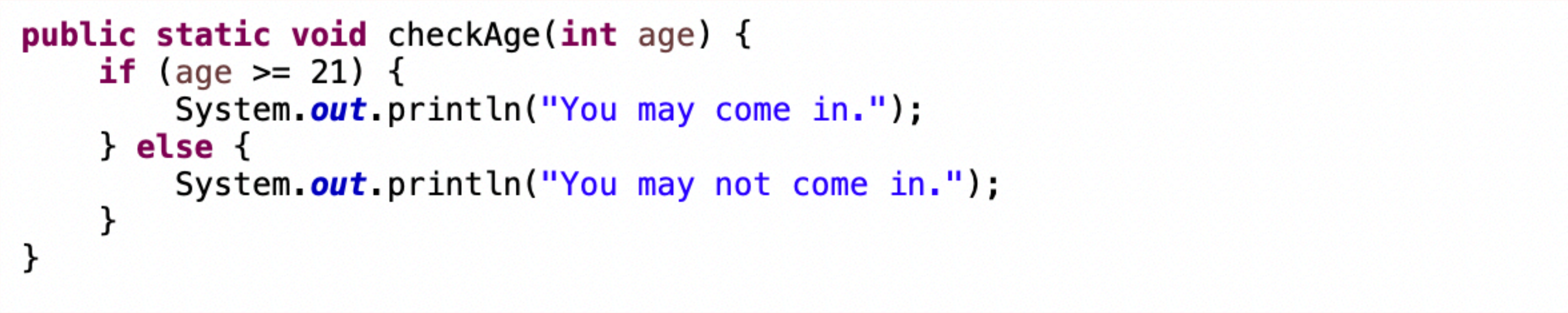
# CONTROL STRUCTURES AND LOGICAL OPERATORS

# LOGICAL OPERATORS

- Logical operators are used to check if a condition is met. For example, we may want to check if a boolean variable is true or a number is above or below some other number.
- Operators:
  - And: & and &&
  - Or: |
  - Not: !
  - Equal: ==
  - Not Equal: !=
- Strings are a special case where using == to compare equality won't work, you must use the .equals() method: string1.equals(string2)

# IF/ELSE STATEMENTS

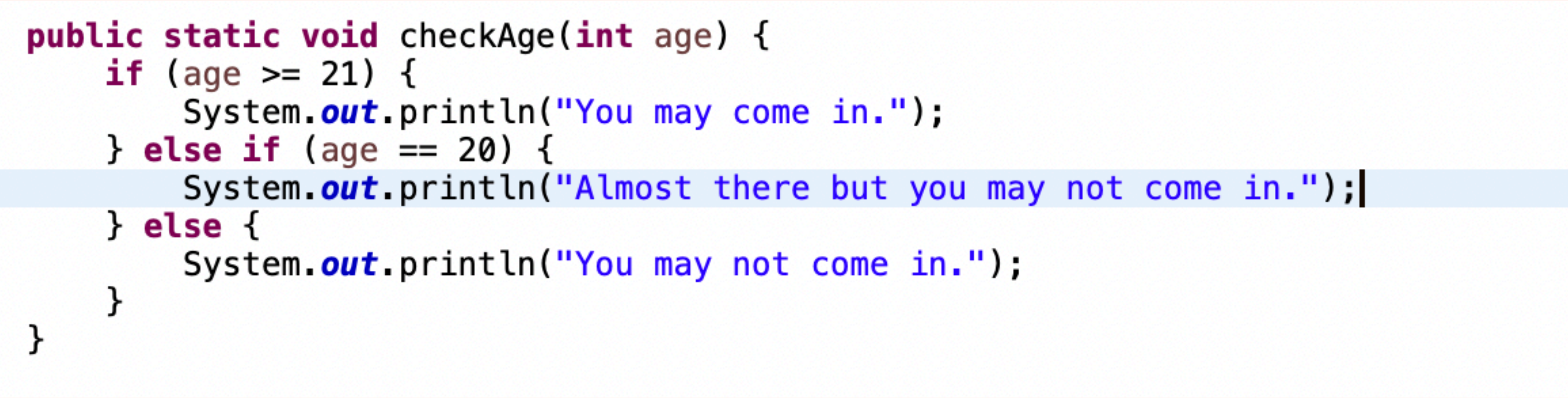
- If/else statements allow you to only run a block of code if a condition is met. For example, if you are signing in at the door of pins with your id. Then we could have an if statement that says if this person’s age is 21 or over, then they can come in, else they can’t come in.



```java
public static void checkAge(int age) {
    if (age >= 21) {
        System.out.println("You may come in.");
    } else {
        System.out.println("You may not come in.");
    }
}
```

## BUT WHAT IF WE HAVE MORE THAN JUST ONE CONDITION TO CHECK FOR THE SAME VARIABLE?

# ELSE IF



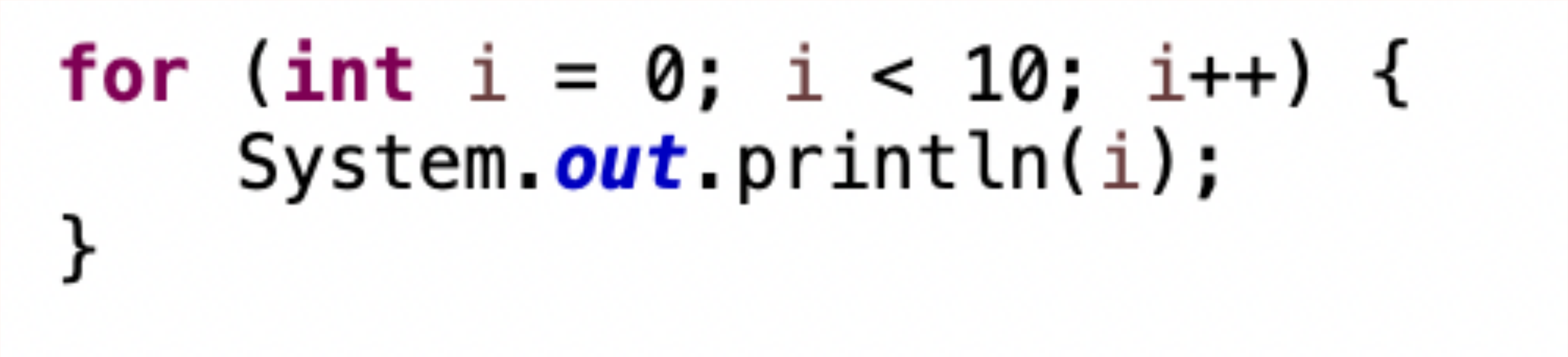
```java
public static void checkAge(int age) {
    if (age >= 21) {
        System.out.println("You may come in.");
    } else if (age == 20) {
        System.out.println("Almost there but you may not come in.");
    } else {
        System.out.println("You may not come in.");
    }
}
```

# LOOPS

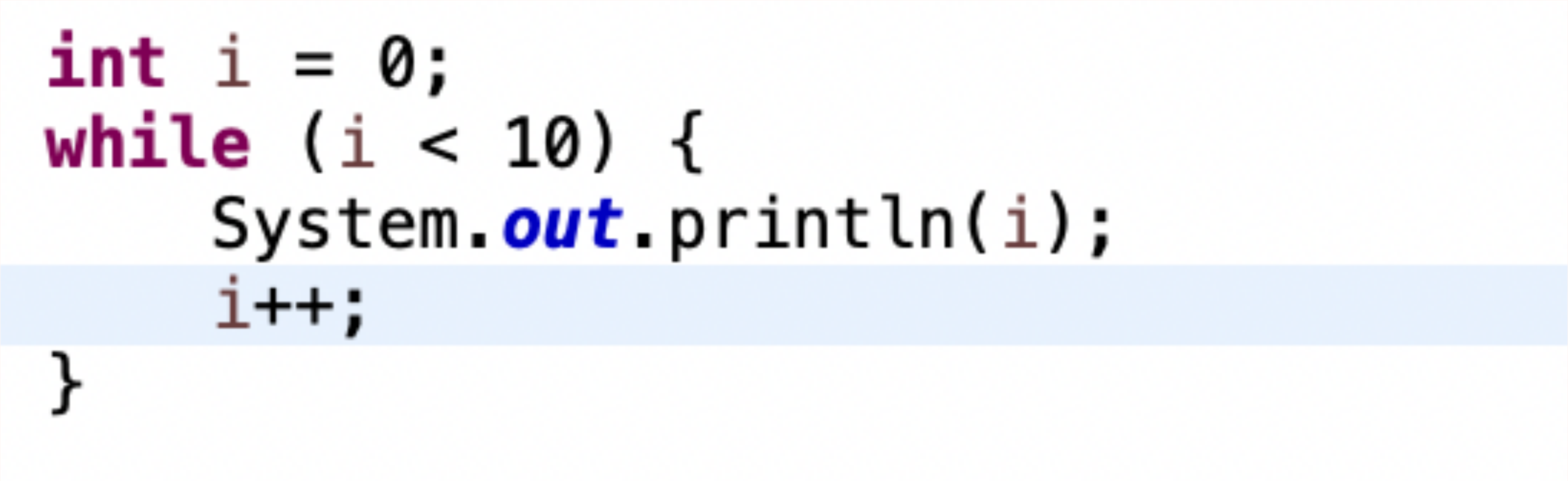
- Loops allow you to repeat the same block of code multiple times. This can be useful if you need to perform an action on every element in a list or you need to a mathematic operation on a series of numbers.
- Types of Loops
    - For loop: A for loop allows you to iterate through a series of numbers while a condition is met. For example, if I wanted to look at every number from 1 to 10, I could use a for loop.
    - While loop: A while loop works just like a for loop except it is more often used when you aren’t looking at a series, but instead just a condition.

# EXAMPLES

**These two loops will print the same output to the console.**



```
for (int i = 0; i < 10; i++) {
    System.out.println(i);
}
```



```
int i = 0;
while (i < 10) {
    System.out.println(i);
    i++;
}
```

**You will notice that I started the counter variable, i, at 0 in each example. That is because of the way that programming languages handle indexing for arrays and lists, which we will learn about in lesson 3.**