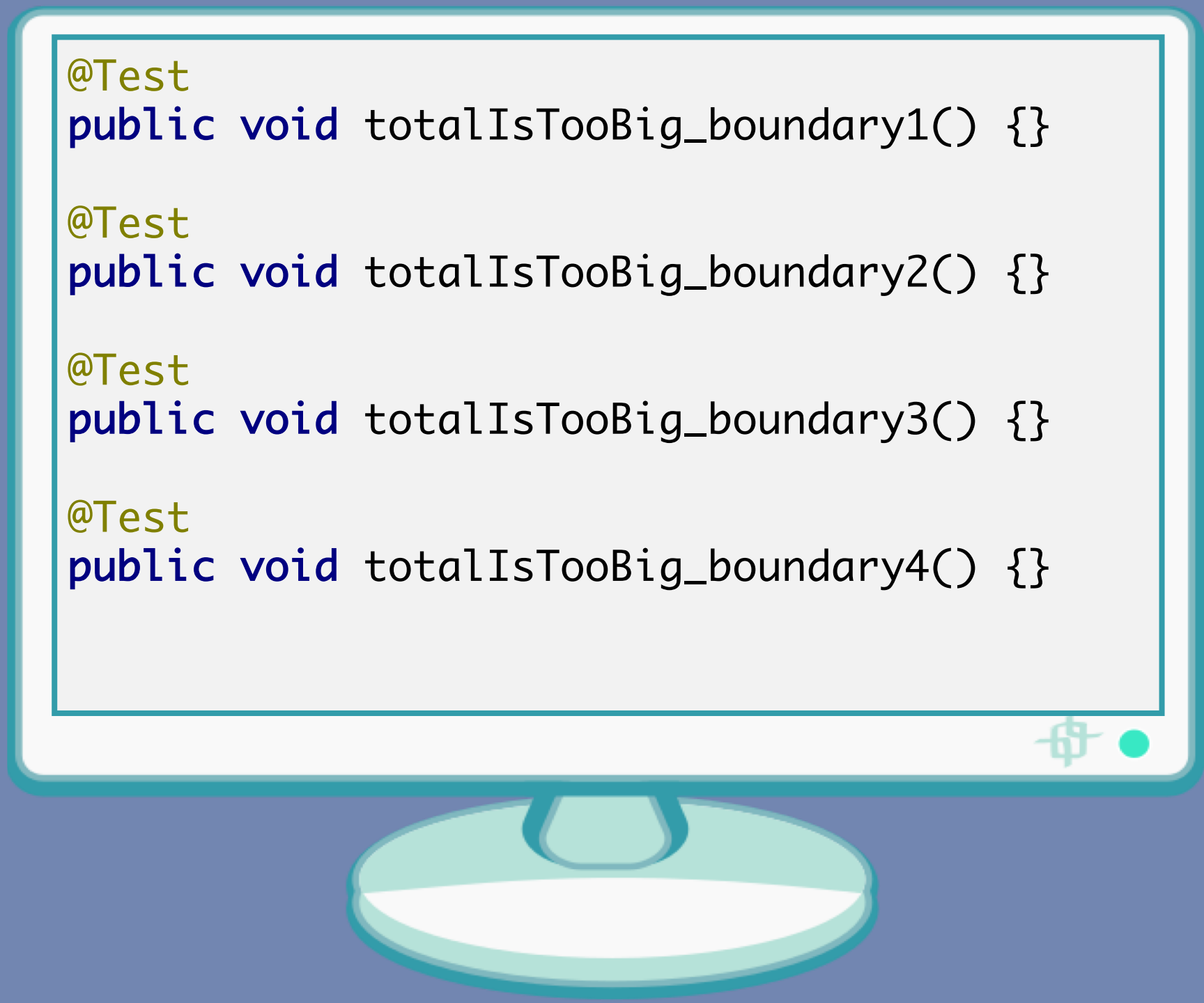
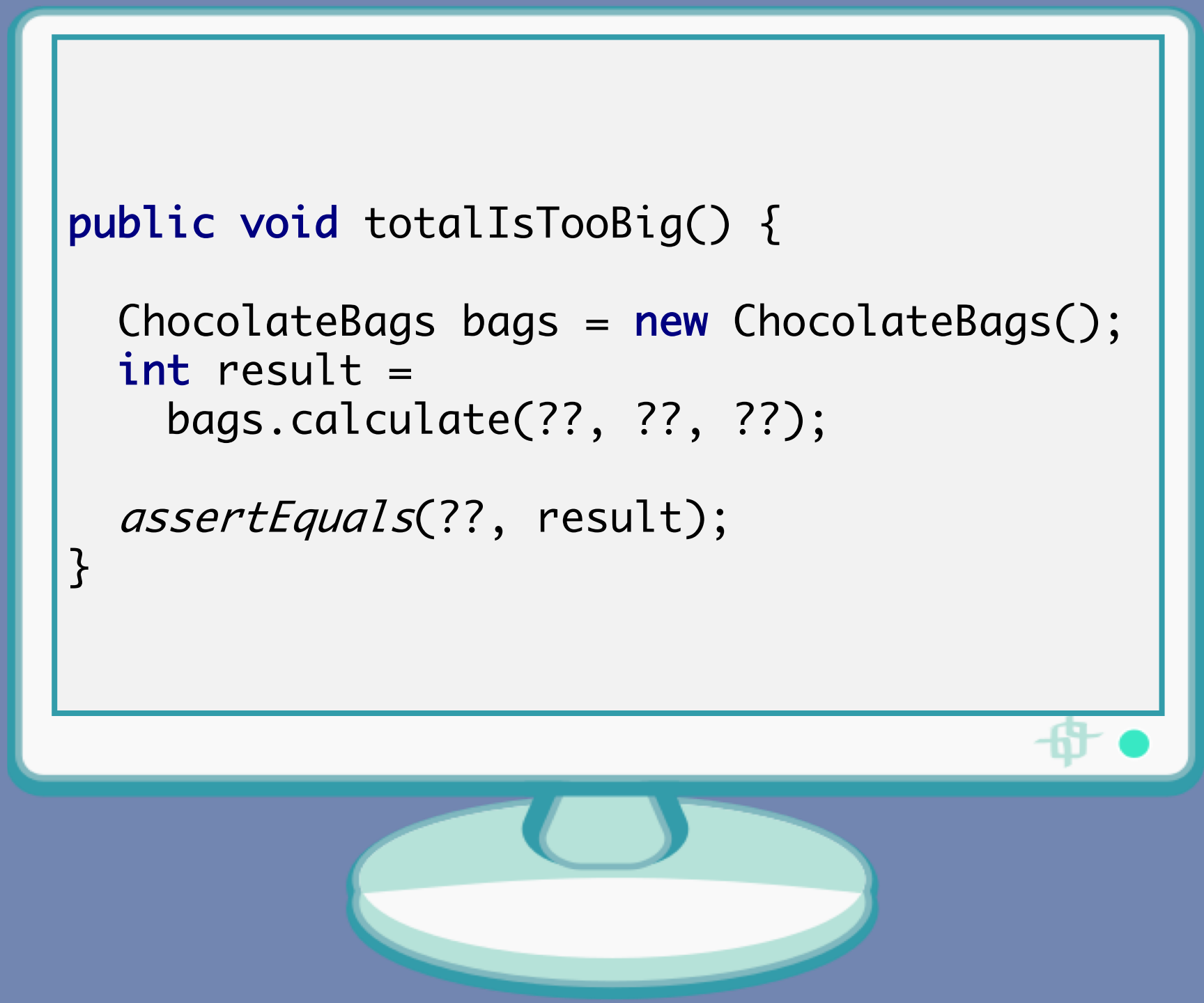
Junit 5 actually has a feature that helps us to write all these tests!

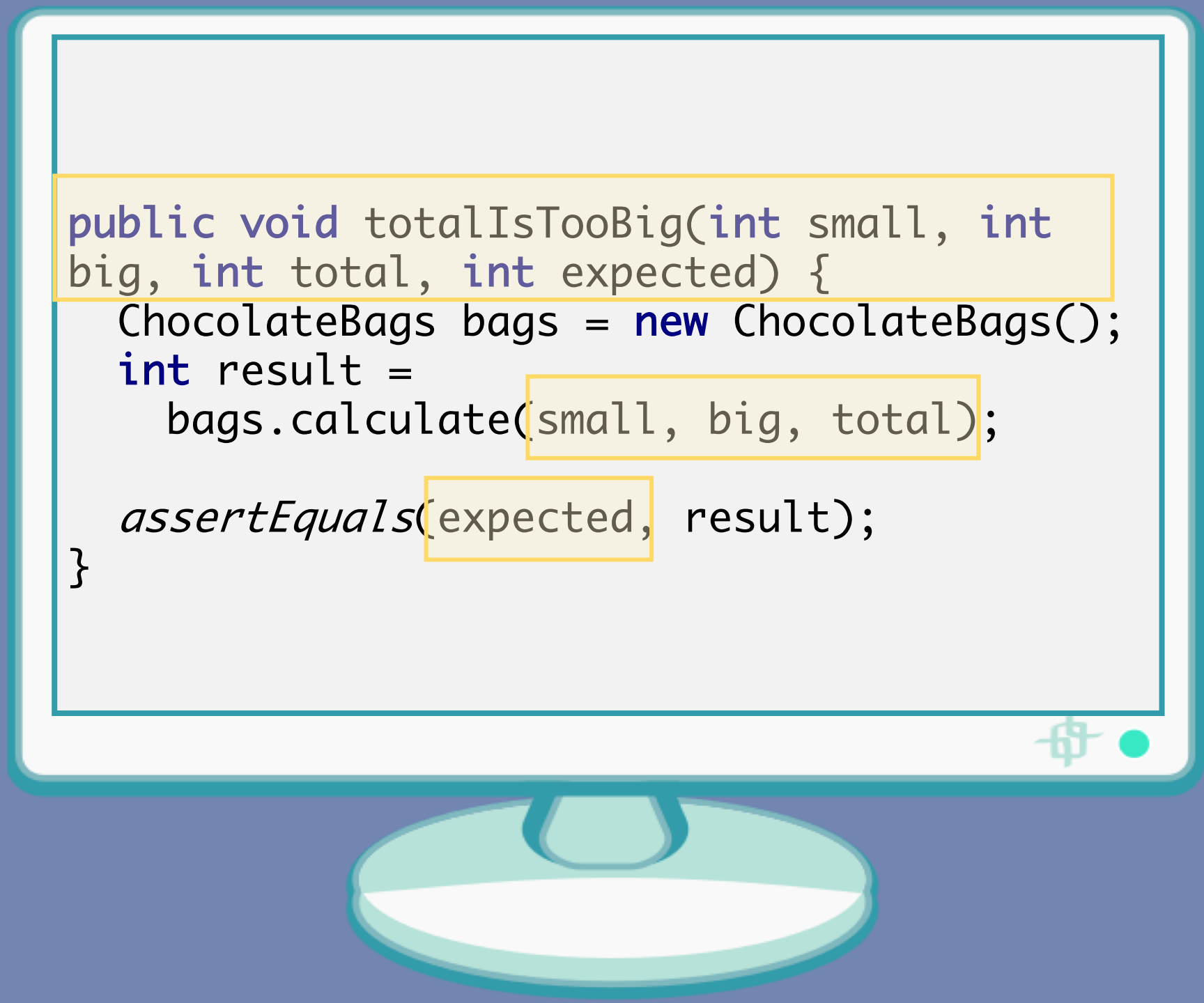
The total is higher than the amount of small and big bars.
Ex: small = 1, big = 1, total = 10

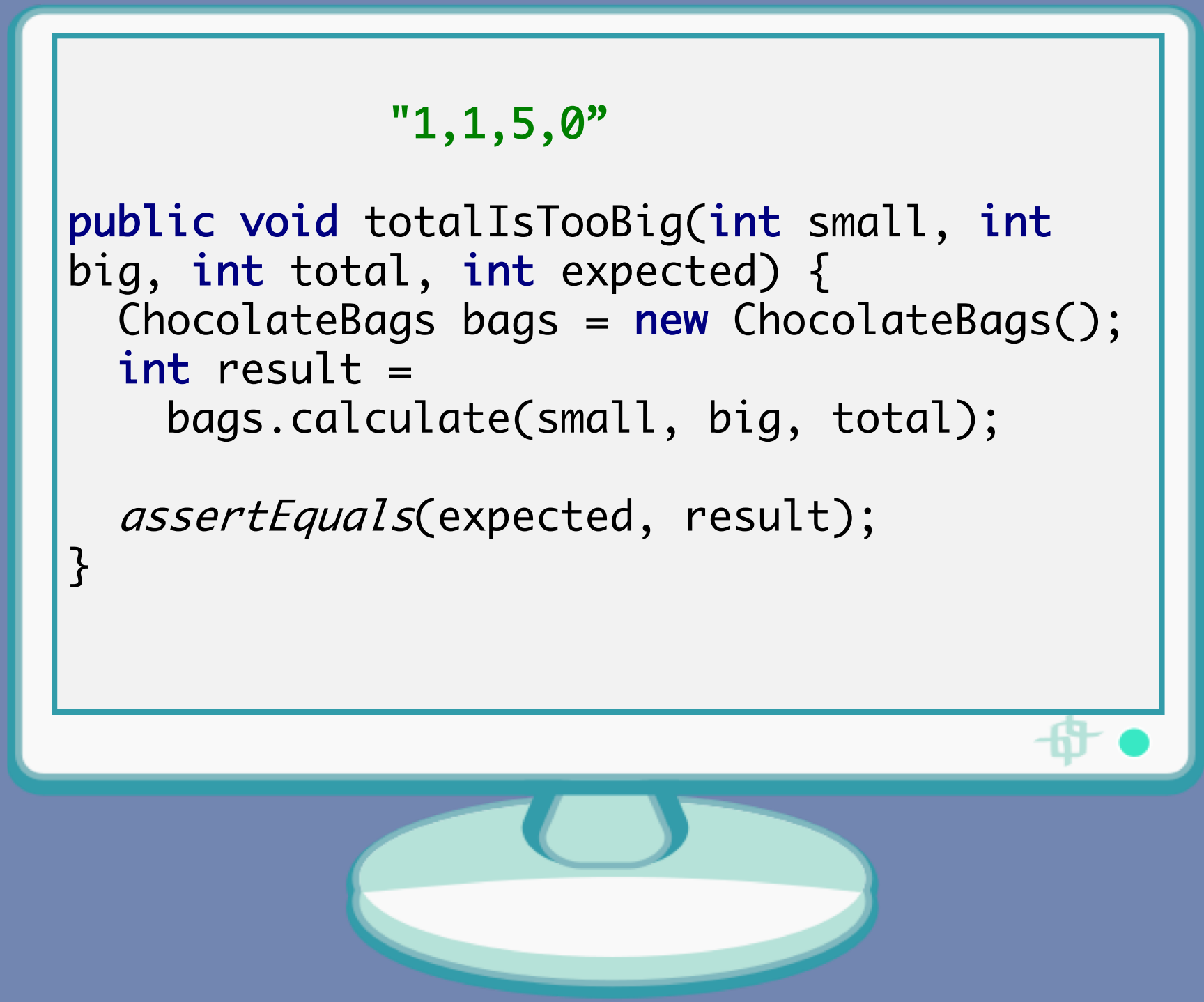
small = 1, big = 1, **total = 5**, = 0
small = 1, big = 1, **total = 6**, = 1

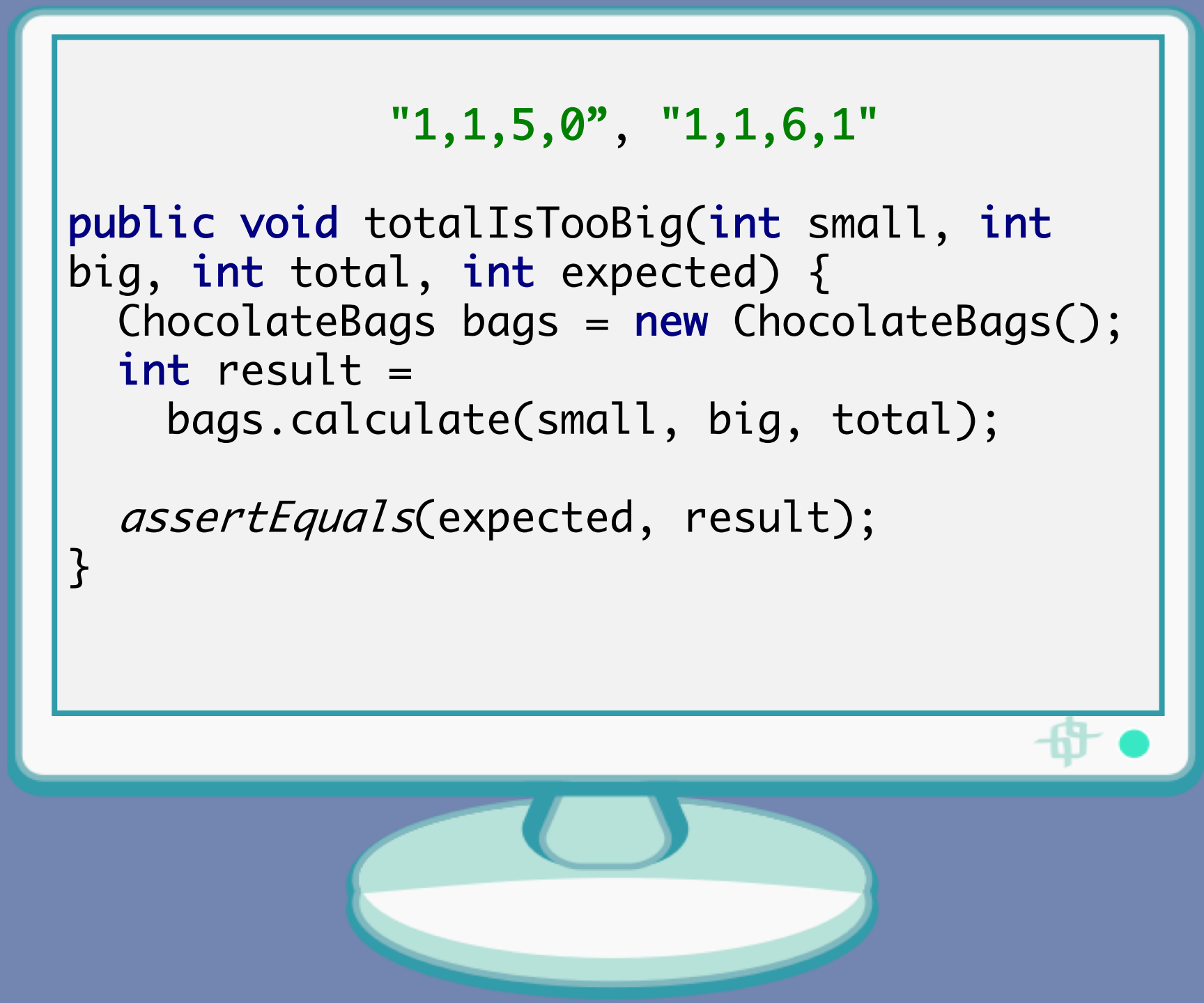
small = 1, big = 1, **total = 7**, = -1
small = 1, big = 1, **total = 8**, = -1

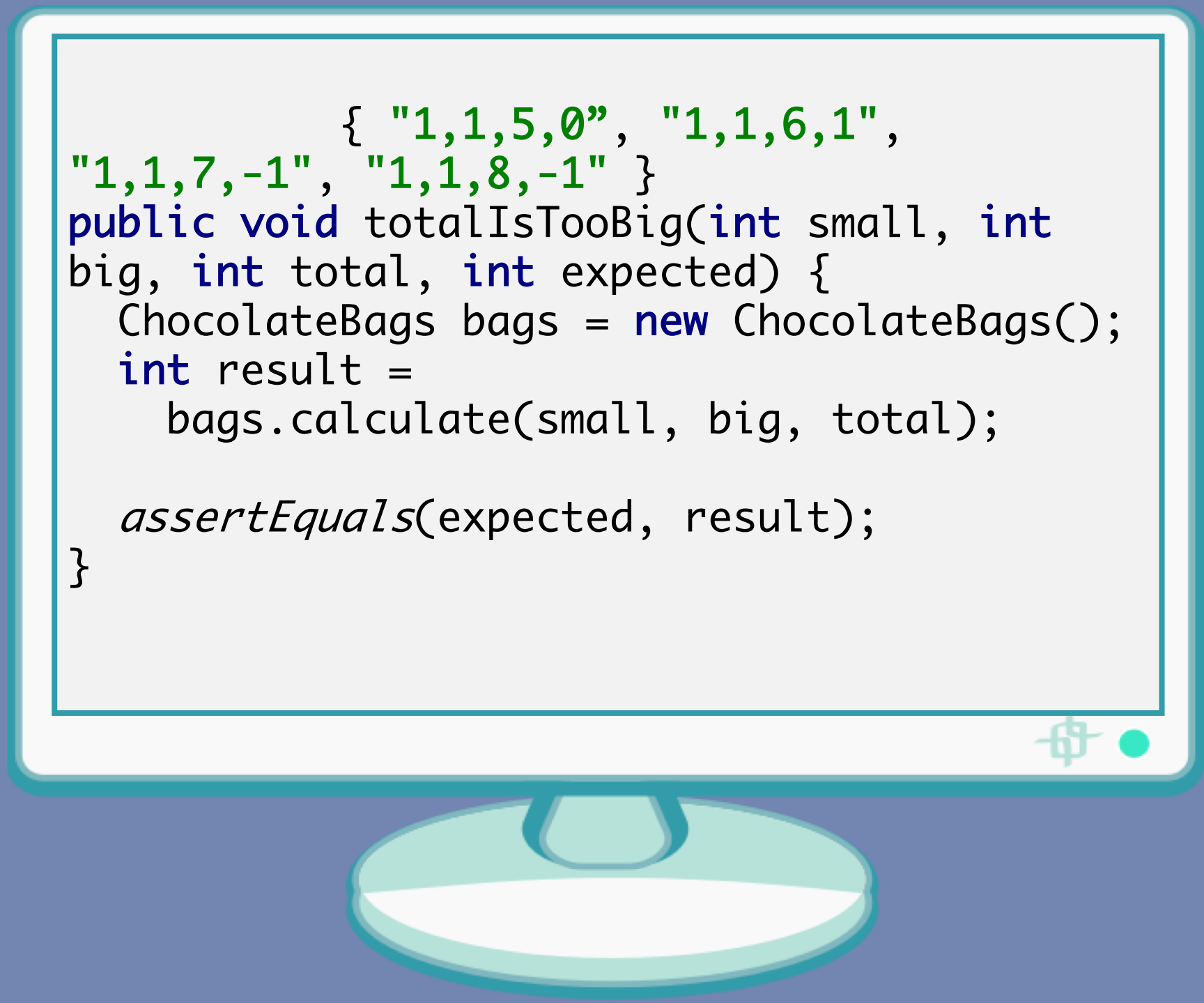

@Test
public void totalIsTooBig_boundary1() {}
@Test
public void totalIsTooBig_boundary2() {}
@Test
public void totalIsTooBig_boundary3() {}
@Test
public void totalIsTooBig_boundary4() {}

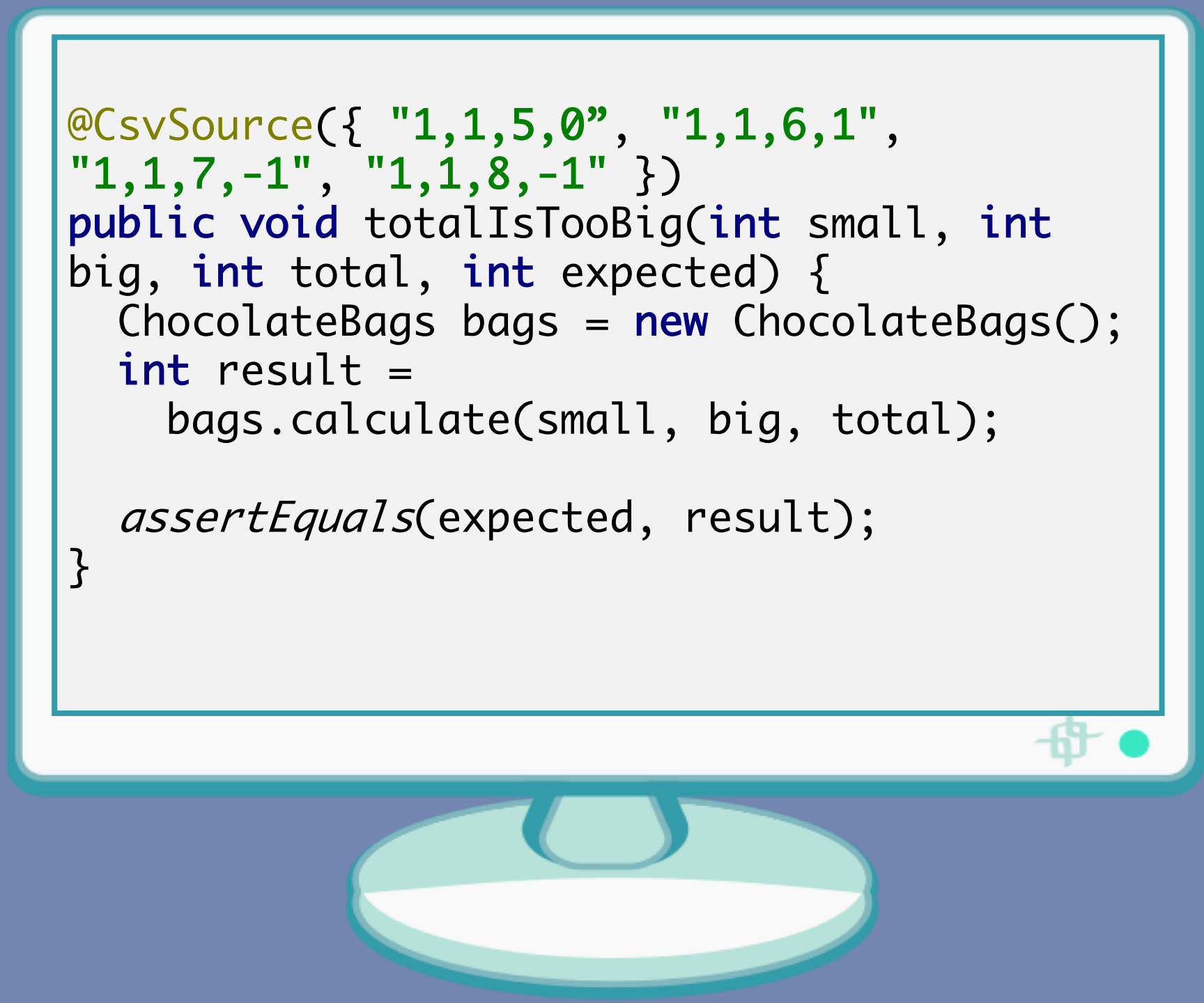

public void totalIsTooBig() {
ChocolateBags bags = new ChocolateBags();
int result =
bags.calculate(??, ??, ??);
assertEquals(??, result);
}

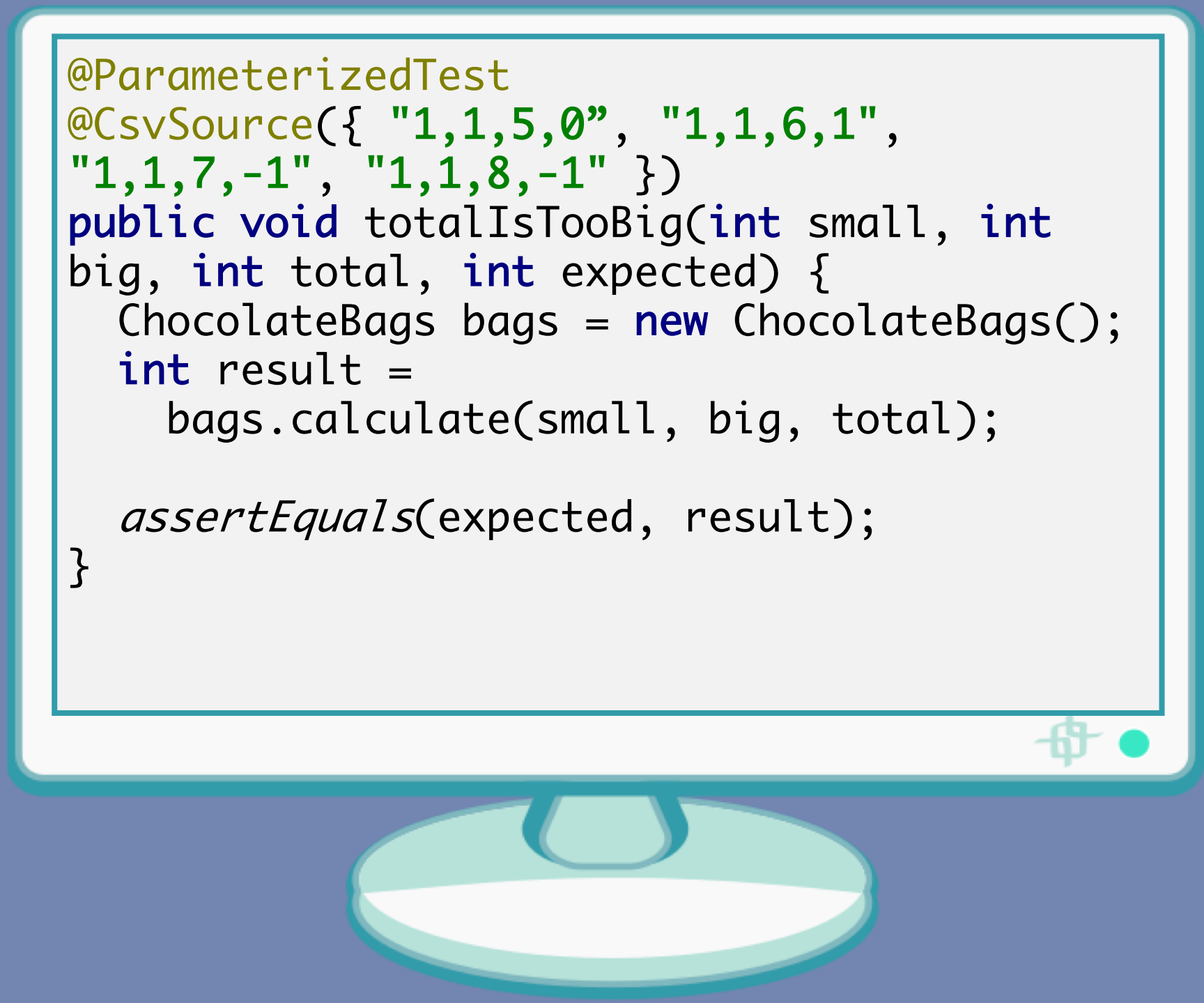

public void totalIsTooBig(int small, int
big, int total, int expected) {
ChocolateBags bags = new ChocolateBags();
int result =
bags.calculate(small, big, total);
assertEquals(expected, result);
}

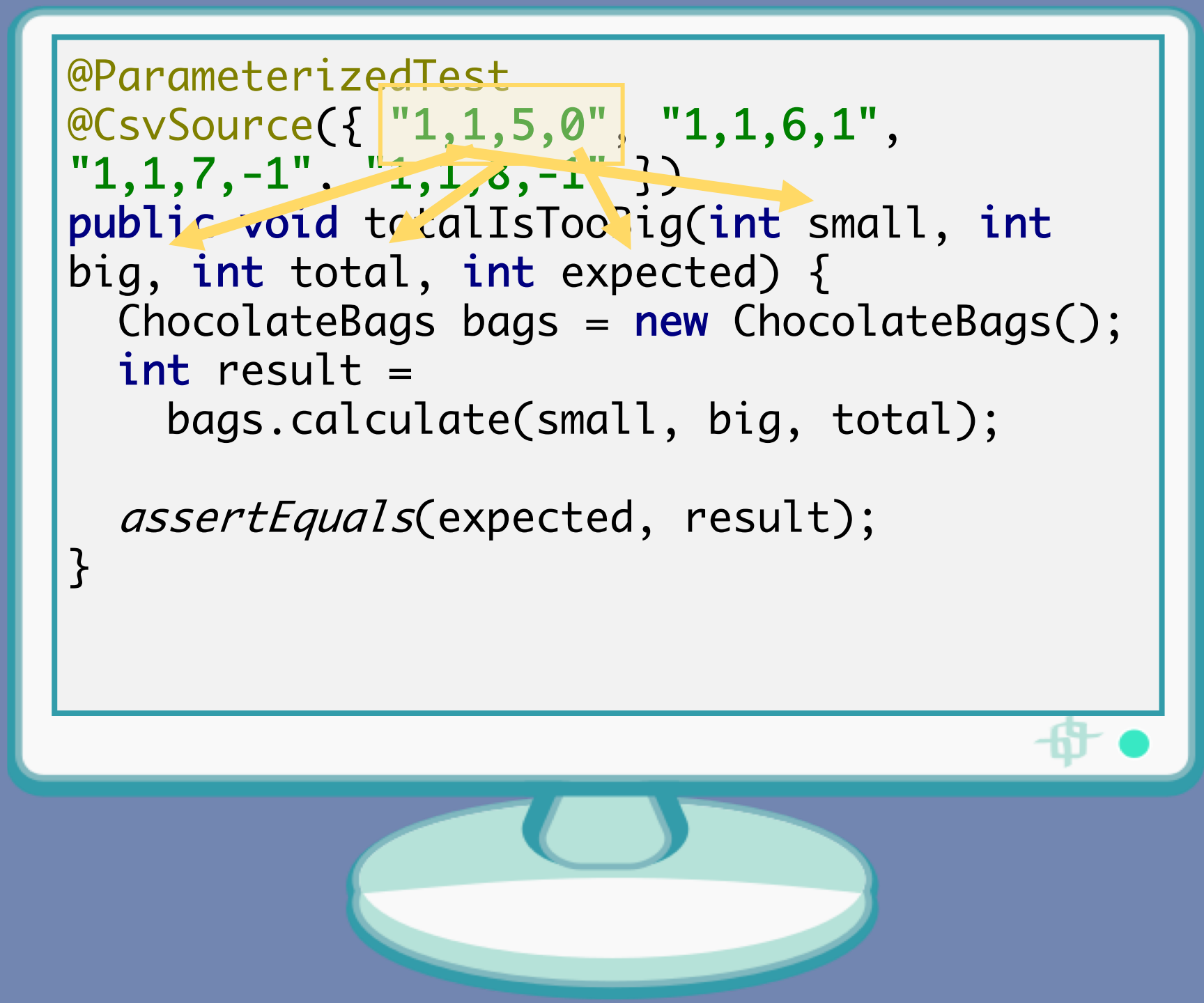

"1,1,5,0”
public void totalIsTooBig(int small, int
big, int total, int expected) {
ChocolateBags bags = new ChocolateBags();
int result =
bags.calculate(small, big, total);
assertEquals(expected, result);
}

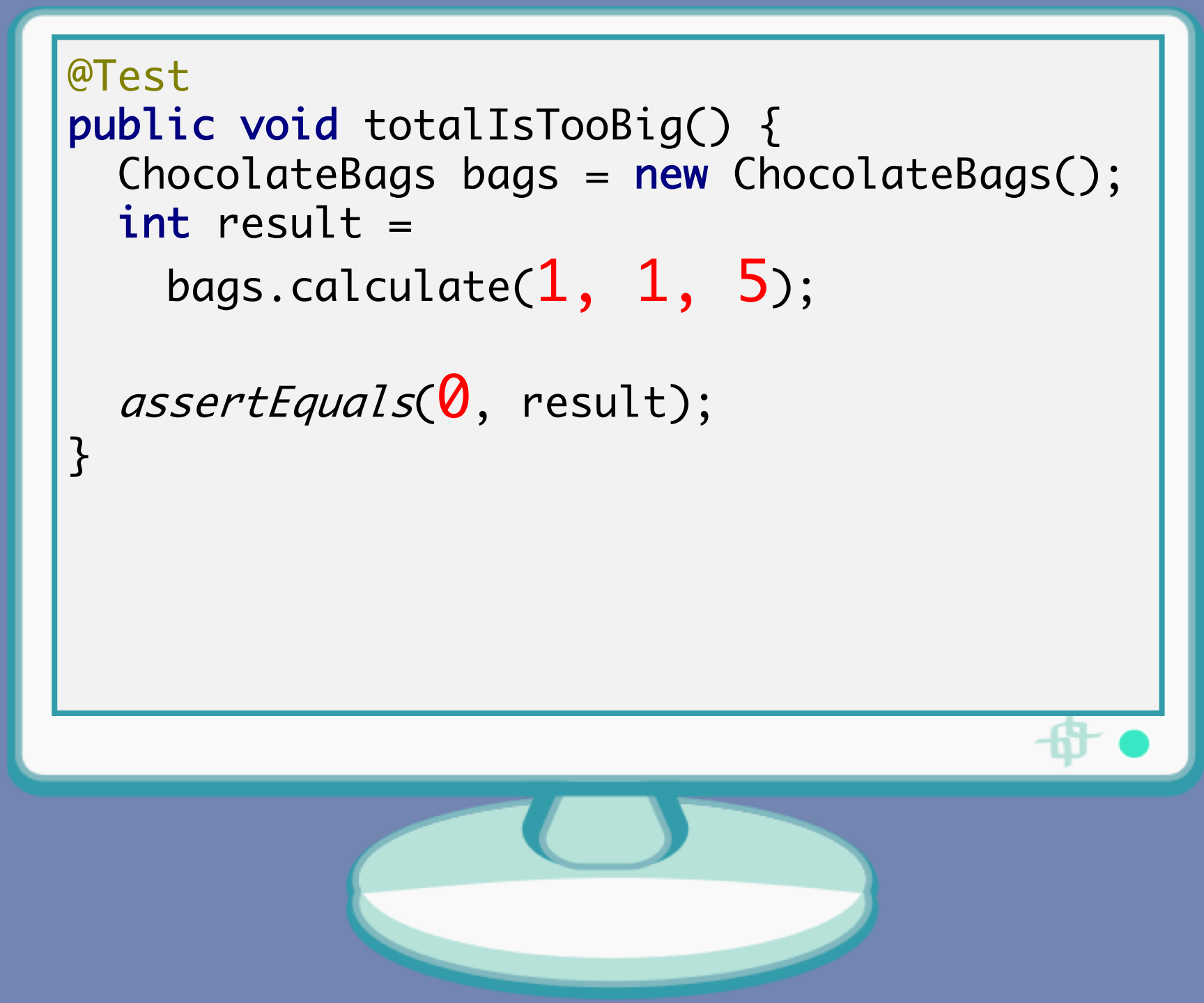

"1,1,5,0", "1,1,6,1"
public void totalIsTooBig(int small, int
big, int total, int expected) {
ChocolateBags bags = new ChocolateBags();
int result =
bags.calculate(small, big, total);
assertEquals(expected, result);
}


{ "1,1,5,0”, "1,1,6,1",
"1,1,7,-1", "1,1,8,-1" }
public void totalIsTooBig(int small, int
big, int total, int expected) {
ChocolateBags bags = new ChocolateBags();
int result =
bags.calculate(small, big, total);
assertEquals(expected, result);
}


@CsvSource({ "1,1,5,0”, "1,1,6,1",
"1,1,7,-1", "1,1,8,-1" })
public void totalIsTooBig(int small, int
big, int total, int expected) {
ChocolateBags bags = new ChocolateBags();
int result =
bags.calculate(small, big, total);
assertEquals(expected, result);
}


@ParameterizedTest
@CsvSource({ "1,1,5,0”, "1,1,6,1",
"1,1,7,-1", "1,1,8,-1" })
public void totalIsTooBig(int small, int
big, int total, int expected) {
ChocolateBags bags = new ChocolateBags();
int result =
bags.calculate(small, big, total);
assertEquals(expected, result);
}

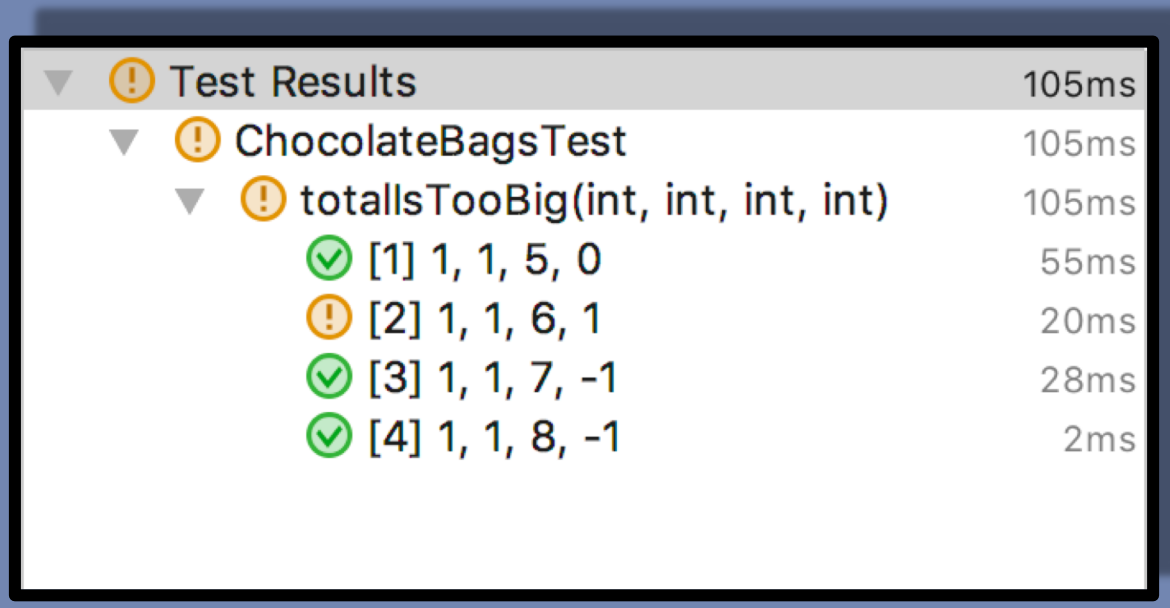

@ParameterizedTest
@CsvSource({ "1,1,5,0", "1,1,6,1",
"1,1,7,-1", "1,1,8,-1" })
public void totalIsTooBig(int small, int
big, int total, int expected) {
ChocolateBags bags = new ChocolateBags();
int result =
bags.calculate(small, big, total);
assertEquals(expected, result);
}


@Test
public void totalIsTooBig() {
ChocolateBags bags = new ChocolateBags();
int result =
bags.calculate(1, 1, 5);
assertEquals(0, result);
}

```
@ParameterizedTest
@CsvSource({ "1,1,5,0", "1,1,6,1",
"1,1,7,-1", "1,1,8,-1" })
public void totalIsTooBig(int small, int
big, int total, int expected) {
  ChocolateBags bags = new ChocolateBags();
  int result =
    bags.calculate(small, big, total);

  assertEquals(expected, result);
}
```

4 tests!


Test Results 105ms
ChocolateBagsTest 105ms
totallsTooBig(int, int, int, int) 105ms
[1] 1, 1, 5, 0 55ms
[2] 1, 1, 6, 1 20ms
[3] 1, 1, 7, -1 28ms
[4] 1, 1, 8, -1 2ms

Only big bars.

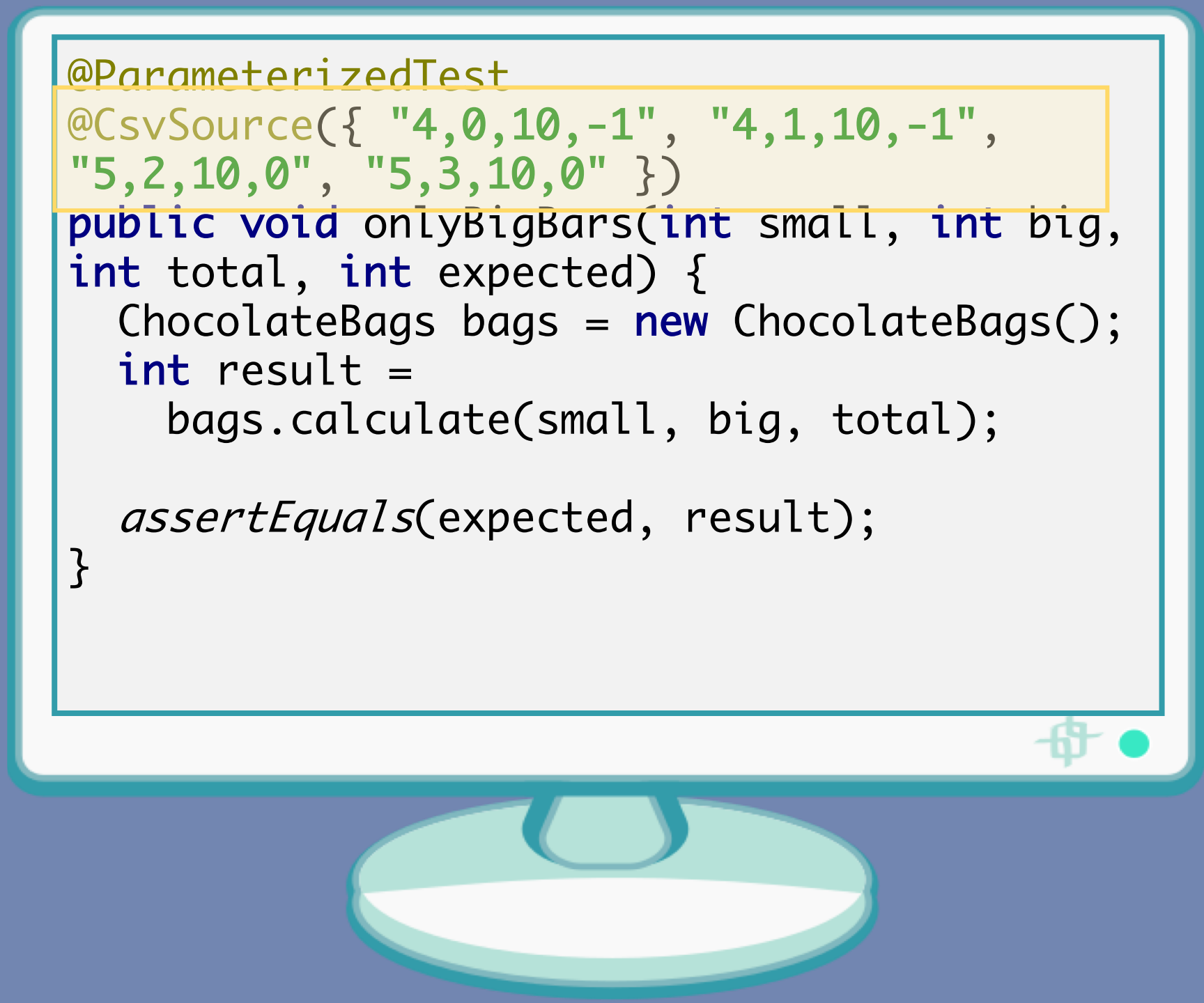
Ex: small = 5, big = 3, total = 10

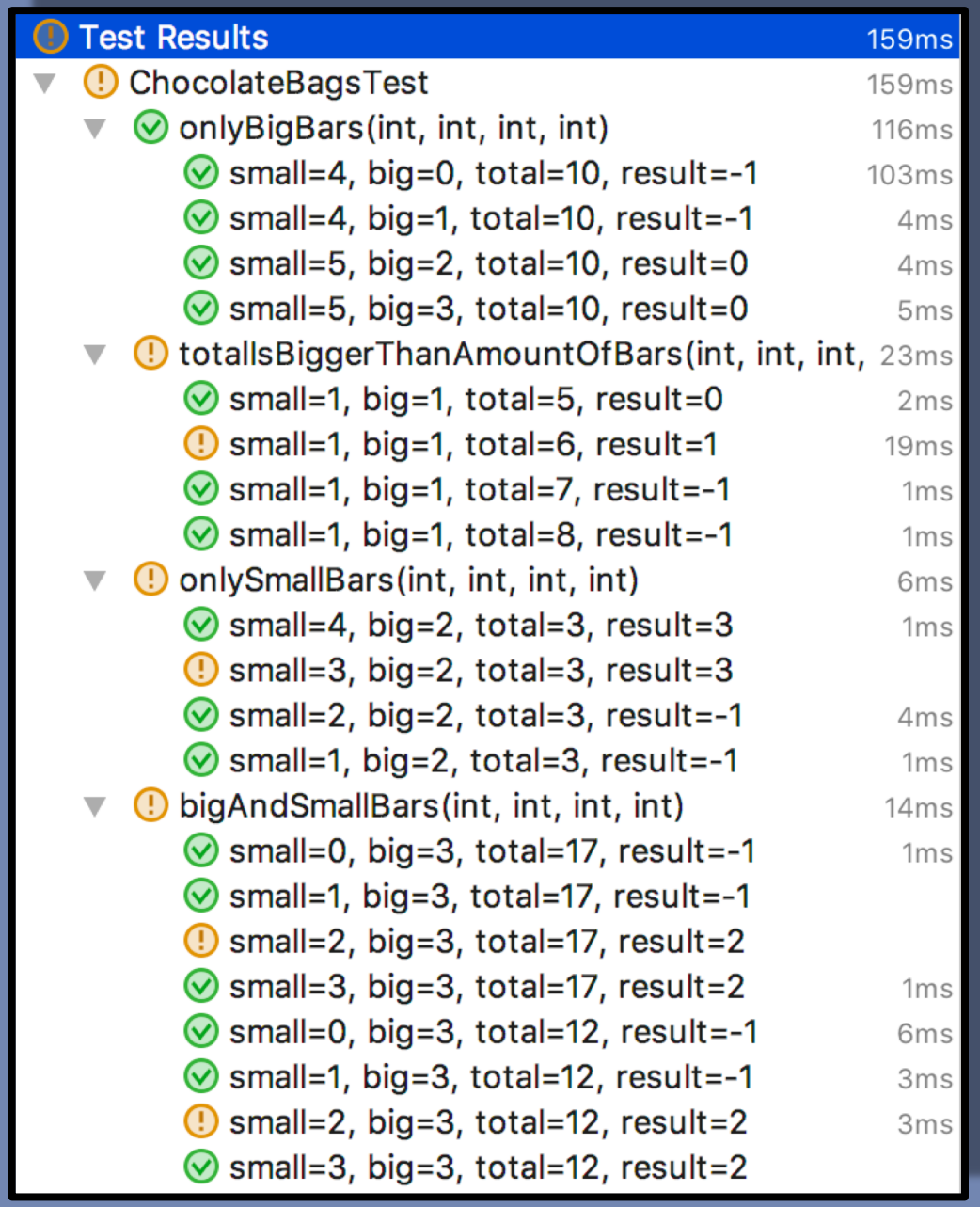
small = 5, **big = 0**, total = 10, = -1

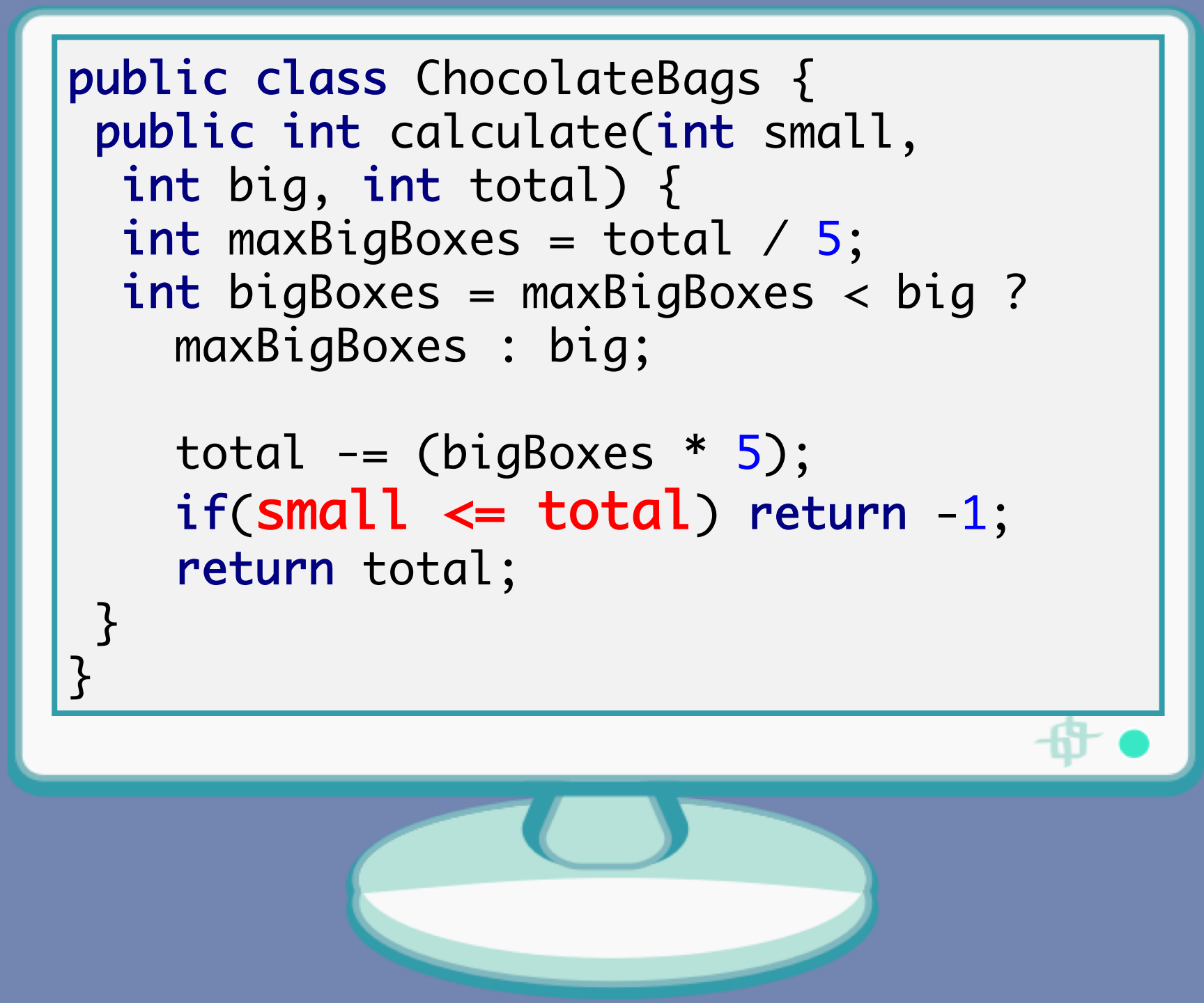
small = 5, **big = 1**, total = 10, = 5

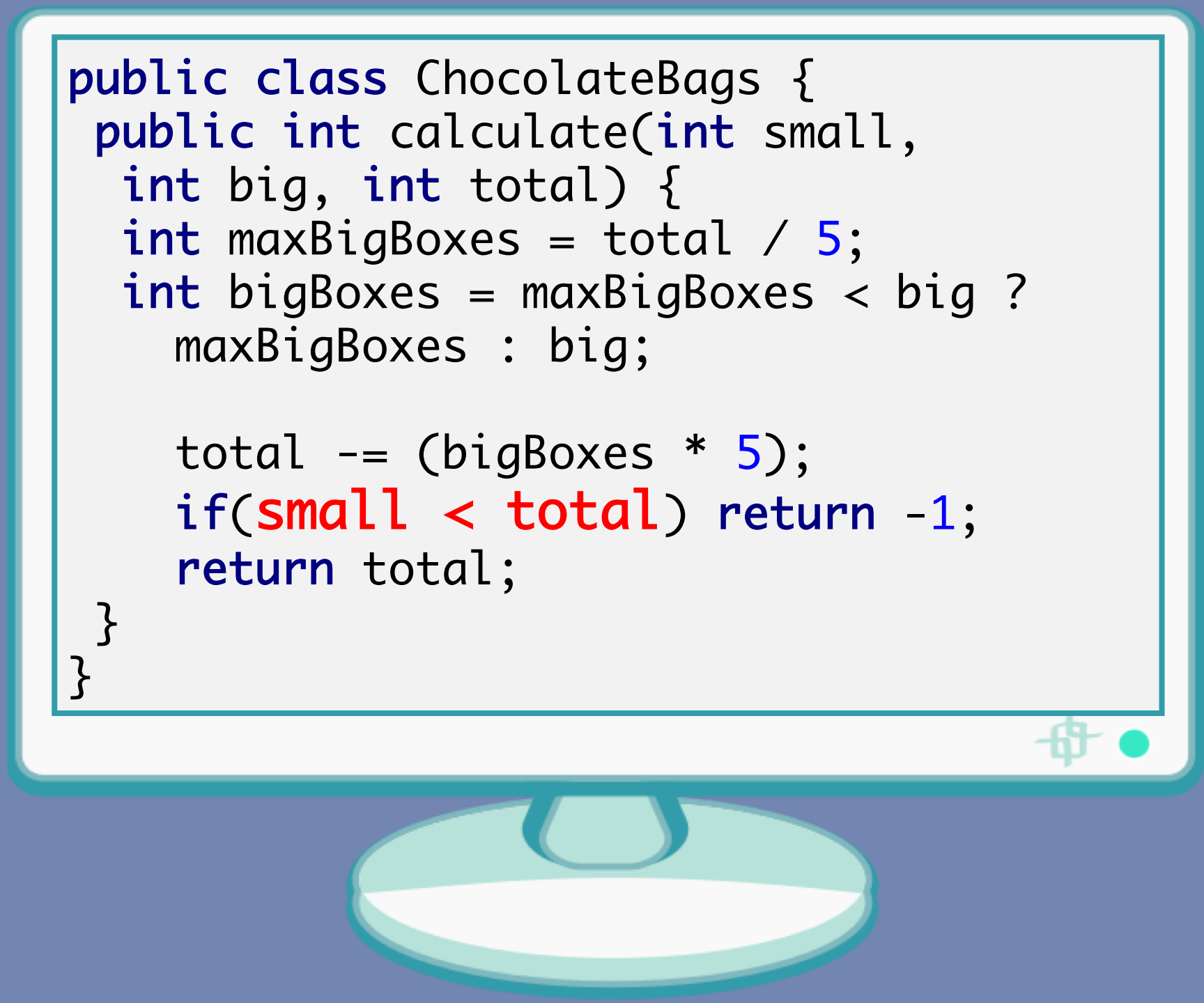
small = 5, **big = 2**, total = 10, = 0

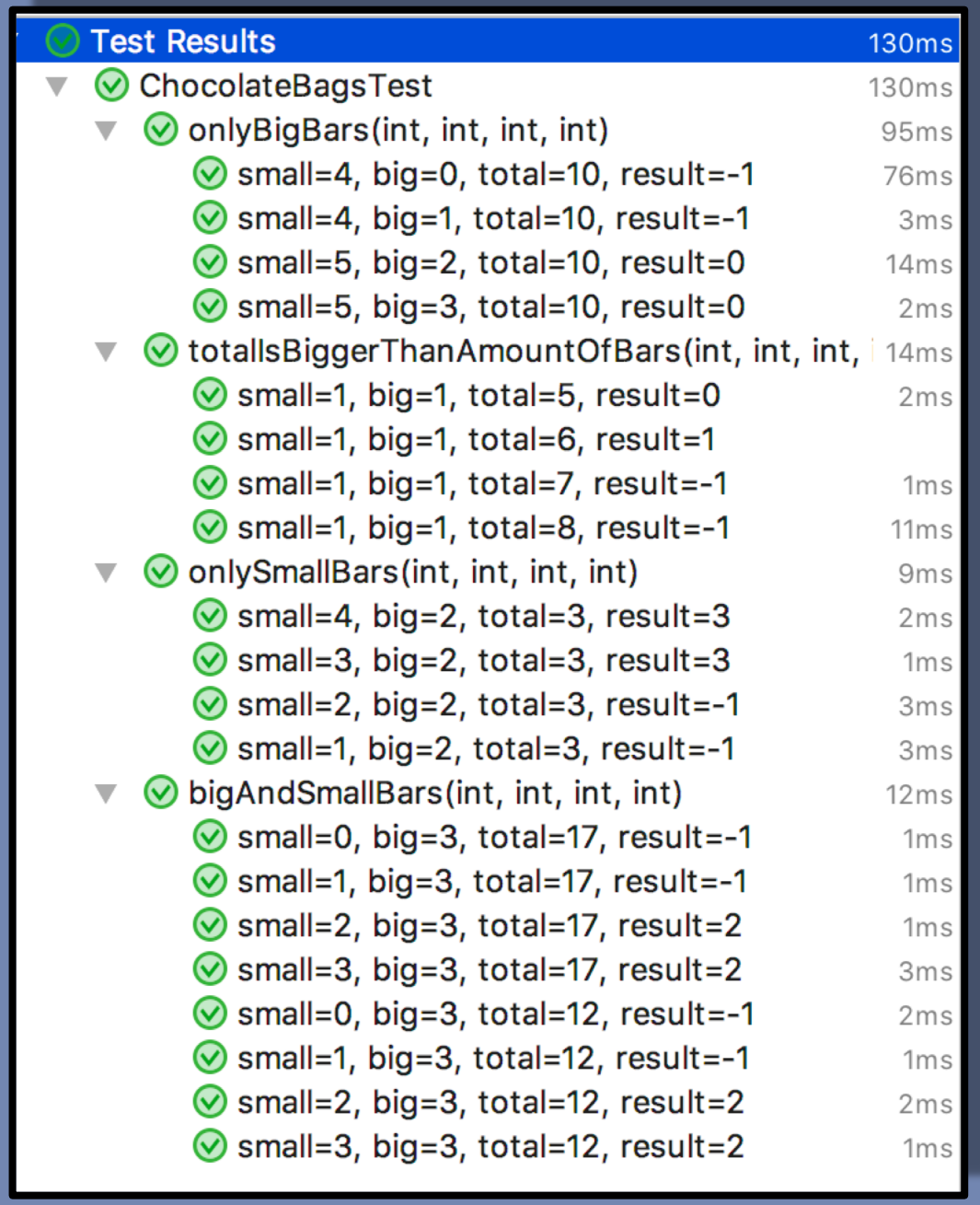
small = 5, **big = 3**, total = 10, = 0


@ParameterizedTest
@CsvSource({ "4,0,10,-1", "4,1,10,-1",
"5,2,10,0", "5,3,10,0" })
public void onlyBigBars(int small, int big,
int total, int expected) {
ChocolateBags bags = new ChocolateBags();
int result =
bags.calculate(small, big, total);
assertEquals(expected, result);
}


Test Results 159ms
ChocolateBagsTest 159ms
onlyBigBars(int, int, int, int) 116ms
small=4, big=0, total=10, result=-1 103ms
small=4, big=1, total=10, result=-1 4ms
small=5, big=2, total=10, result=0 4ms
small=5, big=3, total=10, result=0 5ms
totalIsBiggerThanAmountOfBars(int, int, int, 23ms
small=1, big=1, total=5, result=0 2ms
small=1, big=1, total=6, result=1 19ms
small=1, big=1, total=7, result=-1 1ms
small=1, big=1, total=8, result=-1 1ms
onlySmallBars(int, int, int, int) 6ms
small=4, big=2, total=3, result=3 1ms
small=3, big=2, total=3, result=3
small=2, big=2, total=3, result=-1 4ms
small=1, big=2, total=3, result=-1 1ms
bigAndSmallBars(int, int, int, int) 14ms
small=0, big=3, total=17, result=-1 1ms
small=1, big=3, total=17, result=-1
small=2, big=3, total=17, result=2
small=3, big=3, total=17, result=2 1ms
small=0, big=3, total=12, result=-1 6ms
small=1, big=3, total=12, result=-1 3ms
small=2, big=3, total=12, result=2 3ms
small=3, big=3, total=12, result=2


public class ChocolateBags {
public int calculate(int small,
int big, int total) {
int maxBigBoxes = total / 5;
int bigBoxes = maxBigBoxes < big ?
maxBigBoxes : big;
total -= (bigBoxes * 5);
if(small <= total) return -1;
return total;
}
}


public class ChocolateBags {
public int calculate(int small,
int big, int total) {
int maxBigBoxes = total / 5;
int bigBoxes = maxBigBoxes < big ?
maxBigBoxes : big;
total -= (bigBoxes * 5);
if(small < total) return -1;
return total;
}
}


Test Results 130ms
ChocolateBagsTest 130ms
onlyBigBars(int, int, int, int) 95ms
small=4, big=0, total=10, result=-1 76ms
small=4, big=1, total=10, result=-1 3ms
small=5, big=2, total=10, result=0 14ms
small=5, big=3, total=10, result=0 2ms
totalIsBiggerThanAmountOfBars(int, int, int, 14ms
small=1, big=1, total=5, result=0 2ms
small=1, big=1, total=6, result=1
small=1, big=1, total=7, result=-1 1ms
small=1, big=1, total=8, result=-1 11ms
onlySmallBars(int, int, int, int) 9ms
small=4, big=2, total=3, result=3 2ms
small=3, big=2, total=3, result=3 1ms
small=2, big=2, total=3, result=-1 3ms
small=1, big=2, total=3, result=-1 3ms
bigAndSmallBars(int, int, int, int) 12ms
small=0, big=3, total=17, result=-1 1ms
small=1, big=3, total=17, result=-1 1ms
small=2, big=3, total=17, result=2 1ms
small=3, big=3, total=17, result=2 3ms
small=0, big=3, total=12, result=-1 2ms
small=1, big=3, total=12, result=-1 1ms
small=2, big=3, total=12, result=2 2ms
small=3, big=3, total=12, result=2 1ms