See the Assessment Guide for information on how to interpret this report.

ASSESSMENT SUMMARY

```
Compilation:  PASSED
API:          PASSED

SpotBugs:     PASSED
PMD:          PASSED
Checkstyle:   PASSED

Correctness:  34/34 tests passed
Memory:       6/6 tests passed
Timing:       18/17 tests passed

Aggregate score: 101.18%
[ Compilation: 5%, API: 5%, Style: 0%, Correctness: 60%, Timing: 10%, Memory: 20% ]
```

ASSESSMENT DETAILS

```
The following files were submitted:
----------------------------------
6.8K Nov 22 10:26 SeamCarver.java


*********************************************************************************
*  COMPILING
*********************************************************************************


% javac SeamCarver.java
*-----------------------------------------------------------


================================================================


Checking the APIs of your programs.
*-----------------------------------------------------------
SeamCarver:

================================================================


*********************************************************************************
*  CHECKING STYLE AND COMMON BUG PATTERNS
*********************************************************************************


% spotbugs *.class
*-----------------------------------------------------------


================================================================


% pmd .
*-----------------------------------------------------------


================================================================


% checkstyle *.java
```

```
*-----------------------------------------------------------

% custom checkstyle checks for SeamCarver.java
*-----------------------------------------------------------


=================================================================


********************************************************************************
*  TESTING CORRECTNESS
********************************************************************************

Testing correctness of SeamCarver
*-----------------------------------------------------------
Running 34 total tests.

Test 1a: check energy() with file inputs
  * 6x5.png
  * 4x6.png
  * 10x12.png
  * 3x7.png
  * 5x6.png
  * 7x3.png
  * 7x10.png
  * 12x10.png
  * stripes.png
  * diagonals.png
  * chameleon.png
  * HJoceanSmall.png
  * 1x8.png
  * 8x1.png
  * 1x1.png
==> passed

Test 1b: check energy() with random pictures
  * 100 random 4-by-6 pictures
  * 100 random 5-by-5 pictures
  * 100 random 6-by-4 pictures
  * 100 random 7-by-10 pictures
  * 10 random 100-by-100 pictures
  * 2 random 250-by-250 pictures
==> passed

Test 1c: check energy() with random pictures in which the RGB components
         of each pixel are in a small range
  * 100 random 4-by-6 pictures
  * 100 random 5-by-5 pictures
  * 100 random 6-by-4 pictures
  * 100 random 7-by-10 pictures
  * 10 random 100-by-100 pictures
  * 2 random 250-by-250 pictures
==> passed

Test 2a: check width() with file inputs
  * 6x5.png
  * 4x6.png
==> passed

Test 2b: check width() with random pictures
  * 10 random 4-by-6 pictures
  * 10 random 5-by-5 pictures
  * 10 random 6-by-4 pictures
  * 10 random 7-by-10 pictures
==> passed

Test 3a: check height() with file inputs
  * 6x5.png
  * 4x6.png
==> passed

Test 3b: check height() with random pictures
  * 10 random 4-by-6 pictures
  * 10 random 5-by-5 pictures
```

```
  * 10 random 6-by-4 pictures
  * 10 random 7-by-10 pictures
==> passed

Test 4a: check findVerticalSeam() with file inputs
  * 6x5.png
  * 4x6.png
  * 10x12.png
  * 3x7.png
  * 5x6.png
  * 7x3.png
  * 7x10.png
  * 12x10.png
  * stripes.png
  * diagonals.png
  * chameleon.png
  * HJoceanSmall.png
  * 1x8.png
  * 8x1.png
  * 1x1.png
==> passed

Test 4b: check findVerticalSeam() with random pictures
  * 100 random 4-by-6 pictures
  * 100 random 5-by-5 pictures
  * 100 random 6-by-4 pictures
  * 100 random 8-by-8 pictures
  * 100 random 7-by-10 pictures
  * 10 random 100-by-100 pictures
  * 2 random 250-by-250 pictures
==> passed

Test 4c: check findVerticalSeam() with random pictures in which
         the RGB values of each pixel are in a small range
  * 100 random 4-by-6 pictures
  * 100 random 5-by-5 pictures
  * 100 random 6-by-4 pictures
  * 100 random 7-by-10 pictures
  * 100 random 8-by-8 pictures
  * 10 random 100-by-100 pictures
  * 2 random 250-by-250 pictures
==> passed

Test 5a: check findHorizontalSeam() with file inputs
  * 6x5.png
  * 4x6.png
  * 10x12.png
  * 3x7.png
  * 5x6.png
  * 7x3.png
  * 7x10.png
  * 12x10.png
  * stripes.png
  * diagonals.png
  * chameleon.png
  * HJoceanSmall.png
  * 1x8.png
  * 8x1.png
  * 1x1.png
==> passed

Test 5b: check findHorizontalSeam() with random pictures
  * 100 random 4-by-6 pictures
  * 100 random 5-by-5 pictures
  * 100 random 6-by-4 pictures
  * 100 random 7-by-10 pictures
  * 100 random 8-by-8 pictures
  * 10 random 100-by-100 pictures
  * 2 random 250-by-250 pictures
==> passed

Test 5c: check findHorizontalSeam() with random pictures in which the RGB
         components of each pixel are in a small range
  * 100 random 4-by-6 pictures
```

```
  * 100 random 5-by-5 pictures
  * 100 random 6-by-4 pictures
  * 100 random 7-by-10 pictures
  * 100 random 8-by-8 pictures
  * 10 random 100-by-100 pictures
  * 2 random 250-by-250 pictures
==> passed

Test 6a: check removeVerticalSeam() with file inputs and optimal seams
  * 6x5.png
  * 10x12.png
  * 3x7.png
  * 5x6.png
  * 7x3.png
  * 7x10.png
  * 12x10.png
  * stripes.png
  * diagonals.png
  * chameleon.png
  * HJoceanSmall.png
  * 8x1.png
==> passed

Test 6b: check removeVerticalSeam() with random pictures and optimal seams
  * 100 random 4-by-6 pictures
  * 100 random 5-by-5 pictures
  * 100 random 6-by-4 pictures
  * 100 random 7-by-10 pictures
  * 10 random 100-by-100 pictures
  * 2 random 250-by-250 pictures
==> passed

Test 6c: check removeVerticalSeam() with file inputs and random seams
  * 6x5.png
  * 10x12.png
  * 3x7.png
  * 5x6.png
  * 7x3.png
  * 7x10.png
  * 12x10.png
  * stripes.png
  * diagonals.png
  * chameleon.png
  * HJoceanSmall.png
  * 8x1.png
==> passed

Test 6d: check removeVerticalSeam() with random pictures and random seams
  * 100 random 4-by-6 pictures
  * 100 random 5-by-5 pictures
  * 100 random 6-by-4 pictures
  * 100 random 7-by-10 pictures
  * 10 random 100-by-100 pictures
  * 2 random 250-by-250 pictures
==> passed

Test 7a: check removeHorizontalSeam() with file inputs and optimal seams
  * 6x5.png
  * 10x12.png
  * 3x7.png
  * 5x6.png
  * 7x3.png
  * 7x10.png
  * 12x10.png
  * stripes.png
  * diagonals.png
  * chameleon.png
  * HJoceanSmall.png
  * 1x8.png
==> passed

Test 7b: check removeHorizontalSeam() with random pictures and optimal seams
  * 100 random 4-by-6 pictures
  * 100 random 5-by-5 pictures
```

```
  * 100 random 6-by-4 pictures
  * 100 random 7-by-10 pictures
  * 10 random 100-by-100 pictures
  * 2 random 250-by-250 pictures
==> passed

Test 7c: check removeHorizontalSeam() with file inputs and random seams
  * 6x5.png
  * 10x12.png
  * 3x7.png
  * 5x6.png
  * 7x3.png
  * 7x10.png
  * 12x10.png
  * stripes.png
  * diagonals.png
  * chameleon.png
  * HJoceanSmall.png
  * 1x8.png
==> passed

Test 7d: check removeHorizontalSeam() with random pictures and random seams
  * 100 random 4-by-6 pictures
  * 100 random 5-by-5 pictures
  * 100 random 6-by-4 pictures
  * 100 random 7-by-10 pictures
  * 10 random 100-by-100 pictures
  * 2 random 250-by-250 pictures
==> passed

Test 8: check energy() with invalid arguments
  * picture = 6x5.png, call energy(-1, 4)
  * picture = 6x5.png, call energy(6, 4)
  * picture = 6x5.png, call energy(5, 5)
  * picture = 6x5.png, call energy(4, -1)
  * picture = 6x5.png, call energy(4, 5)
==> passed

Test 9a: check removeVerticalSeam() with invalid seam
  * picture = 10x10.png
  * picture = 3x7.png
  * picture = 7x3.png
  * picture = 10x12.png
  * picture = 12x10.png
  * picture = 1x8.png
  * picture = 8x1.png
  * picture = 1x1.png
==> passed

Test 9b: check removeHorizontalSeam() with invalid seam
  * picture = 10x10.png
  * picture = 3x7.png
  * picture = 7x3.png
  * picture = 10x12.png
  * picture = 12x10.png
  * picture = 1x8.png
  * picture = 8x1.png
  * picture = 1x1.png
==> passed

Test 9c: check removeHorizontalSeam() and removeVerticalSeam() with null arguments
  * picture = 6x5.png
  * picture = 3x7.png
==> passed

Test 10a: check that client can mutate the Picture object that is passed to the constructor
==> passed

Test 10b: check that client can mutate the Picture object that is returned by picture()
==> passed

Test 11: check constructor with null argument
==> passed
```

```
Test 12a: check intermixed calls to findHorizontalSeam(), findVerticalSeam(),
          removeHorizontalSeam(), and removeVerticalSeam(), width(), height(),
          energy(), and picture() made with probabilities p1, p2, p3, p4, p5,
          p6, p7, and p8, respectively with optimal seams and small images
  * 250 random 5-by-6 images with p = (0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.0, 0.5)
  * 250 random 6-by-5 images with p = (0.0, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.5)
  * 250 random 6-by-6 images with p = (0.0, 0.0, 0.3, 0.3, 0.1, 0.1, 0.1, 0.1)
  * 250 random 6-by-6 images with p = (0.3, 0.0, 0.3, 0.0, 0.0, 0.0, 0.2, 0.2)
  * 250 random 6-by-6 images with p = (0.0, 0.3, 0.0, 0.3, 0.0, 0.0, 0.2, 0.2)
  * 250 random 6-by-6 images with p = (0.1, 0.1, 0.2, 0.2, 0.0, 0.0, 0.0, 0.4)
  * 250 random 6-by-6 images with p = (0.2, 0.2, 0.0, 0.0, 0.2, 0.2, 0.2, 0.0)
  * 250 random 6-by-6 images with p = (0.1, 0.1, 0.2, 0.2, 0.1, 0.1, 0.1, 0.1)
==> passed

Test 12b: check intermixed calls to findHorizontalSeam(), findVerticalSeam(),
          removeHorizontalSeam(), and removeVerticalSeam(), width(), height(),
          energy(), and picture() made with probabilities p1, p2, p3, p4, p5,
          p6, p7, and p8, respectively with optimal seams and medium images
  * 5 random 100-by-110 images with p = (0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.0, 0.5)
  * 5 random 110-by-100 images with p = (0.0, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.5)
  * 5 random 100-by-100 images with p = (0.0, 0.0, 0.3, 0.3, 0.1, 0.1, 0.1, 0.1)
  * 5 random 100-by-100 images with p = (0.3, 0.0, 0.3, 0.0, 0.0, 0.0, 0.2, 0.2)
  * 5 random 100-by-100 images with p = (0.0, 0.3, 0.0, 0.3, 0.0, 0.0, 0.2, 0.2)
  * 5 random 100-by-100 images with p = (0.1, 0.1, 0.2, 0.2, 0.0, 0.0, 0.0, 0.4)
  * 5 random 100-by-100 images with p = (0.2, 0.2, 0.0, 0.0, 0.2, 0.2, 0.2, 0.0)
  * 5 random 100-by-100 images with p = (0.1, 0.1, 0.2, 0.2, 0.1, 0.1, 0.1, 0.1)
==> passed

Test 12c: check intermixed calls to findHorizontalSeam(), findVerticalSeam(),
          removeHorizontalSeam(), and removeVerticalSeam(), width(), height(),
          energy(), and picture() made with probabilities p1, p2, p3, p4, p5,
          p6, p7, and p8, respectively with random seams on small images
  * 250 random 5-by-6 images with p = (0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.0, 0.5)
  * 250 random 6-by-5 images with p = (0.0, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.5)
  * 250 random 6-by-6 images with p = (0.1, 0.1, 0.2, 0.2, 0.0, 0.0, 0.0, 0.4)
  * 250 random 6-by-6 images with p = (0.2, 0.2, 0.0, 0.0, 0.2, 0.2, 0.2, 0.0)
  * 250 random 6-by-6 images with p = (0.1, 0.1, 0.2, 0.2, 0.1, 0.1, 0.1, 0.1)
==> passed

Test 12d: check intermixed calls to findHorizontalSeam(), findVerticalSeam(),
          removeHorizontalSeam(), and removeVerticalSeam(), width(), height(),
          energy(), and picture() made with probabilities p1, p2, p3, p4, p5,
          p6, p7, and p8, respectively with random seams on medium images
  * 10 random 100-by-110 images with p = (0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.0, 0.5)
  * 10 random 110-by-100 images with p = (0.0, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.5)
  * 10 random 110-by-110 images with p = (0.1, 0.1, 0.2, 0.2, 0.0, 0.0, 0.0, 0.4)
  * 10 random 100-by-100 images with p = (0.2, 0.2, 0.0, 0.0, 0.1, 0.1, 0.2, 0.2)
  * 10 random 110-by-110 images with p = (0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.2, 0.2)
==> passed

Test 12e: check intermixed calls to findHorizontalSeam(), findVerticalSeam(),
          removeHorizontalSeam(), and removeVerticalSeam(), width(), height(),
          energy(), and picture() made with probabilities p1, p2, p3, p4, p5,
          p6, p7, and p8, respectively with optimal seams
          (tests corner cases when width = 1 or 2 and/or height = 1 or 2)
  * 50 random 1-by-8 images with p = (0.1, 0.1, 0.2, 0.0, 0.1, 0.1, 0.2, 0.2)
  * 50 random 8-by-1 images with p = (0.1, 0.1, 0.0, 0.2, 0.1, 0.1, 0.2, 0.2)
  * 50 random 1-by-1 images with p = (0.2, 0.2, 0.0, 0.0, 0.1, 0.1, 0.2, 0.2)
  * 50 random 2-by-8 images with p = (0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.2, 0.2)
  * 50 random 8-by-2 images with p = (0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.2, 0.2)
  * 50 random 2-by-2 images with p = (0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.2, 0.2)
==> passed

Test 12f: check intermixed calls to removeHorizontalSeam(), and removeVerticalSeam(),
          and picture(), with optimal or invalid seams on small images
  * 250 random 5-by-6 images with p = (0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.0, 0.5)
  * 250 random 6-by-5 images with p = (0.0, 0.0, 0.0, 0.5, 0.0, 0.0, 0.0, 0.5)
  * 250 random 6-by-6 images with p = (0.0, 0.0, 0.3, 0.3, 0.0, 0.0, 0.0, 0.4)
==> passed


Total: 34/34 tests passed!
```

```
================================================================
********************************************************************************
*  MEMORY
********************************************************************************

Analyzing memory of SeamCarver
*-----------------------------------------------------------
Running 6 total tests.

Memory usage of a SeamCarver after removing 2 horizontal
and 2 vertical seams from an n-by-n image.

Maximum allowed memory is ~ 12 n^2 bytes.

                 n       student (bytes)
------------------------------------------
=> passed       16         3928
=> passed       32        10328
=> passed       64        35416
=> passed      128       134752
=> passed      256       530016
=> passed      512      2106976
==> 6/6 tests passed

Total: 6/6 tests passed!

Estimated student memory (bytes) = 8.00 n^2 + 16.07 n + 1622.28   (R^2 = 1.000)

================================================================



********************************************************************************
*  TIMING
********************************************************************************

Timing SeamCarver
*-----------------------------------------------------------
Reference solution is unoptimized.

Running 17 total tests.

Test 1: create a SeamCarver object for a given 736-by-584 picture;
        then call findHorizontalSeam(), removeHorizontalSeam(),
        findVerticalSeam(), removeVerticalSeam(), and picture()
        one each; count total number of calls to methods in Picture
  * constructor calls        = 4
  * get()    calls per pixel = 9.9
  * set()    calls per pixel = 2.0
  * getRGB() calls per pixel = 0.0
  * setRGB() calls per pixel = 0.0
==> passed

Test 2: create a SeamCarver object for a given 736-by-584 picture;
        then call findHorizontalSeam(), removeHorizontalSeam(),
        findVerticalSeam(), and removeVerticalSeam(), and picture();
        once each; count total number of calls to methods in Color
  * constructor calls per pixel = 9.9
  * getRed()    calls per pixel = 7.9
  * getGreen()  calls per pixel = 7.9
  * getBlue()   calls per pixel = 7.9
  * getRGB()    calls per pixel = 0.0
  * equal number of calls to getRed(), getGreen(), and getBlue()
==> passed

Tests 3a-3c: time removeVerticalSeam() for a given 736-by-584 picture
  * student   solution calls per second:      70.10
  * reference solution calls per second:      68.87
  * reference / student ratio:                 0.98

=> passed      student <= 150.0x reference
=> passed      student <=  15.0x reference
=> passed      student <=   4.5x reference
```

```
Tests 4a-4c: time findVerticalSeam() and removeVerticalSeam()
             for a given 736-by-584 picture
  * student   solution calls per second:      13.83
  * reference solution calls per second:      10.39
  * reference / student ratio:                 0.75

=> passed      student <= 150.0x reference
=> passed      student <=  15.0x reference
=> passed      student <=   2.3x reference

Tests 5a-5c: time removeHorizontalSeam() for a given 736-by-584 picture
  * student   solution calls per second:      54.99
  * reference solution calls per second:      18.43
  * reference / student ratio:                 0.34

=> passed      student <= 150.0x reference
=> passed      student <=  15.0x reference
=> passed      student <=   4.5x reference

Tests 6a-6c: time findHorizontalSeam() and removeHorizontalSeam()
             for a given 736-by-584 picture
  * student   solution calls per second:      12.66
  * reference solution calls per second:       6.26
  * reference / student ratio:                 0.49

=> passed      student <= 150.0x reference
=> passed      student <=  15.0x reference
=> passed      student <=   2.3x reference

Tests 7a-7c: time findHorizontalSeam(), removeHorizontalSeam(), findVerticalSeam(),
             and removeVerticalSeam() for a given 736-by-584 picture
  * student   solution calls per second:       6.72
  * reference solution calls per second:       4.02
  * reference / student ratio:                 0.60

=> passed      student <= 150.0x reference
=> passed      student <=  15.0x reference
=> passed      student <=   1.5x reference
=> optimized   student <=   0.8x reference

Total: 18/17 tests passed!


================================================================
```