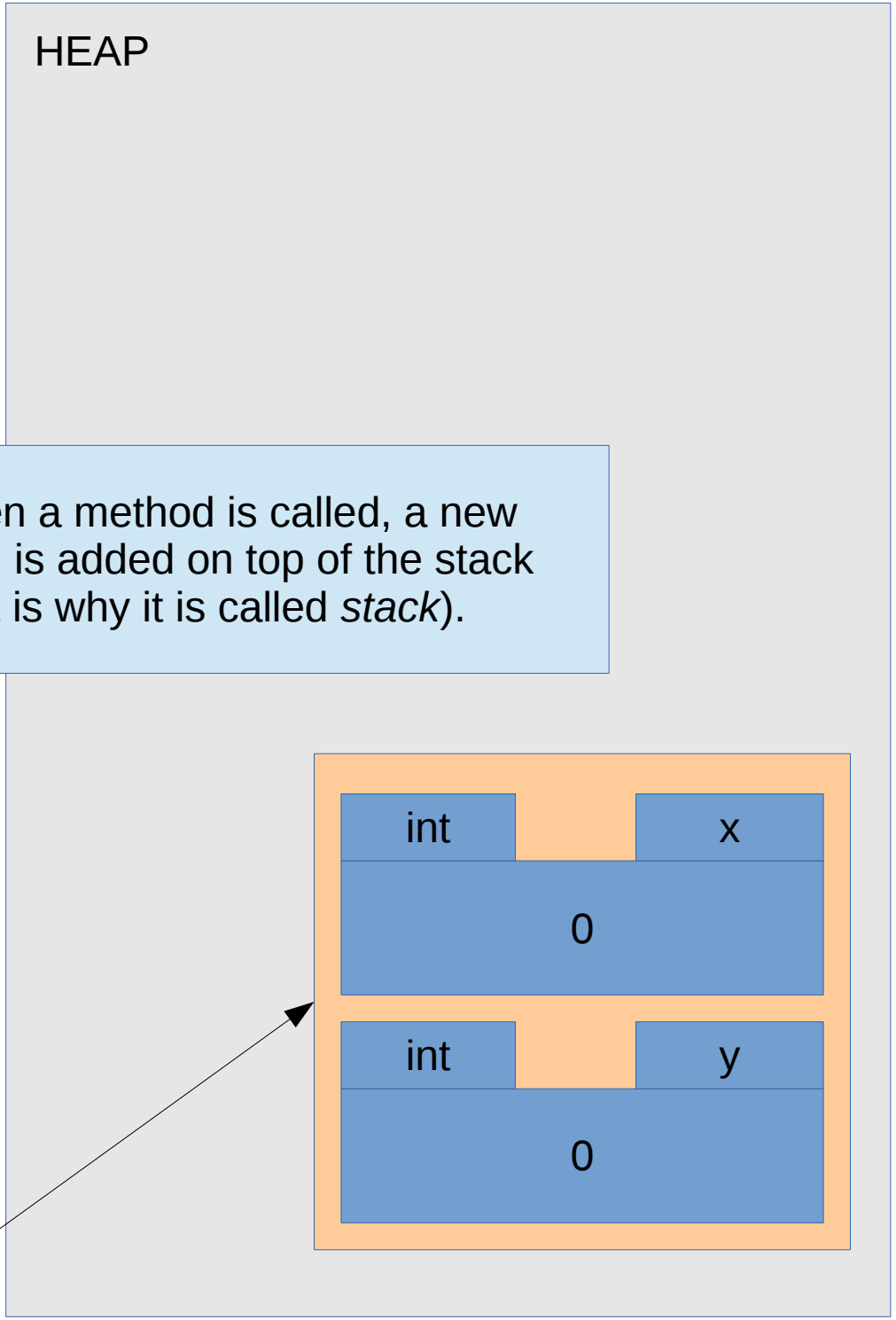
After printing, we call method increment(Point, int)



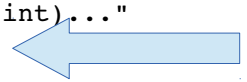
```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

When a method is called, a new level is added on top of the stack (that is why it is called *stack*).

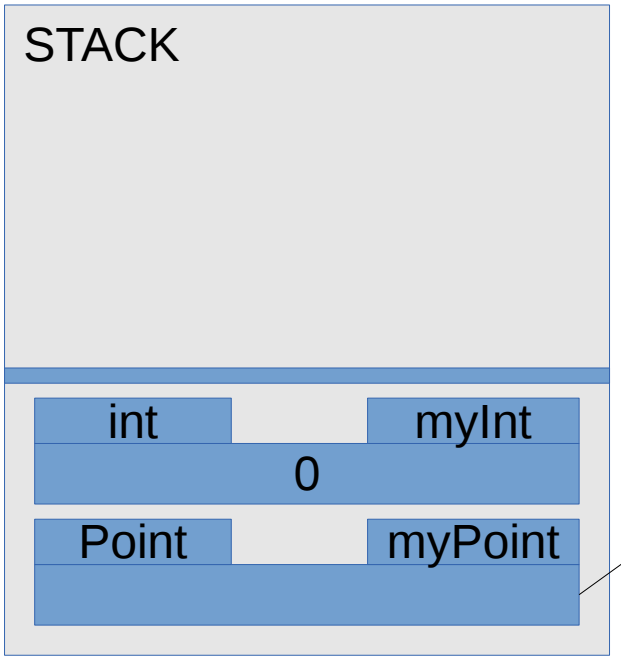


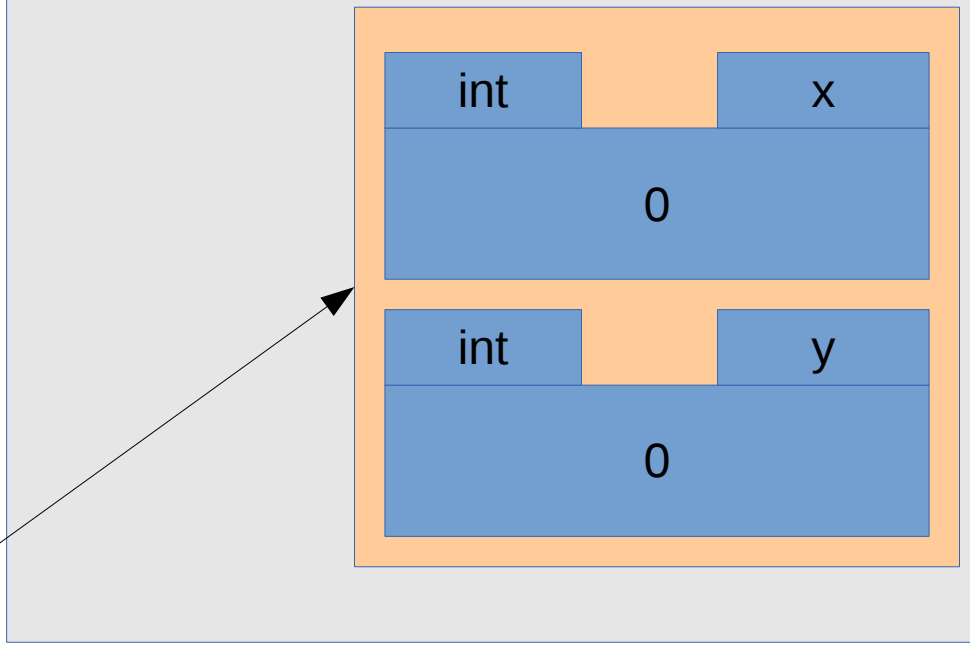


```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```



Then the method parameters are copied to the new level.

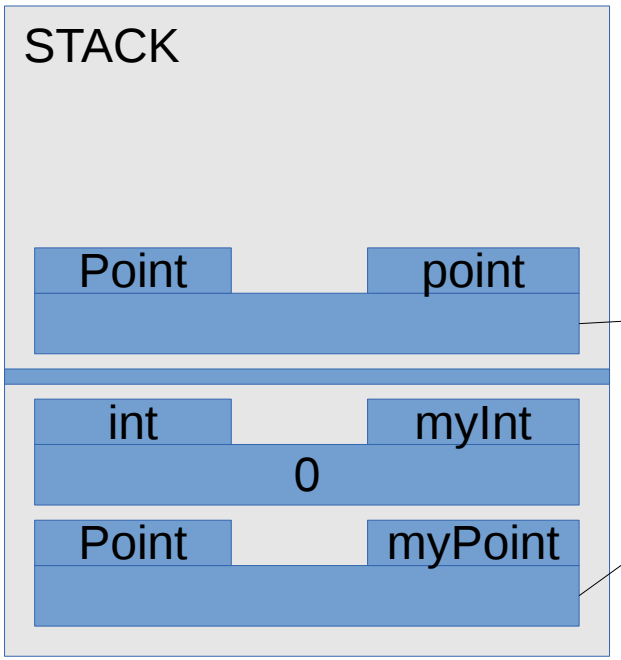


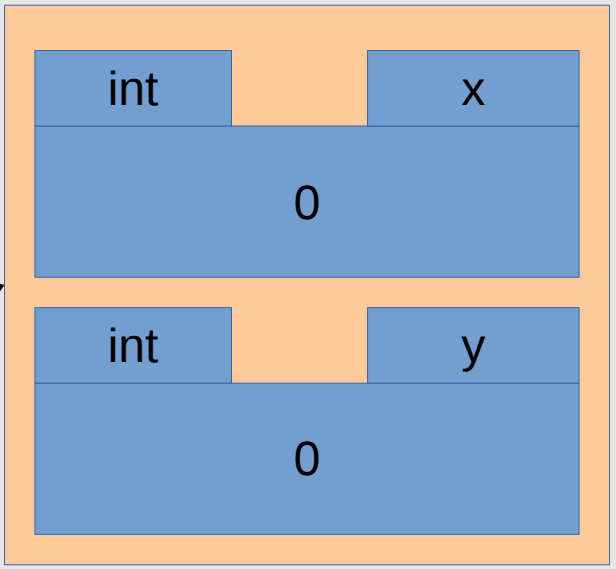


```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

HEAP

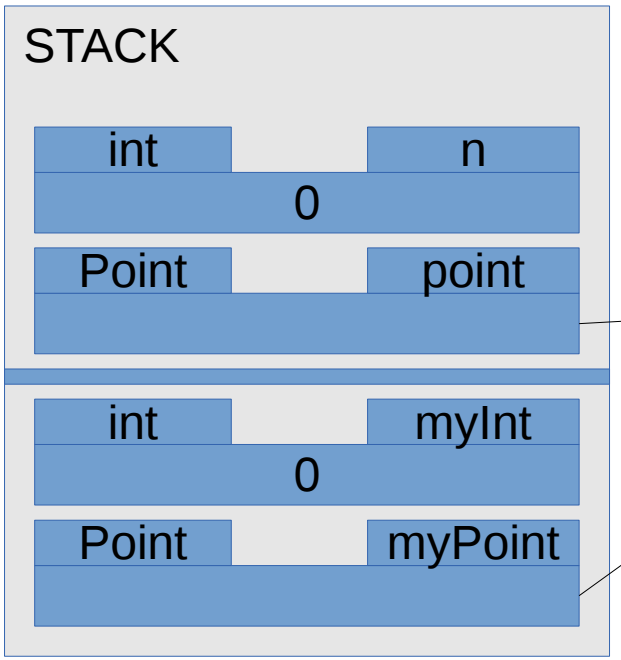
In this case, we need to copy the value of *myPoint* into *point*...

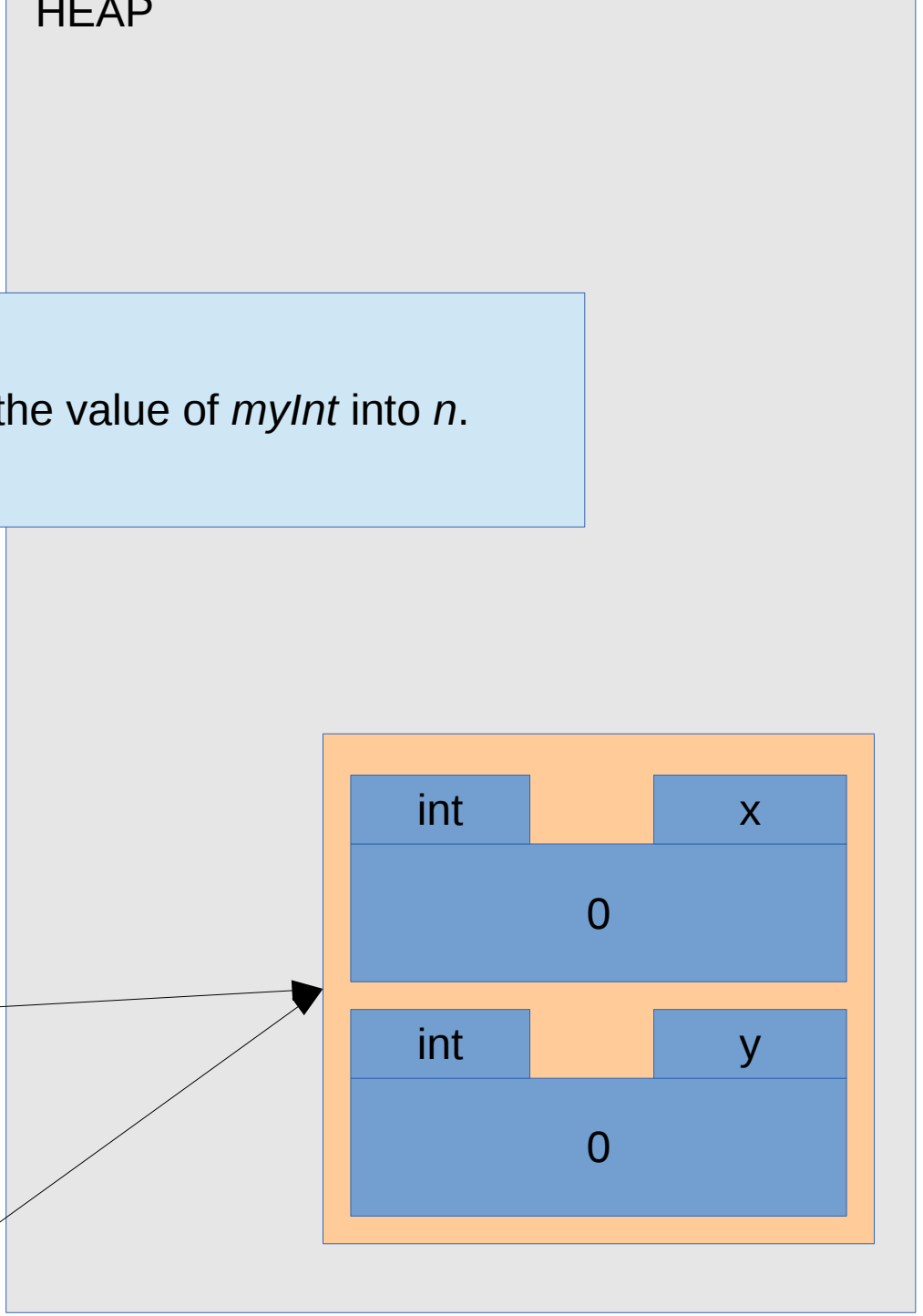




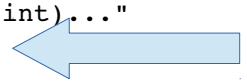
```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

...and the value of *myInt* into *n*.



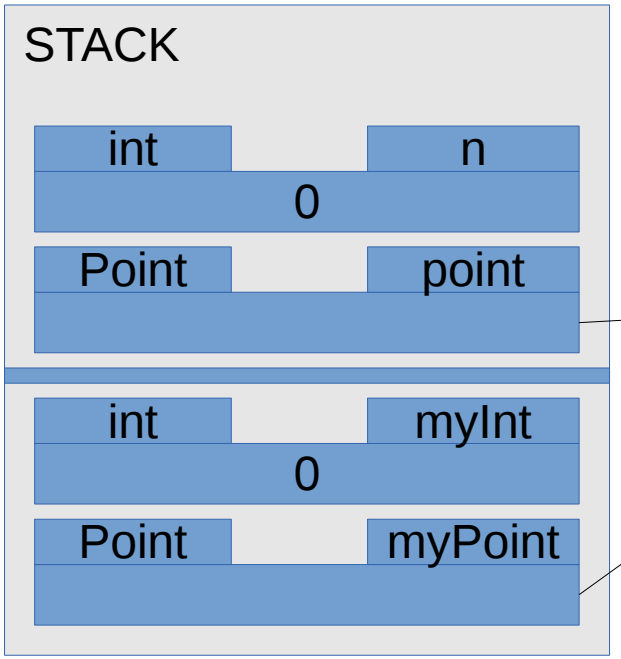


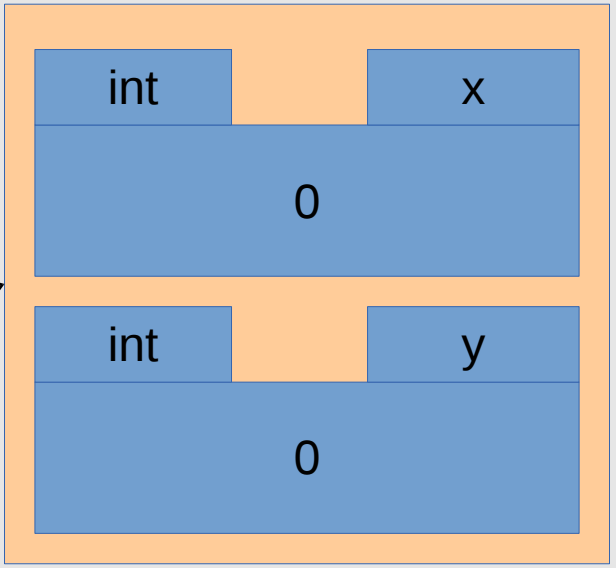
```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```



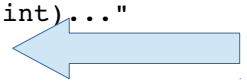
HEAP

Once the parameters are initialised, the execution of the program continues normally the method.

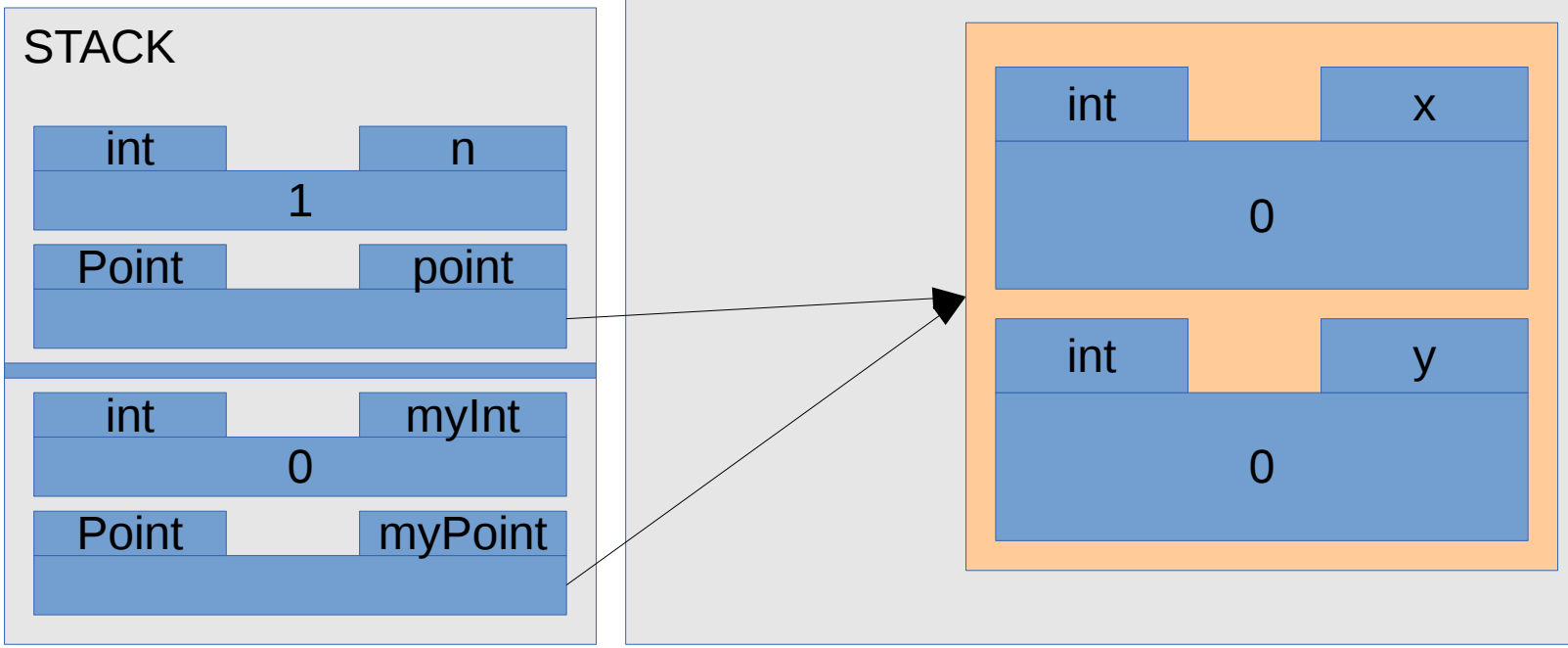




```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```



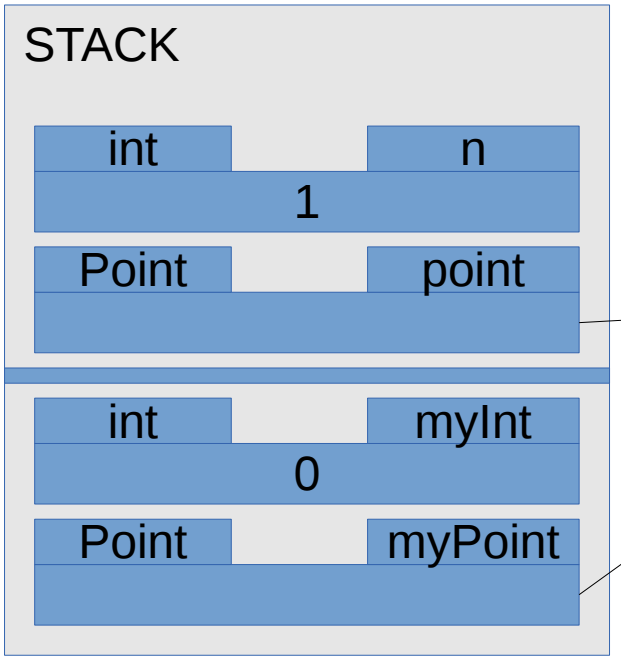
We increment *n* by 1...

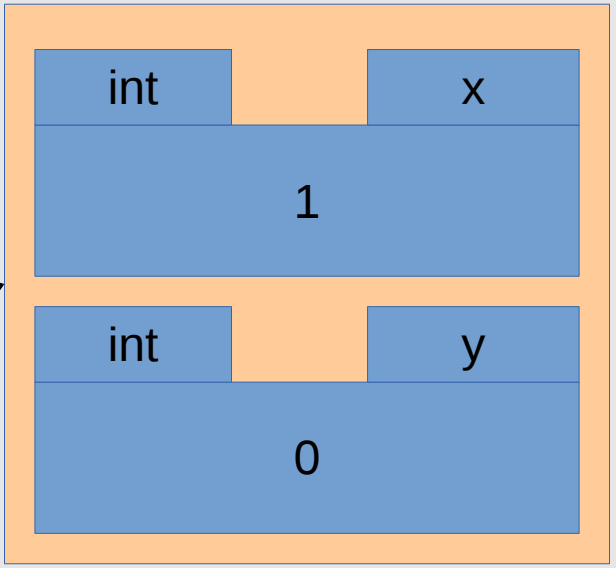


```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

HEAP

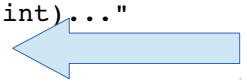
We increment point.x by 1, that is, variable *x* inside (variable) object *point*

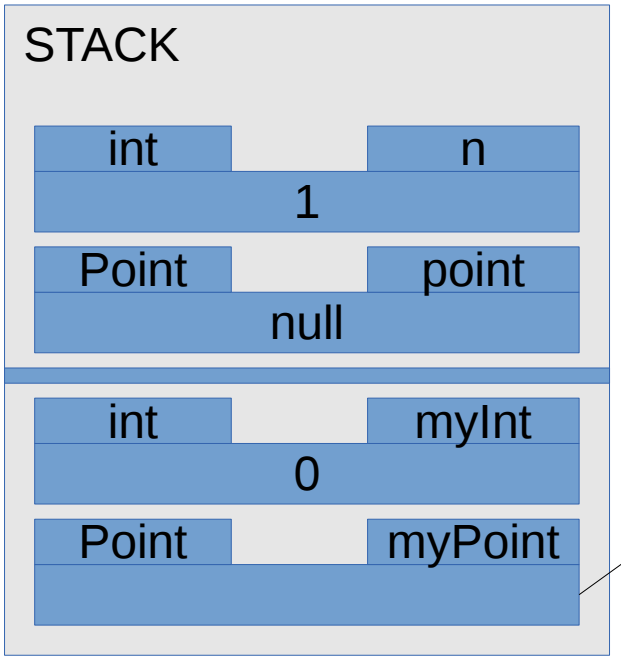


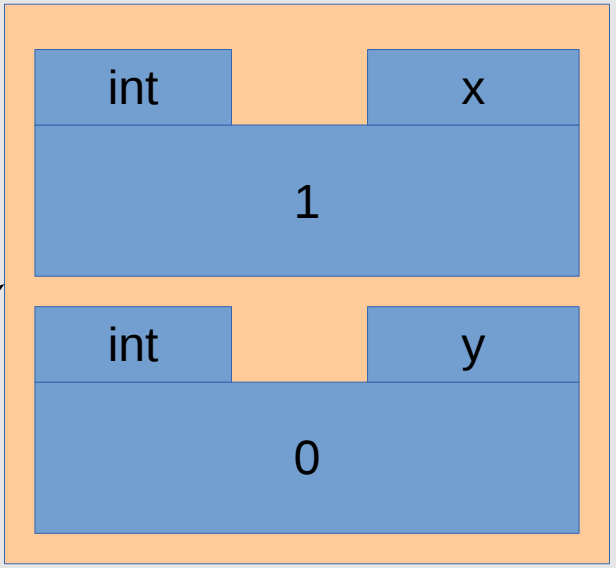


```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

We set point to null

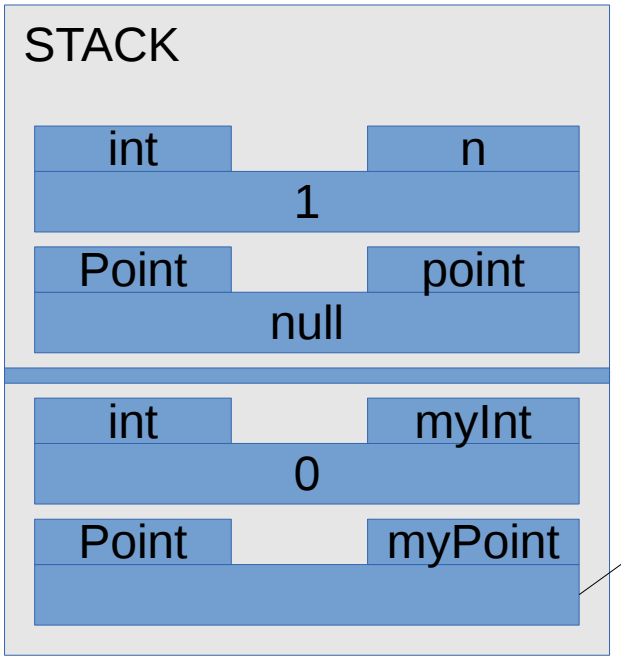


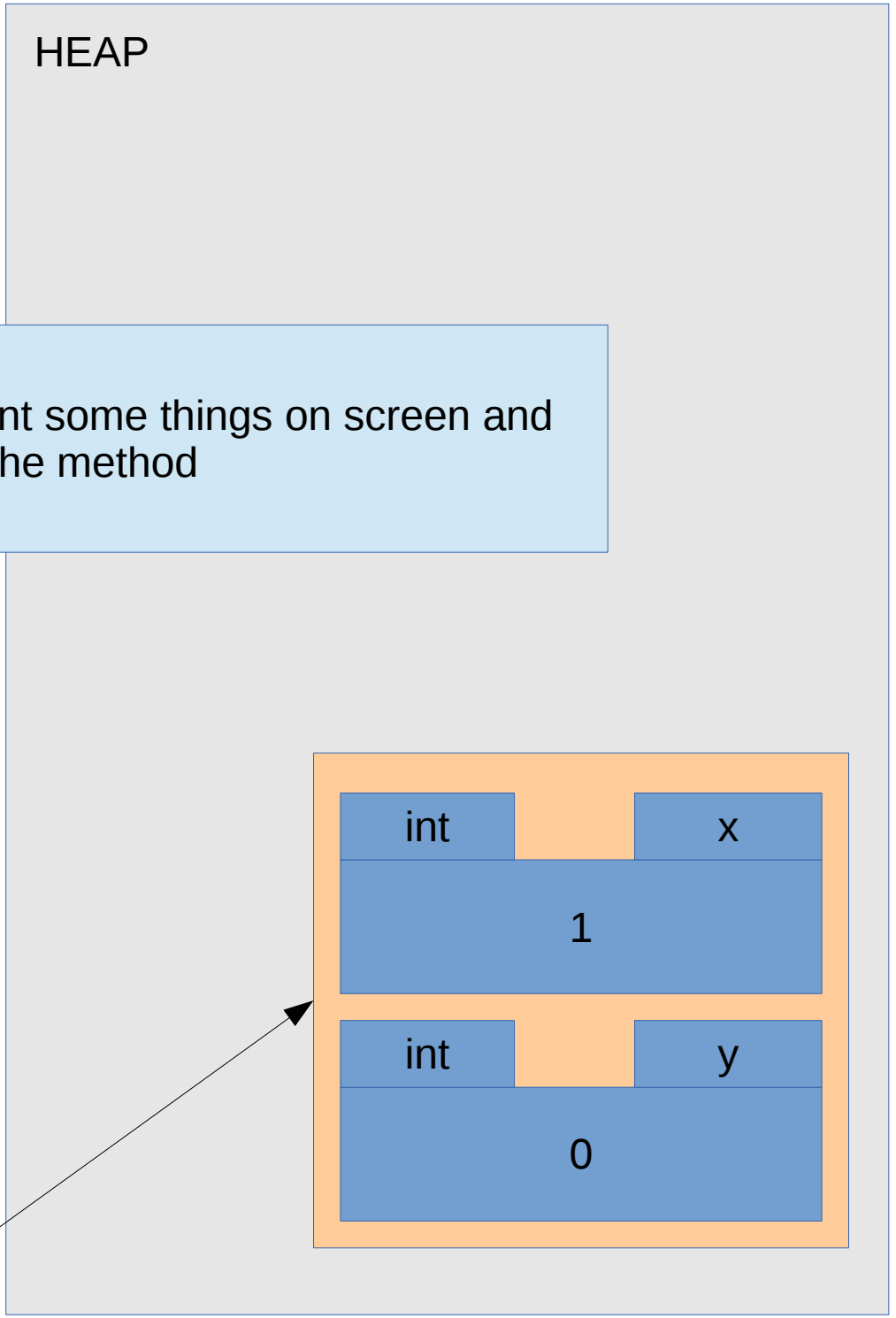




```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

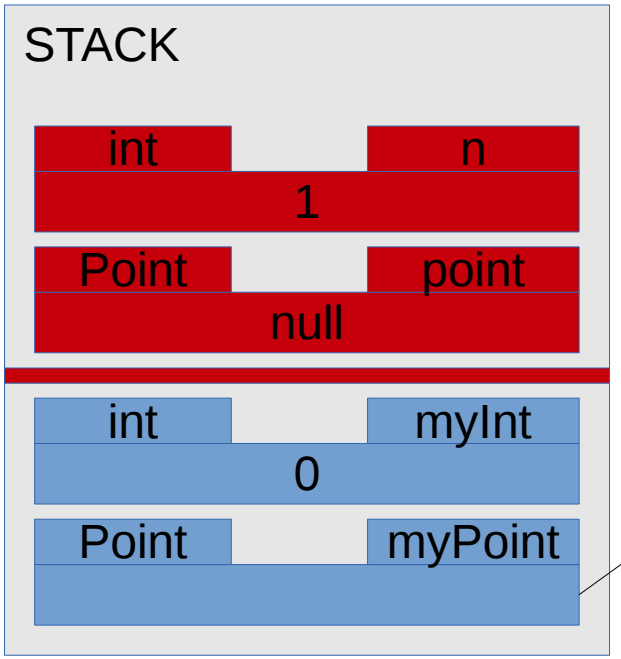
We print some things on screen and finish the method

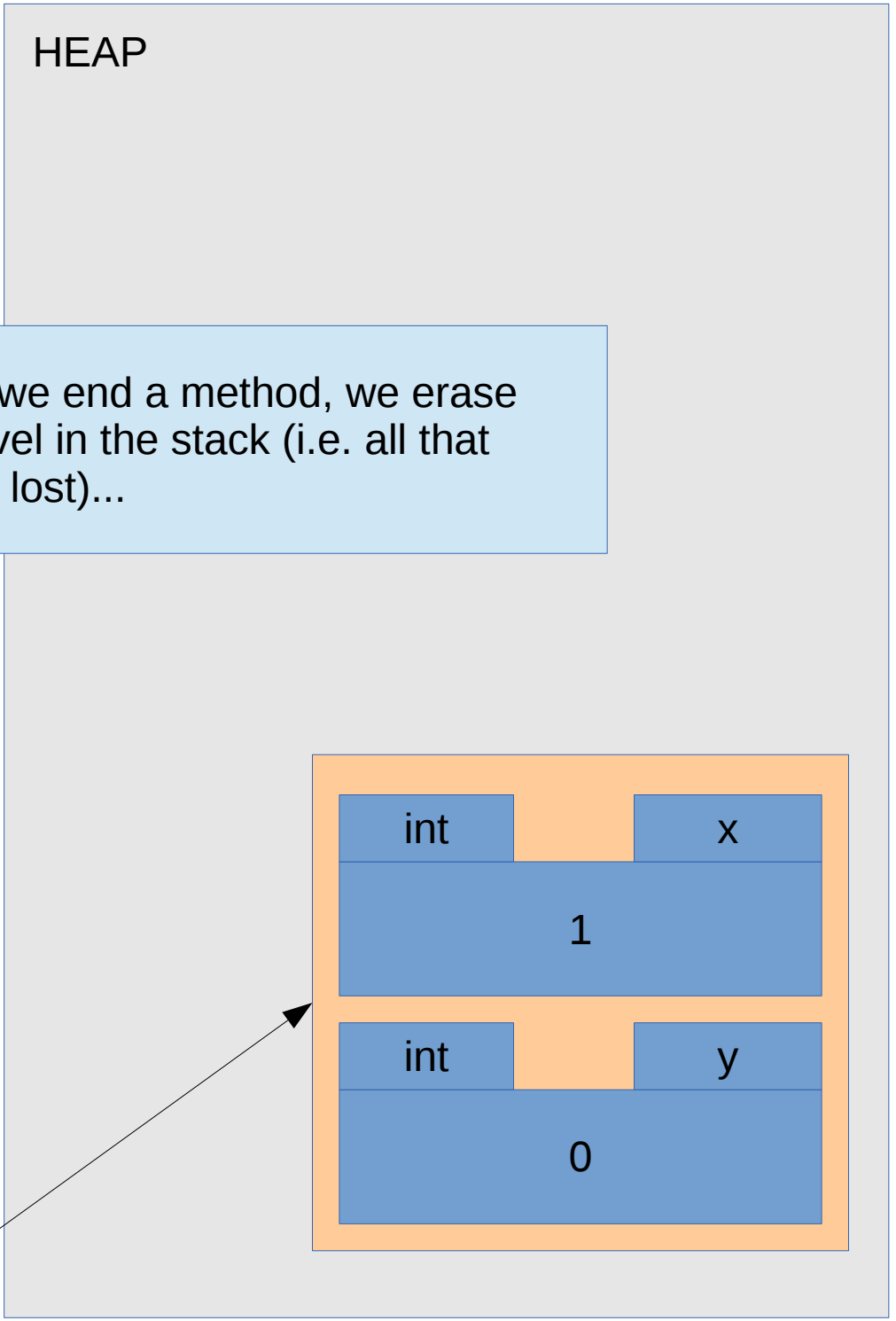




```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

When we end a method, we erase that level in the stack (i.e. all that data is lost)...



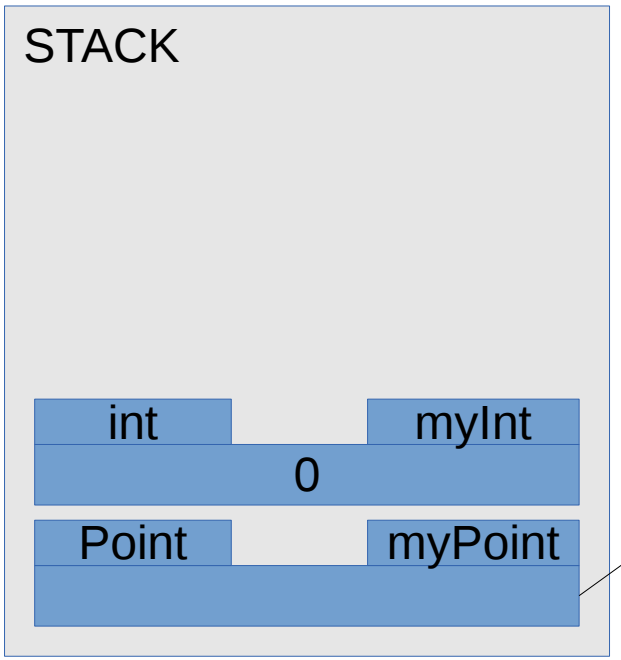


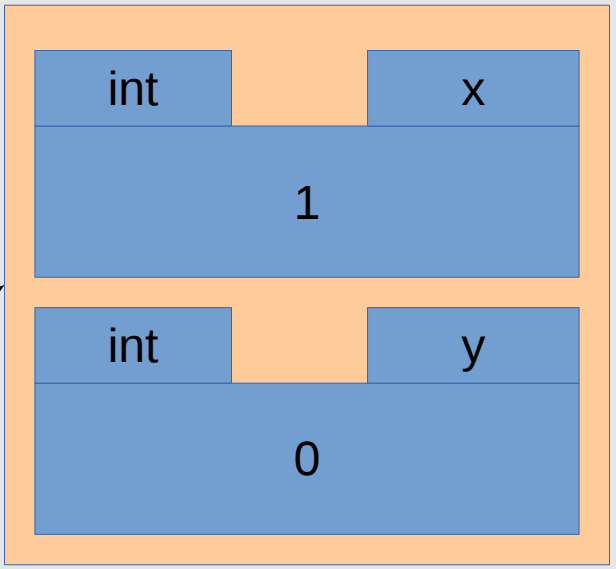
```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

HEAP

When we end a method, we erase that level in the stack (i.e. all that data is lost)...

STACK



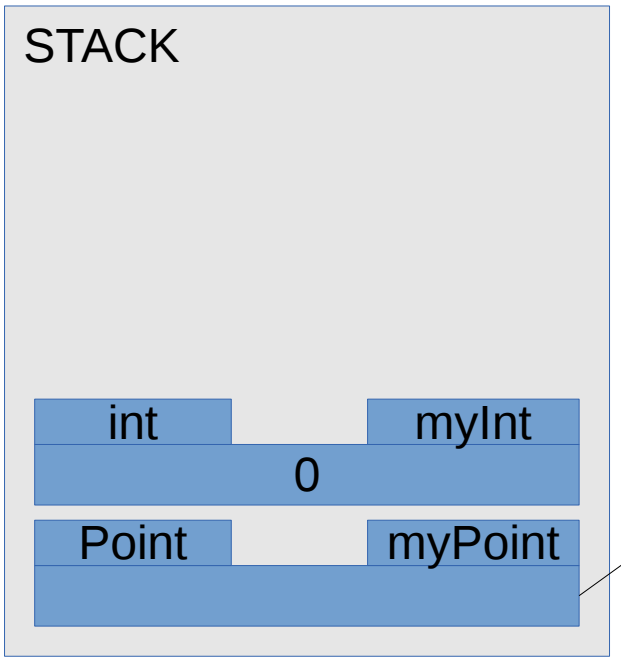


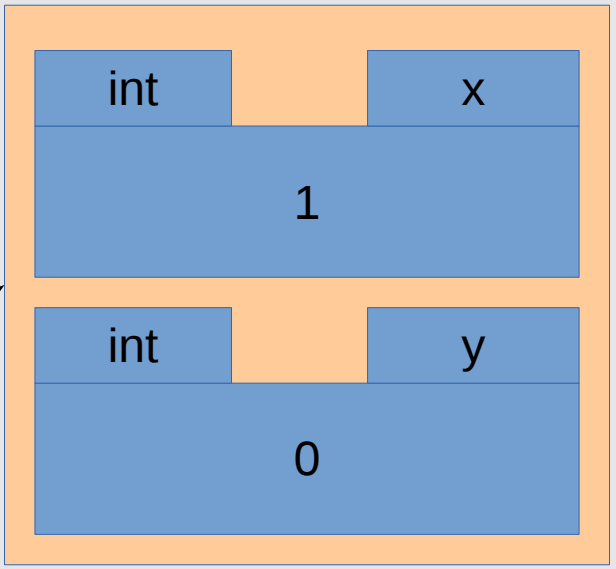
```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```

HEAP

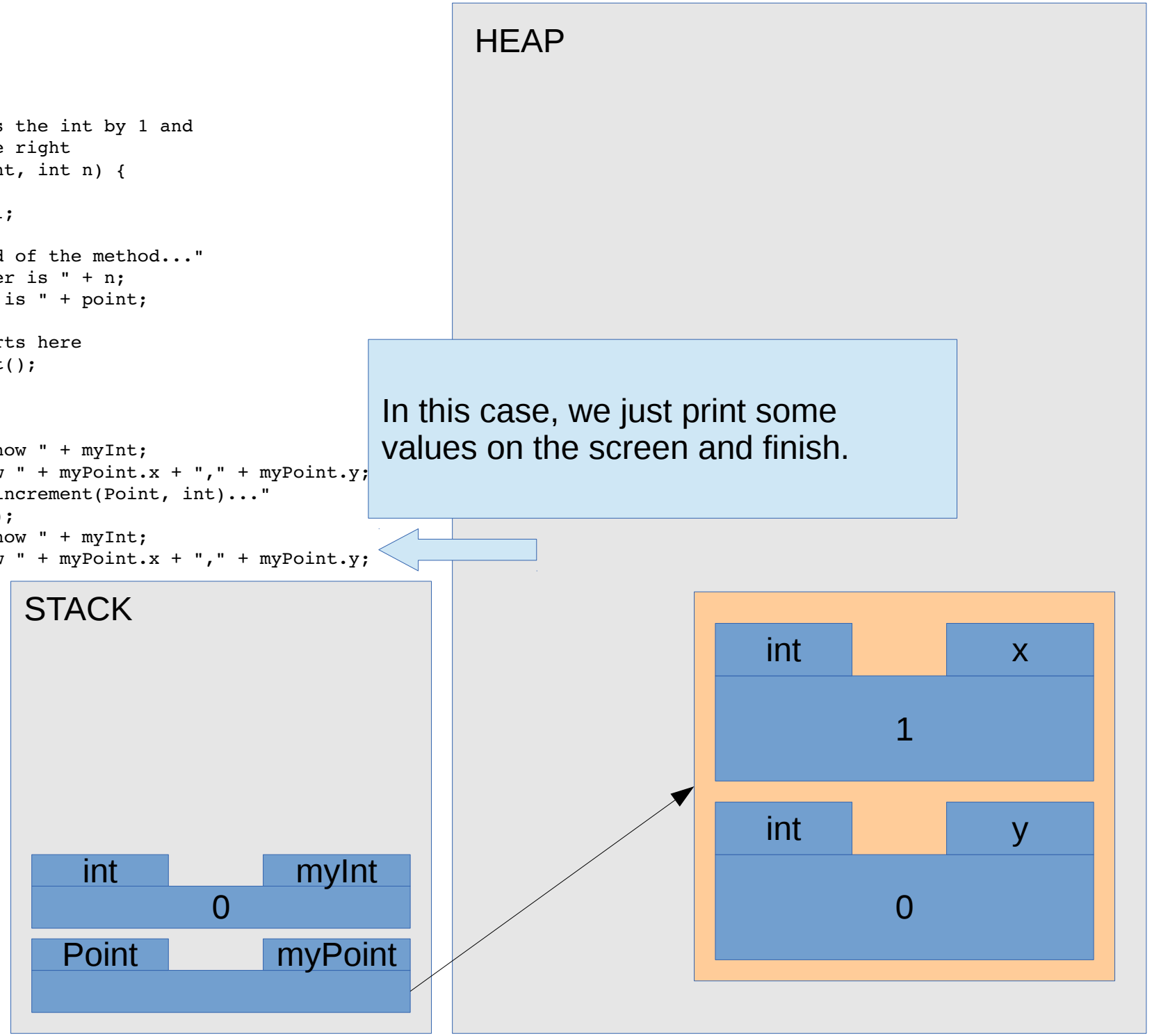
...and the execution of the program continues from the point where the method was called.

STACK

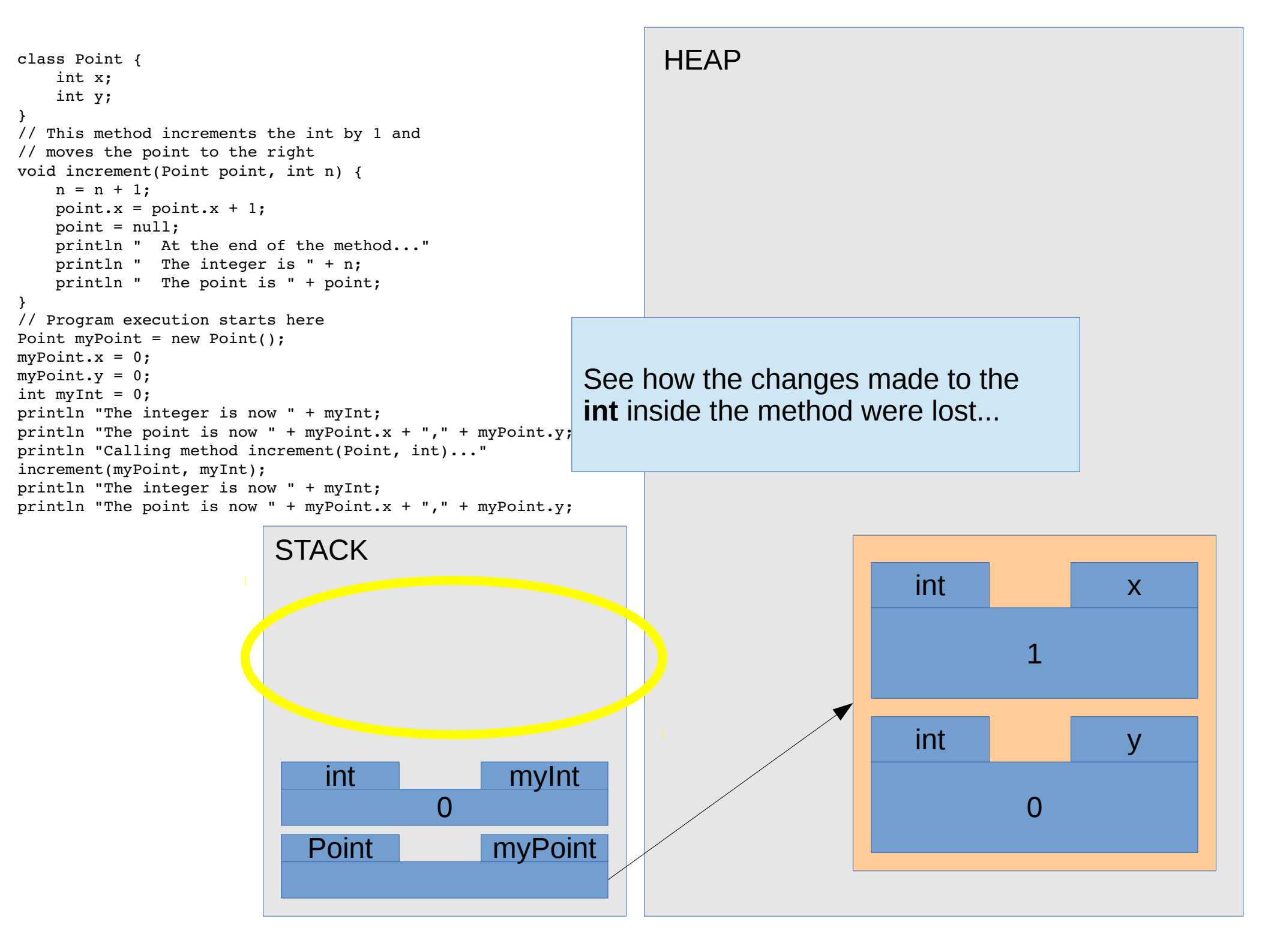




```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```



```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```



```
class Point {
    int x;
    int y;
}
// This method increments the int by 1 and
// moves the point to the right
void increment(Point point, int n) {
    n = n + 1;
    point.x = point.x + 1;
    point = null;
    println "  At the end of the method..."
    println "  The integer is " + n;
    println "  The point is " + point;
}
// Program execution starts here
Point myPoint = new Point();
myPoint.x = 0;
myPoint.y = 0;
int myInt = 0;
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
println "Calling method increment(Point, int)..."
increment(myPoint, myInt);
println "The integer is now " + myInt;
println "The point is now " + myPoint.x + "," + myPoint.y;
```