

Search catalog Catalog

Q

For Enterprise

Next

Prev

◀ Back to Week 1

Welcome

X Lessons

Getting Started with BlueJ

Variables and Mathematical Operators

Functions and Conditionals

Classes, Types, and For Each Loops

Seven Steps for Solving Programming Problems

- Solving Programming: A 6 min Seven Step Approach
- Seven Steps in Action:

7 min

4 min

5 min

- Developing an Algorithm Seven Steps in Action: Testing the Algorithm
- Seven Steps in Action: Translating to Code
- Calculating the Perimeter of a Shape
- Quiz: Calculating the
 - 7 questions Perimeter of a Shape
- **End of Module Survey** 5 min

Step 4: Test Algorithm

- 0. Start with totalPerim = 0
- 1. Start with prevPt = the last point
- 2. For each point currPt in the shape,
 - 3. Find distance from prevPt pt to currPt, name it currDist
 - 4. Update totalPerim to be totalPerim + currDist
 - Update prevPt to be currPt.
- 6. total Perim is my answer

Duke

After all, there were a lot we had to do to generalize our steps.

Seven Steps in Action: Testing the Algorithm

Have a guestion? Discuss this lecture in the week forums.

Downloads

English ▼

Lecture Video mp4 Subtitles (English) WebVTT

Transcript (English) txt

Would you like to help us translate the transcript and subtitles into additional languages?

Search Transcript

Interactive Transcript

0:03

Hi. We've developed an algorithm to find the perimeter of a shape. But are we ready to turn this algorithm into code? Well we could, but we'd like to be confident that it's right before we do that. After all, there were a lot we had to do to generalize our steps. And it's entirely possible that we made a mistake. Or perhaps we just didn't think through all the special cases. So before we turn this into code, we should test it out. To test the algorithm, we need a different instance of the problem, something other than what we used to make the algorithm. In fact, it's good if our test instance is pretty different from the one we used to make the algorithm. Here, we've shown a triangle instead of the four-sided trapezoid we used when we developed the algorithm. Before we go any further, take a second to figure out what the right answer is. What is the perimeter of this shape?

0:58

When you finish, you'll want to check if the answer to our simulated algorithm is correct. And to do, that you'll need to know the right answer.

1:08

Now, we'll execute the algorithm by hand for this particular input. As we test the algorithm, notice the similarities between the code and English. We're going to execute this English algorithm by hand, just as we executed code by hand. They both work pretty much the same way. And that's not a coincidence. When you turn this algorithm into code, you want to write down code that has the same semantics as the English. The same meaning. The code should transform the program state in the same way that the English transforms this diagram. So we'll start with totalPerim and set it to 0, and prevPt being the last point in the shape. But what is the last point? We'll say that points in this shape start at the top and go counter-clockwise. So this point in the lower right-hand corner is the last one. And we'll initialized prevPt to be that point.

2:09

You will note briefly that if it is were actual Java object, prevPt might be an error pointing at an object. But we're going to just write down the coordinates here to keep the diagram simple and legible. Next, we're going to do the steps for each point. So, we need to start at the first point, which is this one at the top of the diagram. And that will be the initial value of currPt as we enter the for each repetition. Then, we'll find the distance between these two points, which is 10. And we'll update totalPerim to be 10, 0 plus 10. And then, we'll update prevPt to be currPt.

2:50

Now we're at the end of our for each repetition. So we'll update currPt to be the next point in the shape, which is (-3, -4). When we update currPt, we go back to repeat these steps. We repeat the steps again. Find the distance between those two points, which is 8. And then we update total perimeter to be 10 plus 8, or 18. And then we update prevPt to be the current point.

3:23

Then we go back to the top of our loop, updating currPt.

3:27

We repeat these steps for the last point in our shape. After which we've gone through all the points. So we skip to the steps after our repetition. Here, we can say that totalPerim is our answer.

3:43

TotalPerim is 24. Is that the answer you came up with earlier?

3:50

Yes, that is the perimeter of this shape. The fact that our algorithm came up with the right answer here gives us more confidence that we generalized correctly. We're done executing the algorithm by hand and we have some great confidence that it's correct. So we're ready to turn the algorithm into code.