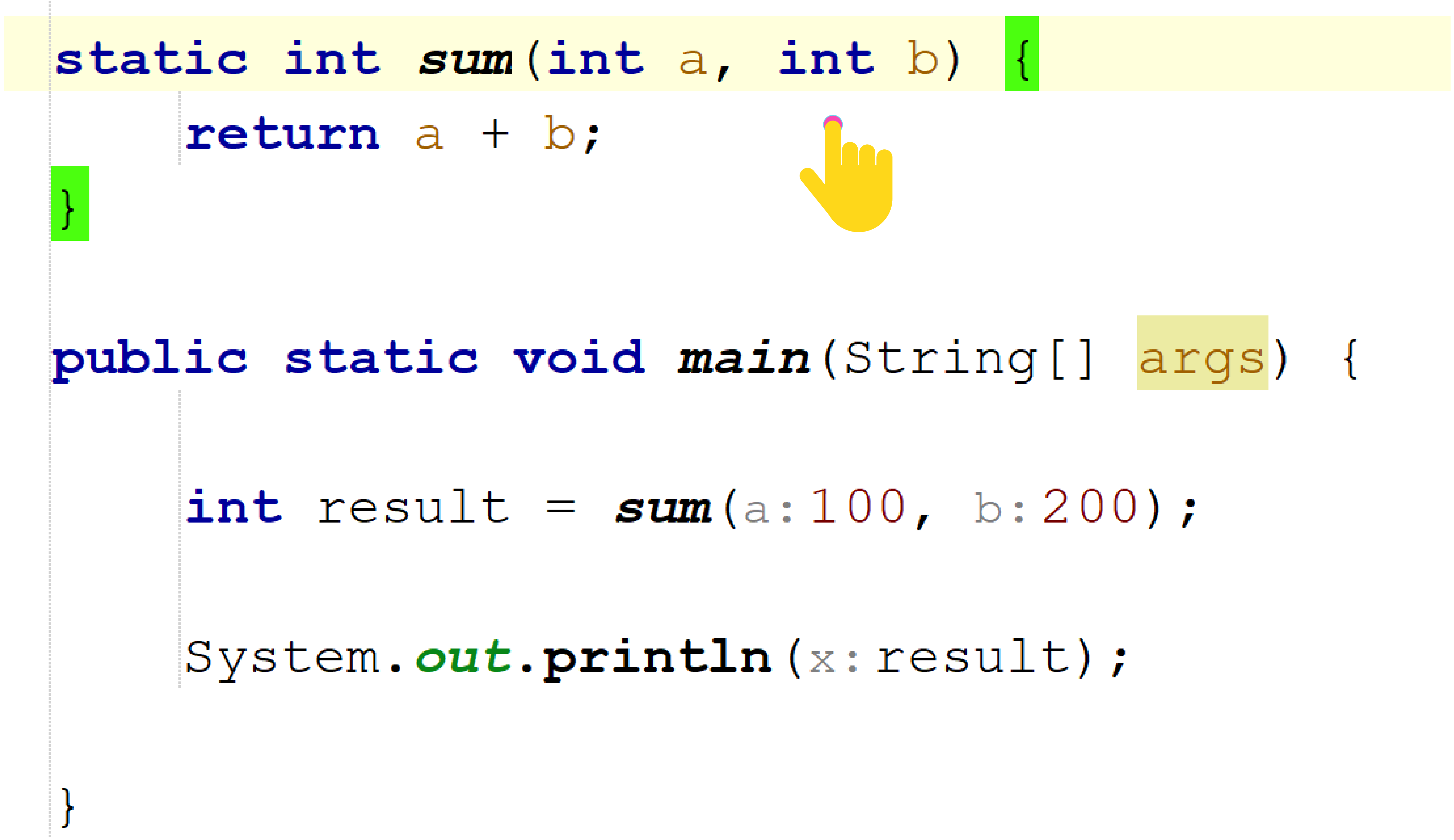
Fundamentos de Java

Parte 6

Presenta

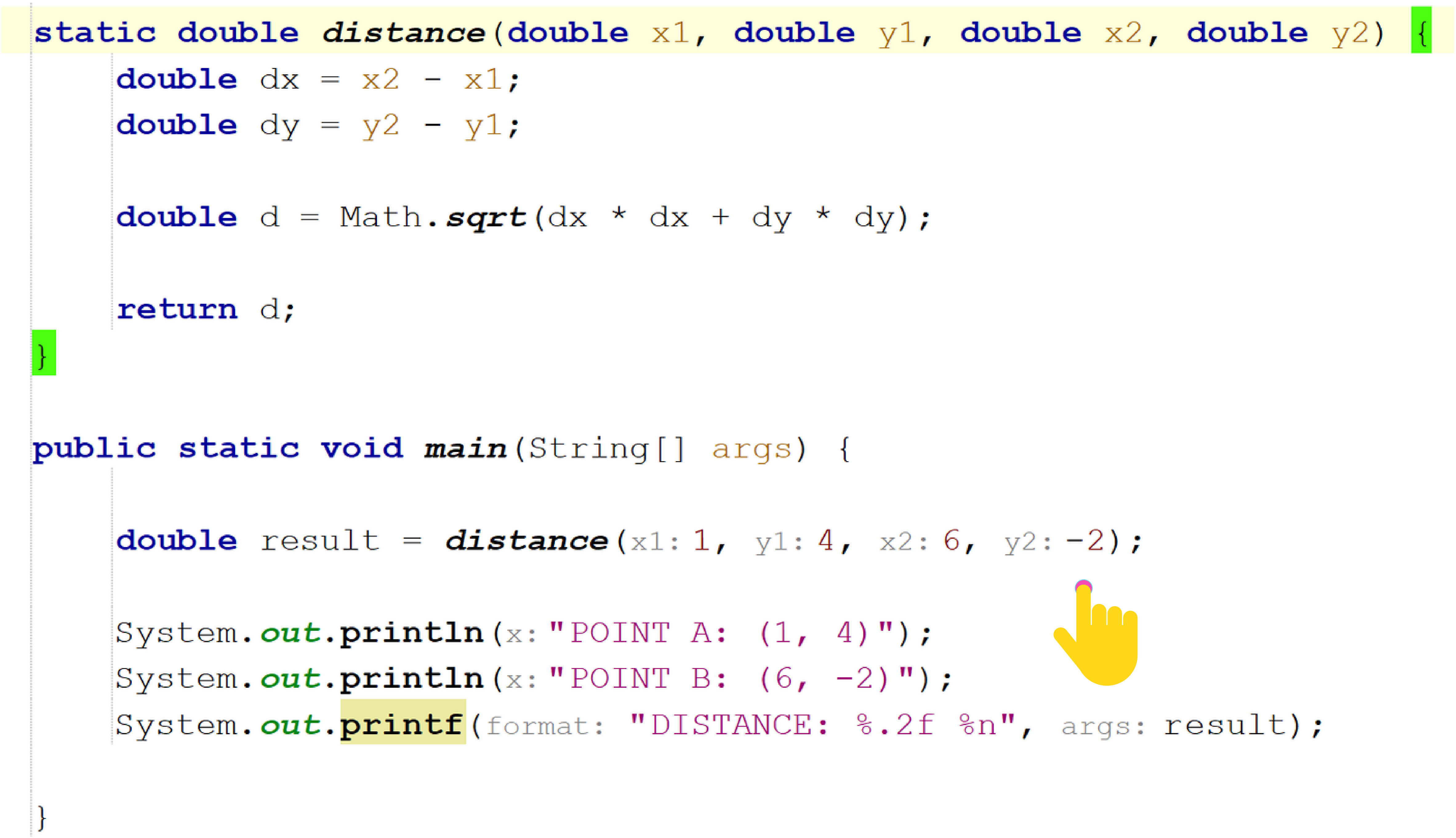
Alan Badillo Salas

Marzo 2023



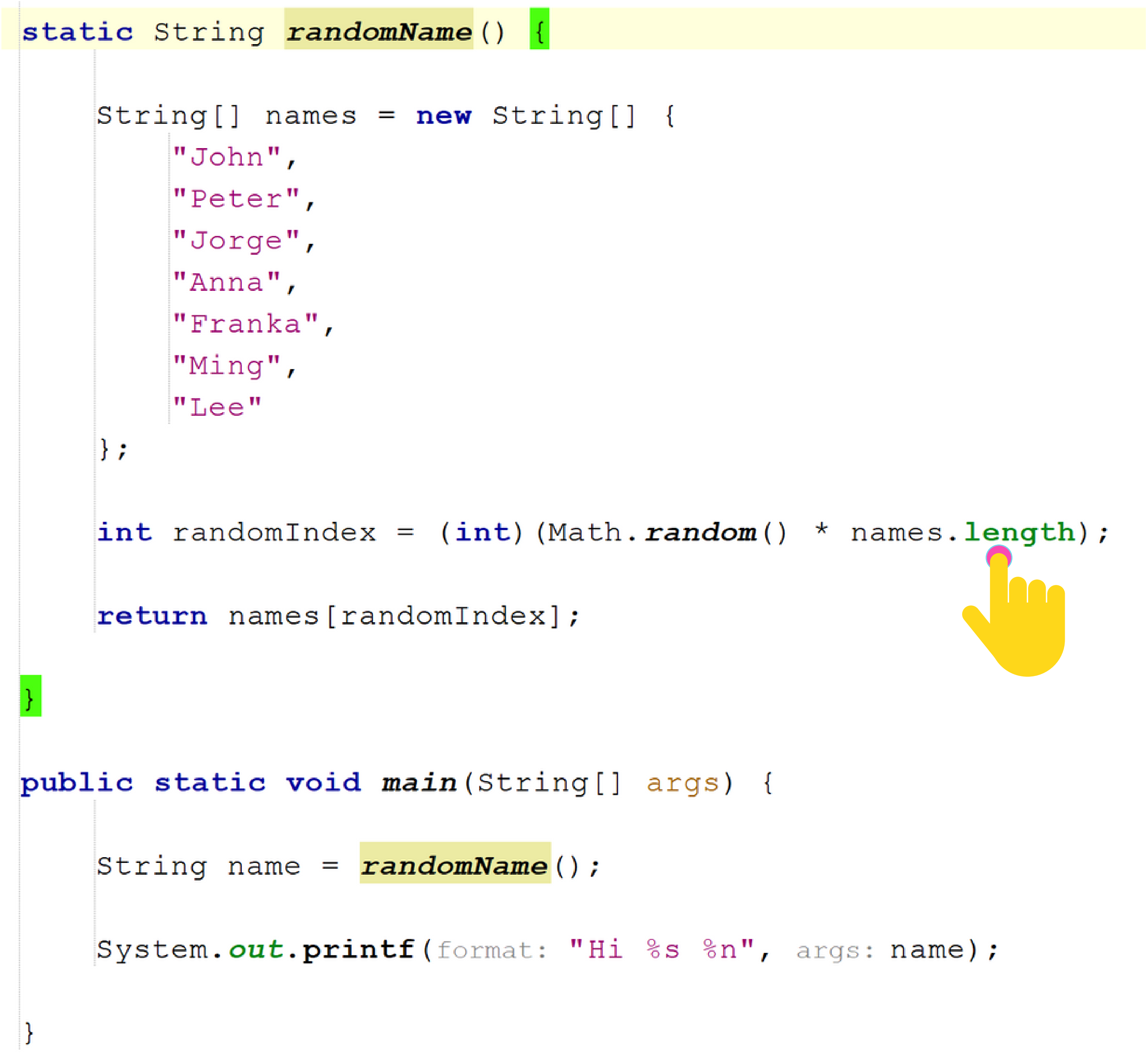
```java
static int sum(int a, int b) {
    return a + b;
}

public static void main(String[] args) {

    int result = sum(a:100, b:200);

    System.out.println(x:result);

}
```

```
run:
300
BUILD SUCCESSFUL (total time: 7 seconds)
```

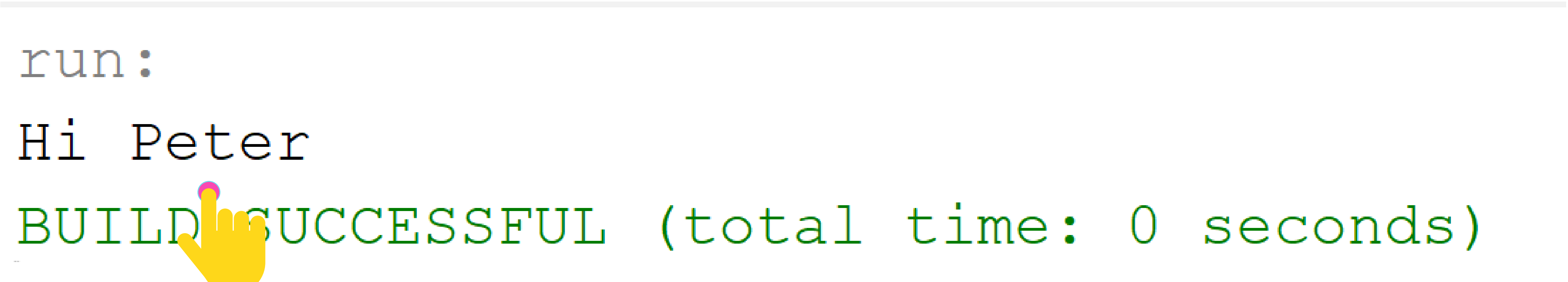


```java
static double distance(double x1, double y1, double x2, double y2) {
    double dx = x2 - x1;
    double dy = y2 - y1;

    double d = Math.sqrt(dx * dx + dy * dy);

    return d;
}

public static void main(String[] args) {

    double result = distance(x1: 1, y1: 4, x2: 6, y2: -2);

    System.out.println(x: "POINT A: (1, 4)");
    System.out.println(x: "POINT B: (6, -2)");
    System.out.printf(format: "DISTANCE: %.2f %n", args: result);

}
```

```
run:
POINT A: (1, 4)
POINT B: (6, -2)
DISTANCE: 7.81
BUILD SUCCESSFUL (total time: 0 seconds)
```

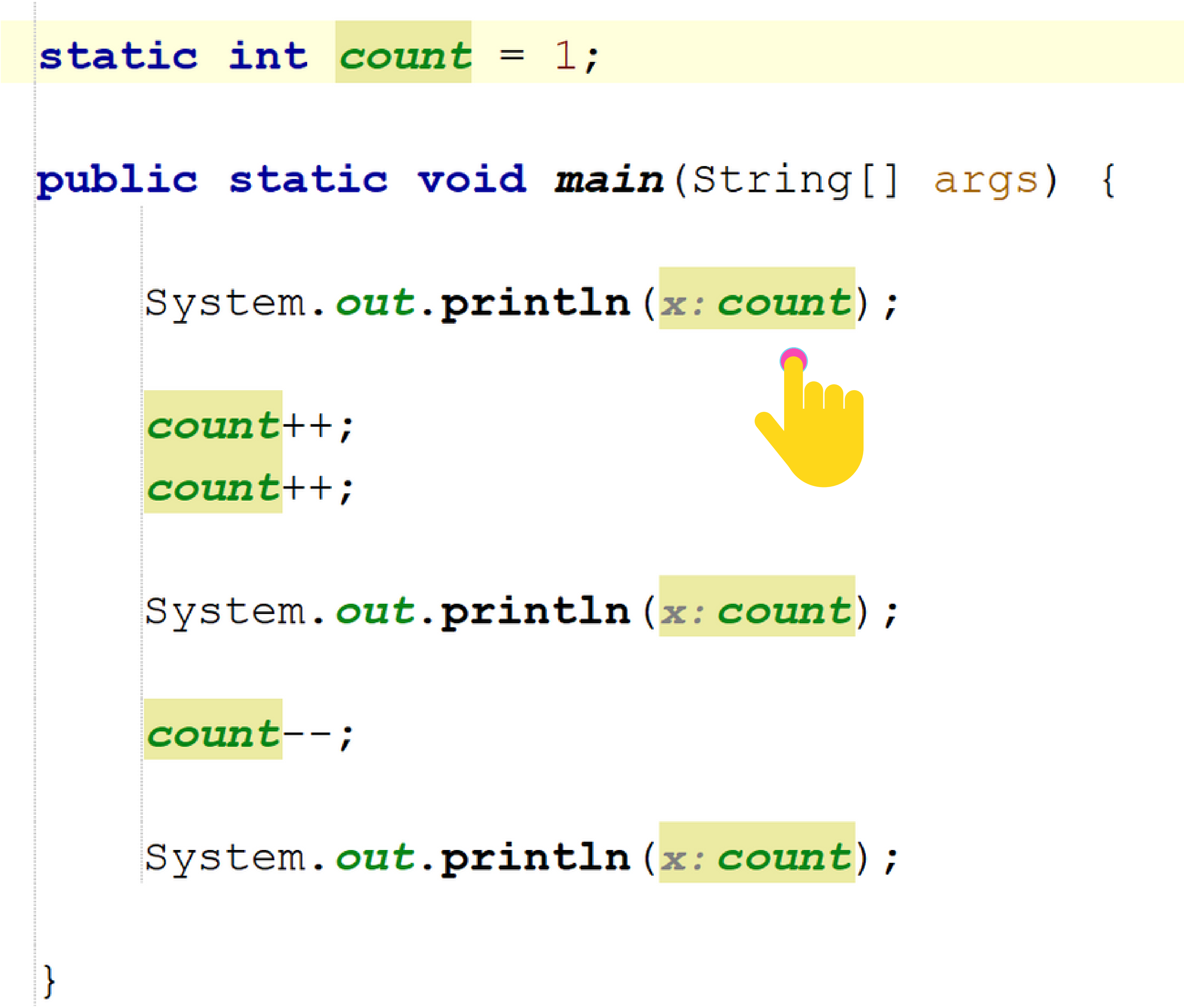


```
static String randomName() {

    String[] names = new String[] {
        "John",
        "Peter",
        "Jorge",
        "Anna",
        "Franka",
        "Ming",
        "Lee"
    };

    int randomIndex = (int)(Math.random() * names.length);

    return names[randomIndex];

}

public static void main(String[] args) {

    String name = randomName();

    System.out.printf(format: "Hi %s %n", args: name);

}
```



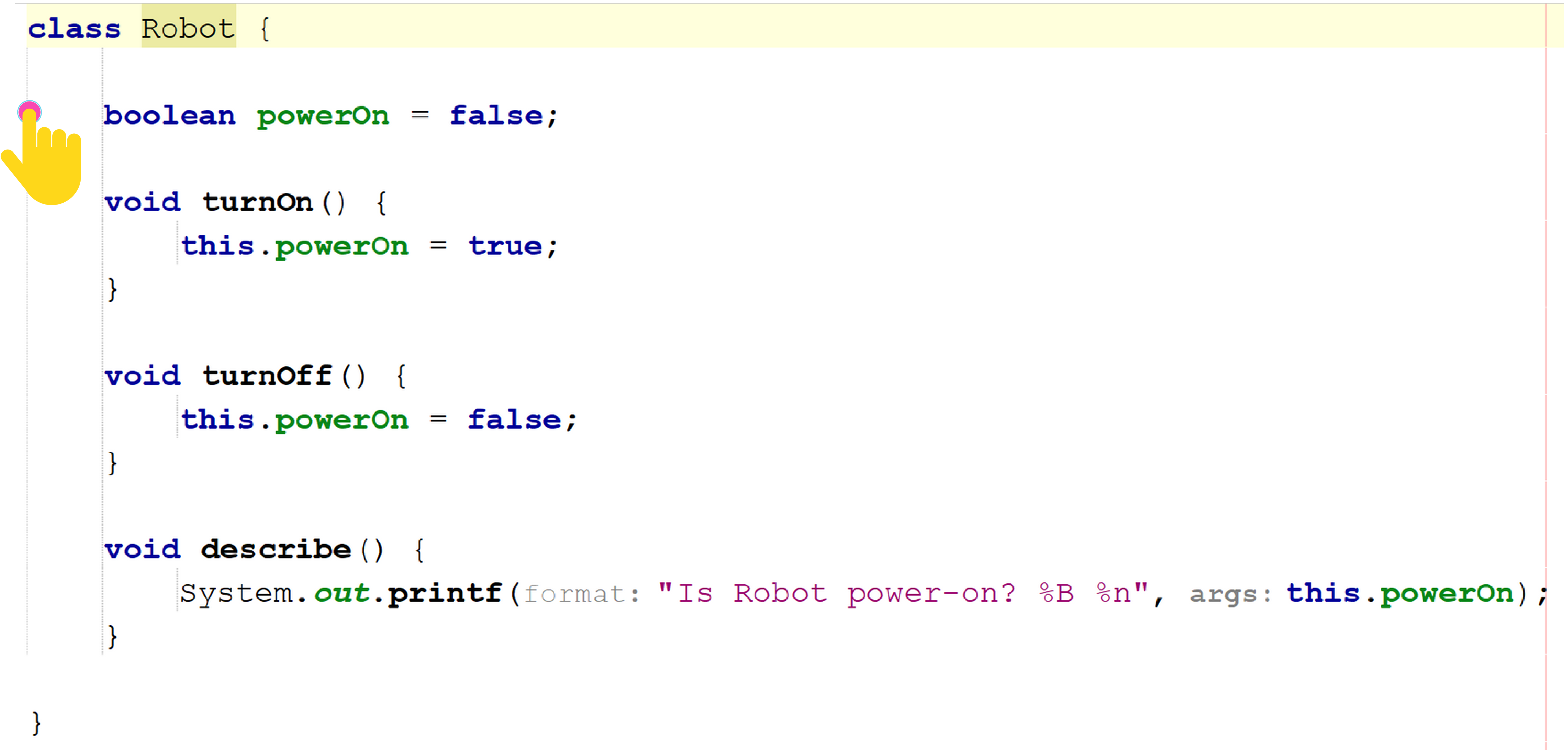
```
run:
Hi Peter
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
run:
Hi Ming
BUILD SUCCESSFUL (total time: 0 seconds)
```

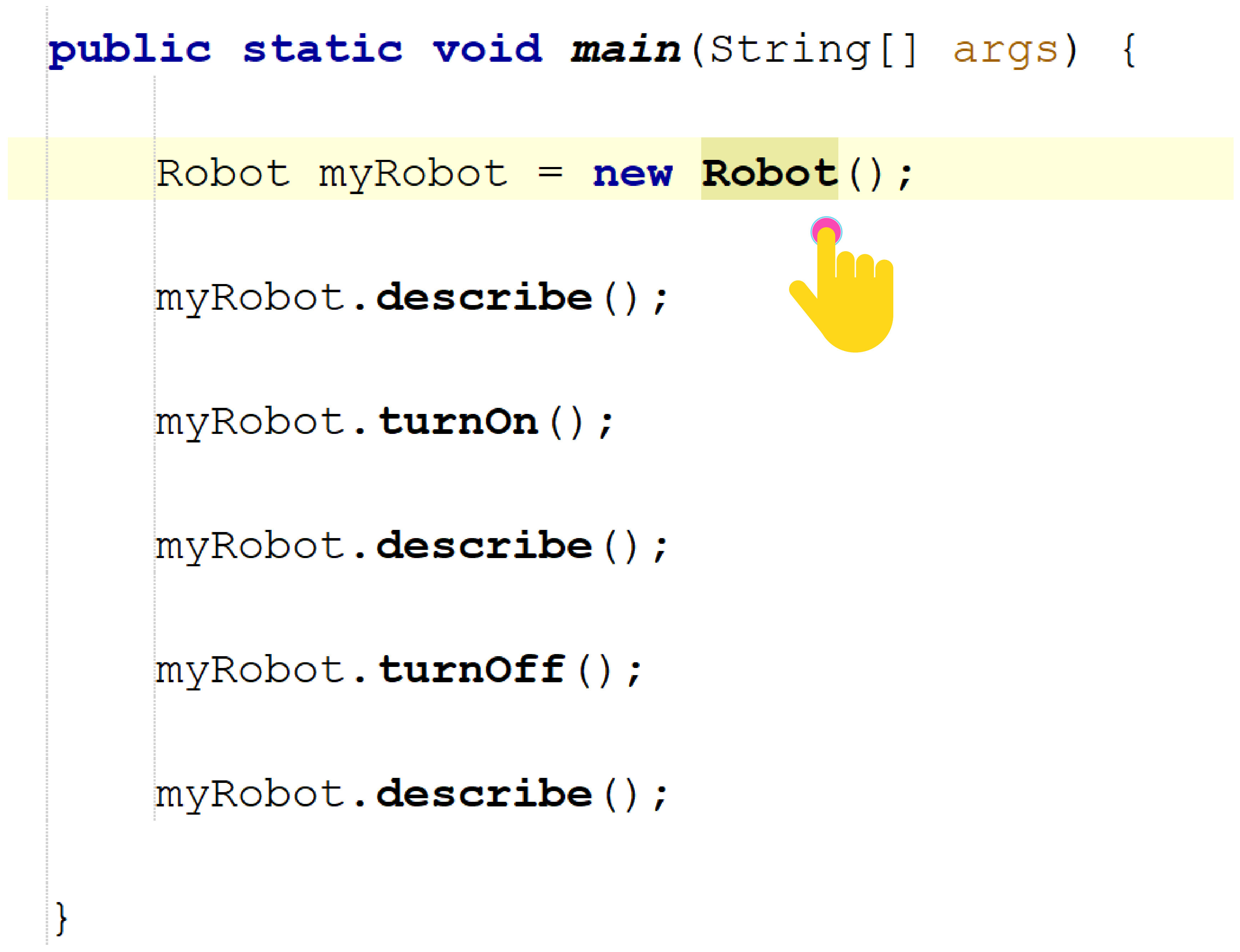


```
static int count = 1;

public static void main(String[] args) {

    System.out.println(x: count);

    count++;
    count++;

    System.out.println(x: count);

    count--;

    System.out.println(x: count);

}
```

```
run:
1
3
2
BUILD SUCCESSFUL (total time: 0 seconds)
```

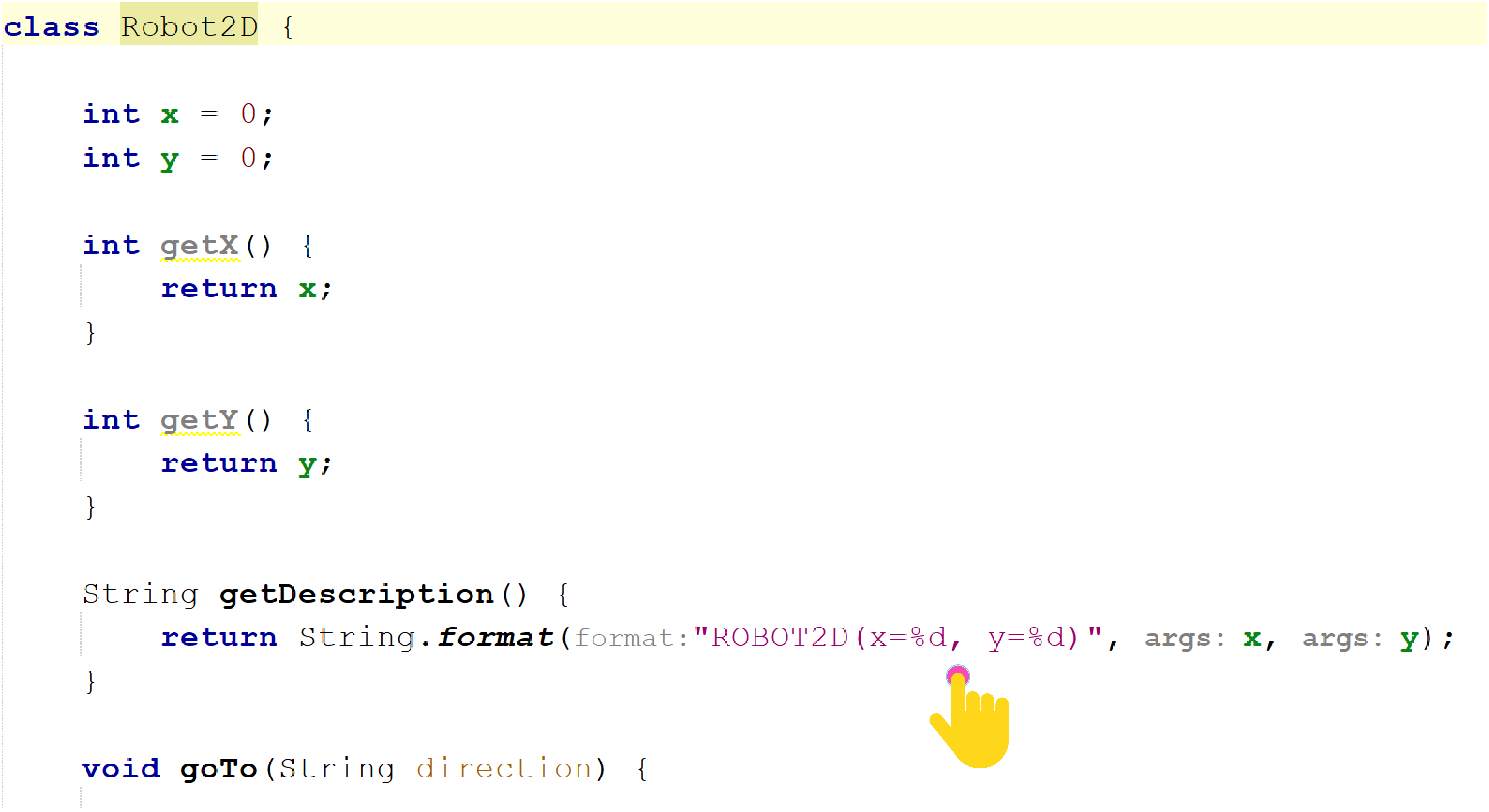


```
class Robot {

    boolean powerOn = false;

    void turnOn() {
        this.powerOn = true;
    }

    void turnOff() {
        this.powerOn = false;
    }

    void describe() {
        System.out.printf(format: "Is Robot power-on? %B %n", args: this.powerOn);
    }

}
```

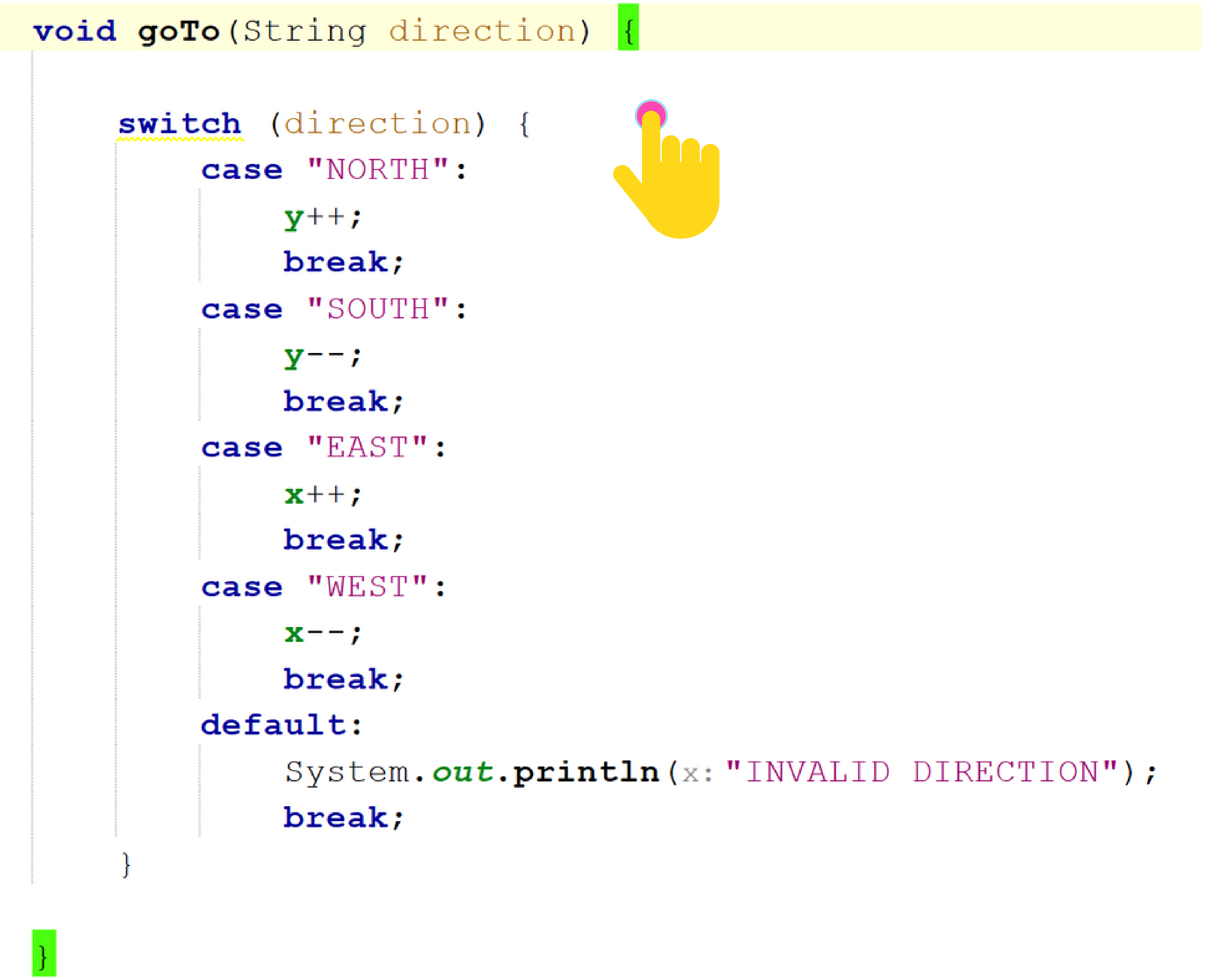


```
public static void main(String[] args) {

    Robot myRobot = new Robot();

    myRobot.describe();

    myRobot.turnOn();

    myRobot.describe();

    myRobot.turnOff();

    myRobot.describe();

}
```

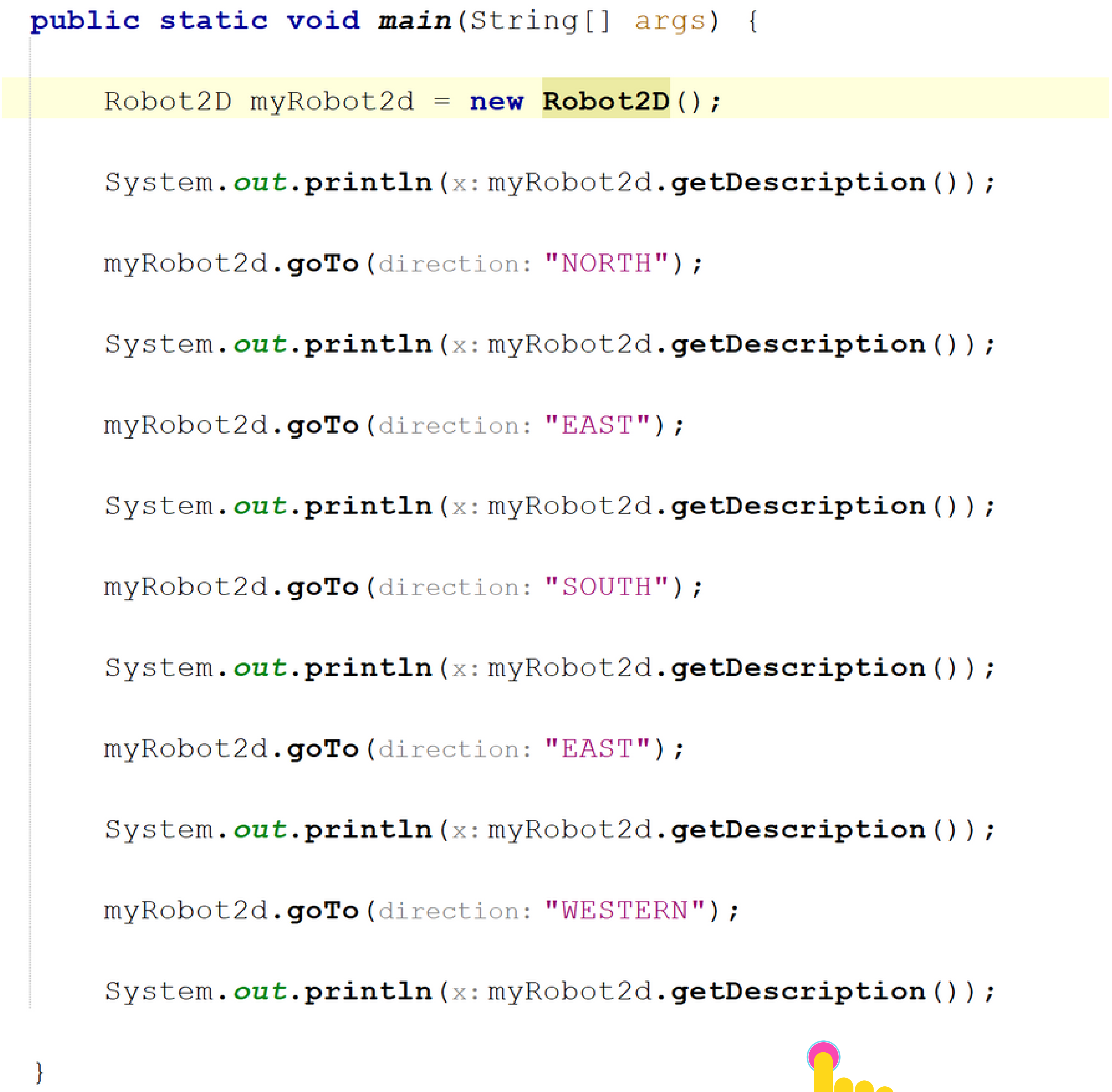
```
run:
Is Robot power-on? FALSE
Is Robot power-on? TRUE
Is Robot power-on? FALSE
BUILD SUCCESSFUL (total time: 0 seconds)
```



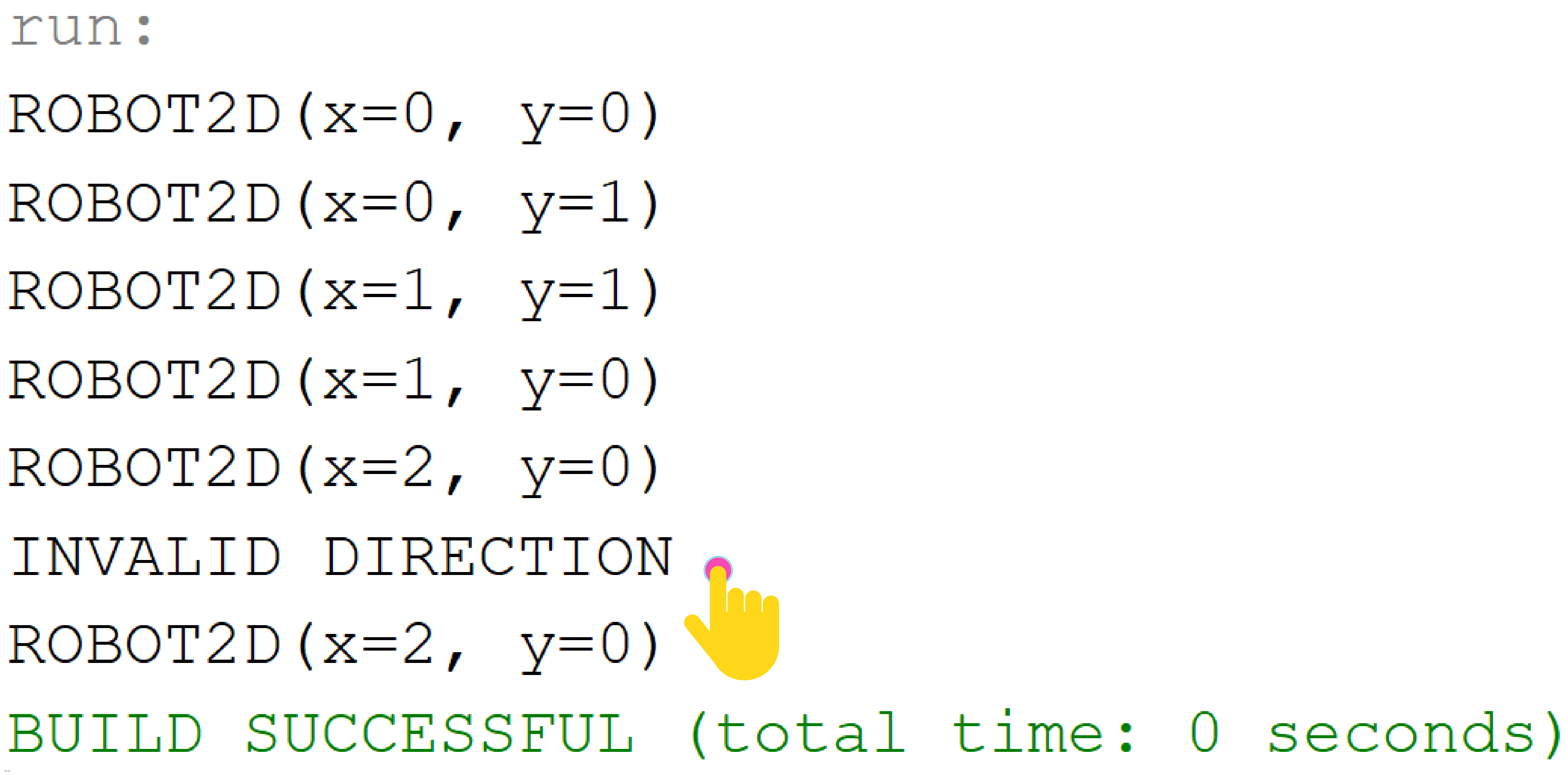
```
class Robot2D {

    int x = 0;
    int y = 0;

    int getX() {
        return x;
    }

    int getY() {
        return y;
    }

    String getDescription() {
        return String.format(format:"ROBOT2D(x=%d, y=%d)", args: x, args: y);
    }

    void goTo(String direction) {
```



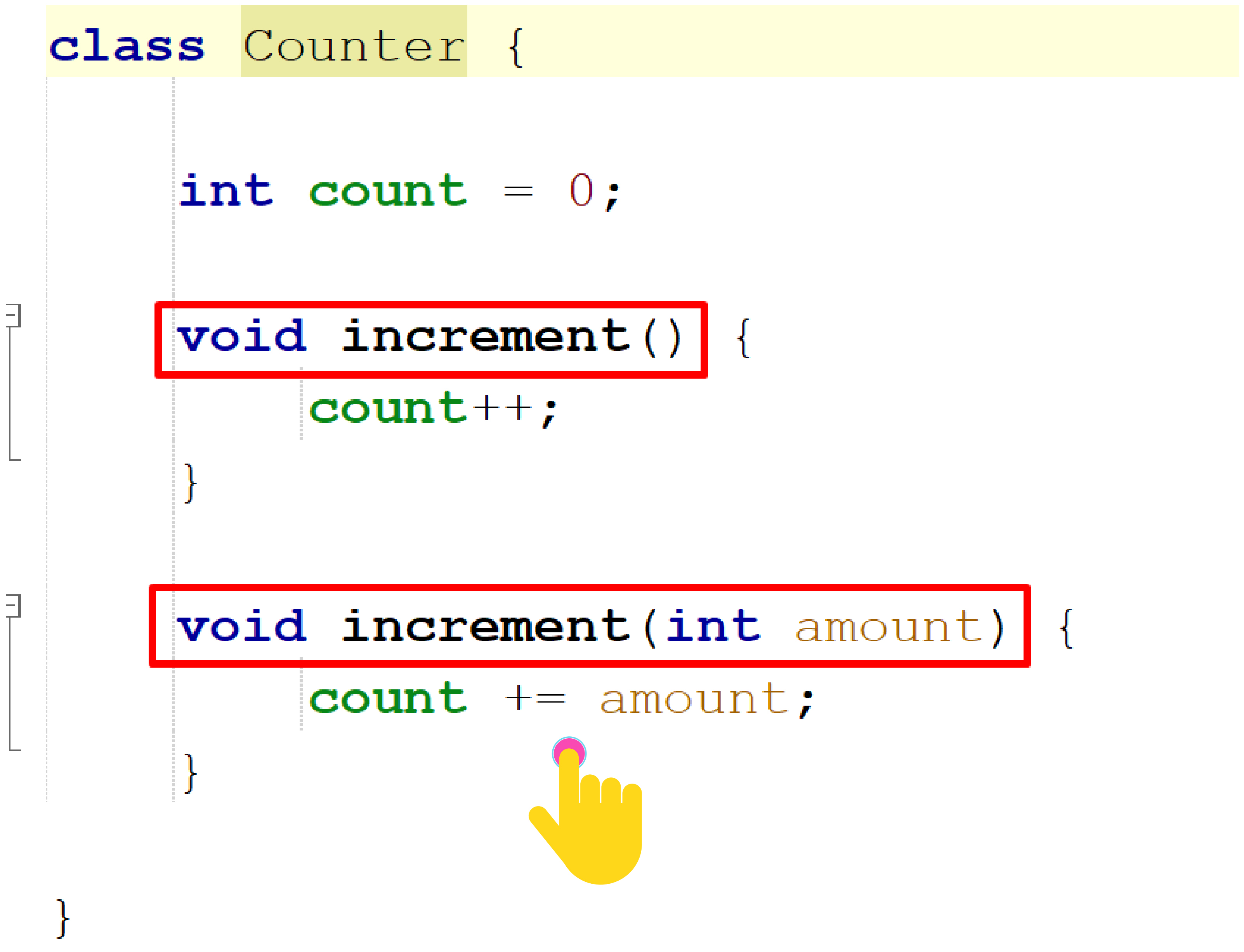
```java
void goTo(String direction) {

    switch (direction) {
        case "NORTH":
            y++;
            break;
        case "SOUTH":
            y--;
            break;
        case "EAST":
            x++;
            break;
        case "WEST":
            x--;
            break;
        default:
            System.out.println(x: "INVALID DIRECTION");
            break;
    }

}
```



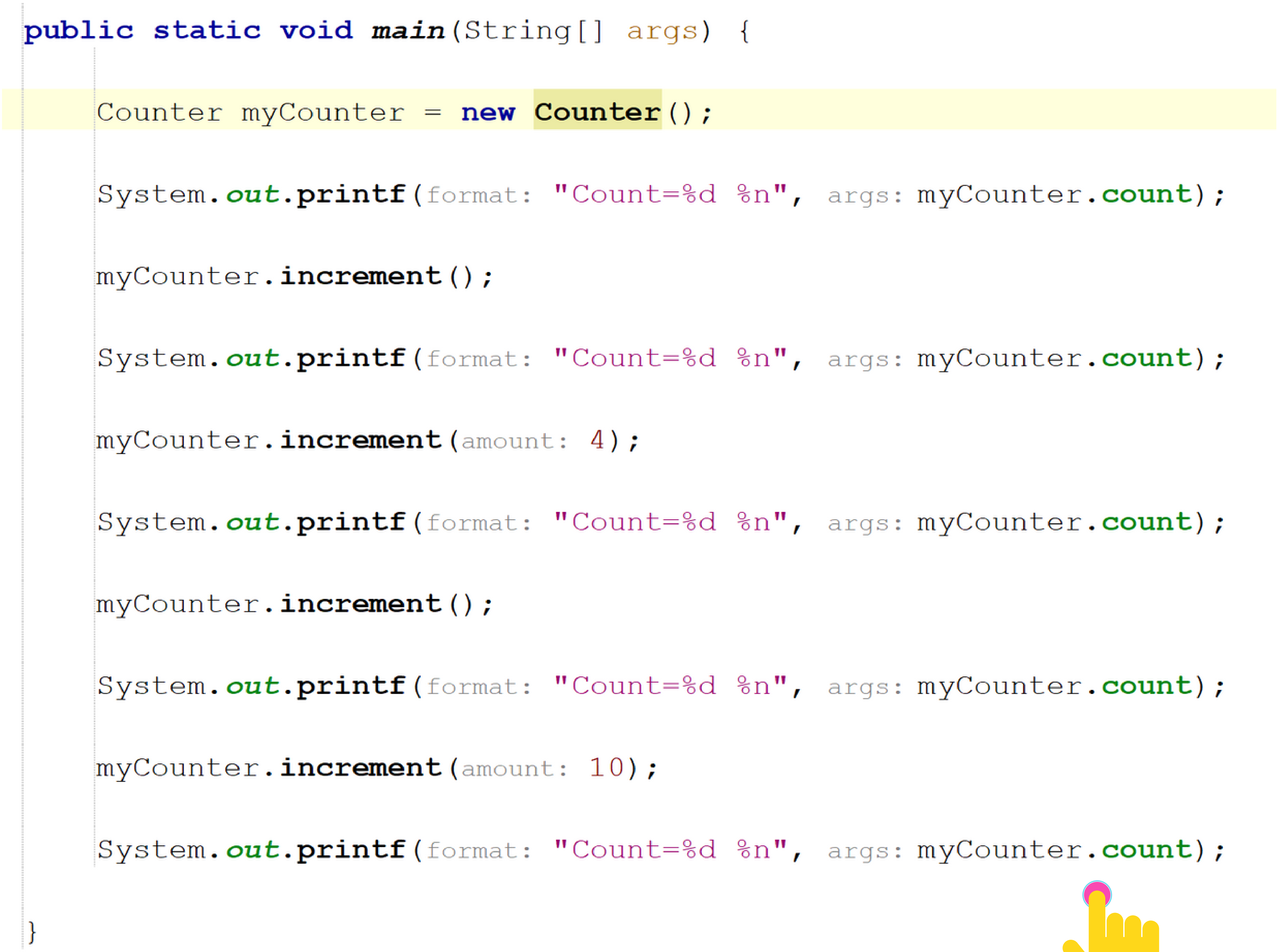
```
public static void main(String[] args) {

    Robot2D myRobot2d = new Robot2D();

    System.out.println(x: myRobot2d.getDescription());

    myRobot2d.goTo(direction: "NORTH");

    System.out.println(x: myRobot2d.getDescription());

    myRobot2d.goTo(direction: "EAST");

    System.out.println(x: myRobot2d.getDescription());

    myRobot2d.goTo(direction: "SOUTH");

    System.out.println(x: myRobot2d.getDescription());

    myRobot2d.goTo(direction: "EAST");

    System.out.println(x: myRobot2d.getDescription());

    myRobot2d.goTo(direction: "WESTERN");

    System.out.println(x: myRobot2d.getDescription());

}
```



```
run:
ROBOT2D(x=0, y=0)
ROBOT2D(x=0, y=1)
ROBOT2D(x=1, y=1)
ROBOT2D(x=1, y=0)
ROBOT2D(x=2, y=0)
INVALID DIRECTION
ROBOT2D(x=2, y=0)
BUILD SUCCESSFUL (total time: 0 seconds)
```

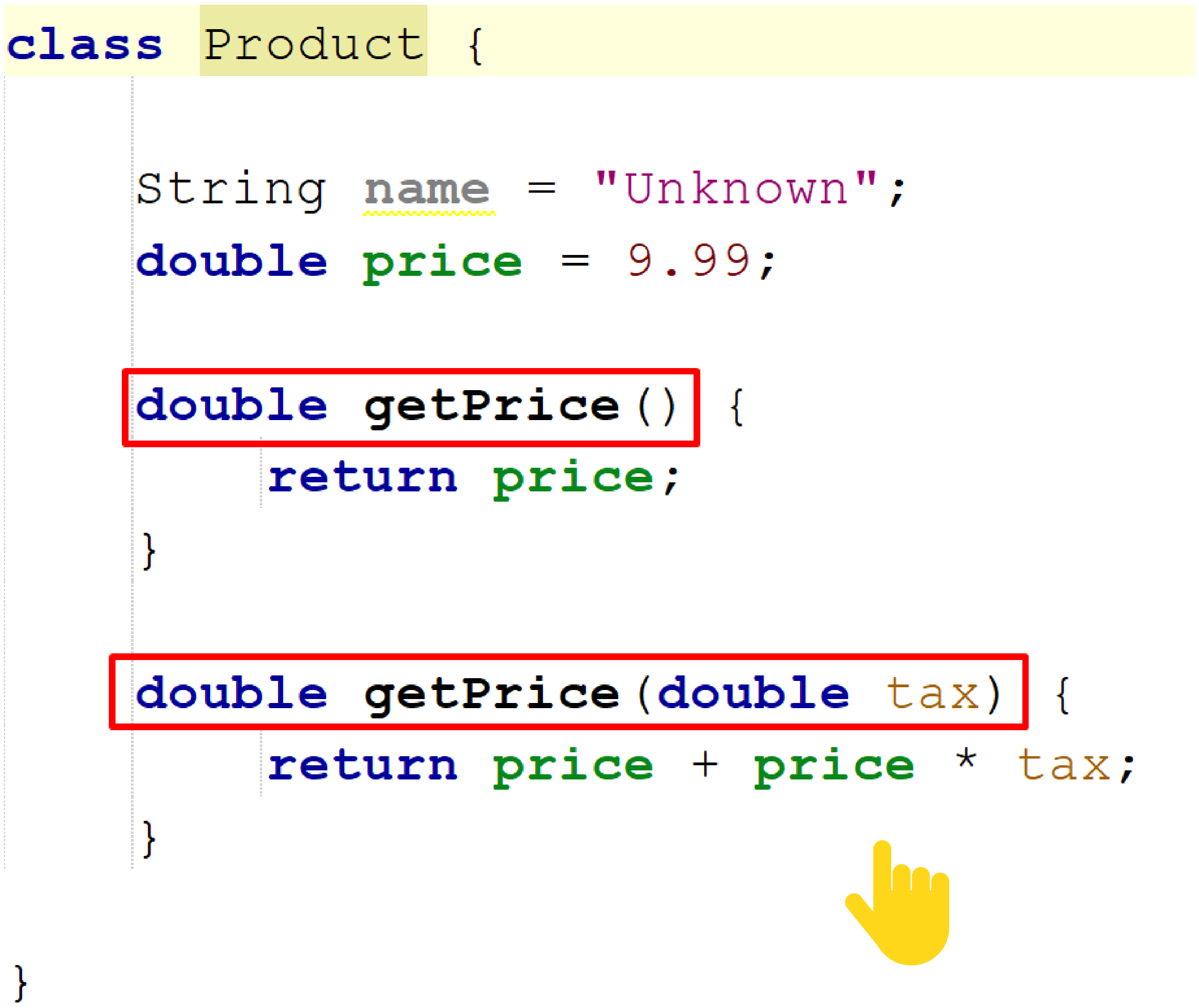


```
class Counter {

    int count = 0;

    void increment() {
        count++;
    }

    void increment(int amount) {
        count += amount;
    }

}
```

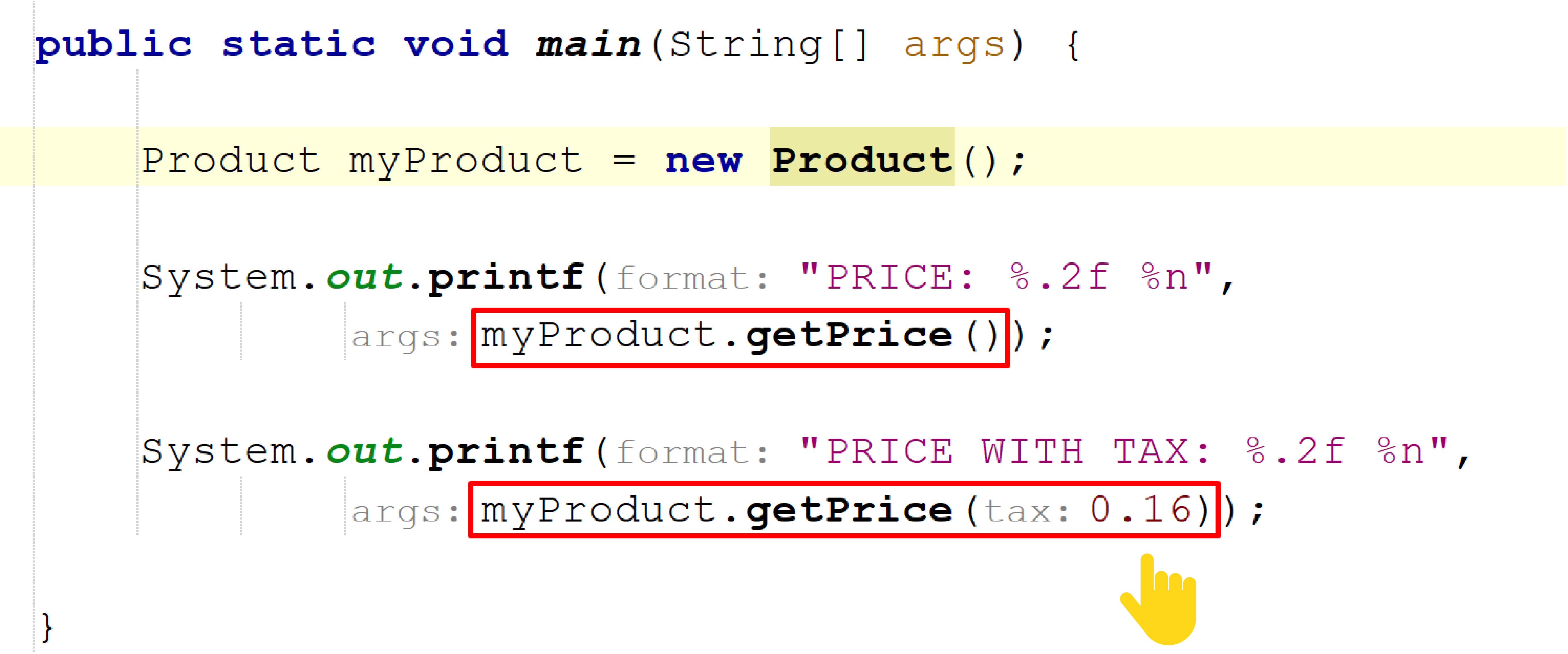


```
public static void main(String[] args) {
    Counter myCounter = new Counter();
    System.out.printf(format: "Count=%d %n", args: myCounter.count);
    myCounter.increment();
    System.out.printf(format: "Count=%d %n", args: myCounter.count);
    myCounter.increment(amount: 4);
    System.out.printf(format: "Count=%d %n", args: myCounter.count);
    myCounter.increment();
    System.out.printf(format: "Count=%d %n", args: myCounter.count);
    myCounter.increment(amount: 10);
    System.out.printf(format: "Count=%d %n", args: myCounter.count);
}
```

```
run:
Count=0
Count=1
Count=5
Count=6
Count=16
BUILD SUCCESSFUL (total time: 0 seconds)
```



```
class Product {

    String name = "Unknown";
    double price = 9.99;

    double getPrice() {
        return price;
    }

    double getPrice(double tax) {
        return price + price * tax;
    }

}
```



```java
public static void main(String[] args) {

    Product myProduct = new Product();

    System.out.printf(format: "PRICE: %.2f %n",
            args: myProduct.getPrice());

    System.out.printf(format: "PRICE WITH TAX: %.2f %n",
            args: myProduct.getPrice(tax: 0.16));

}
```

```
run:
PRICE: 9.99
PRICE WITH TAX: 11.59
BUILD SUCCESSFUL (total time: 0 seconds)
```