knóldus

Profile Picture

knól[x]

Question

We want to change our profile picture on Facediary. Now Facediary has some restriction over the dimension of picture that we can upload.

Minimum dimension of the picture can be L x L, where L is the length of the side of square. Now we have photos of various dimensions. Dimension of a photo is denoted as W x H where W - width of the photo and H - height of the photo.

When any photo is uploaded following events may occur:

1. If any of the width or height is less than L, user is prompted to upload another one. Print in this case.
2. If width and height, both are large enough and if the photo is already square then it is accepted. Print in this case. Else user is prompted to crop it. Print in this case.

Given L, N, W and H as input, print appropriate text as output.

Sample Input: L = 180, **W** = 640, **H** = 480
Sample Output: CROP IT

Program

Main.scala | ProfilePictureTest.scala | build.sbt

```scala
import scala.io.StdIn.readInt

class ProfilePicture {
  //This method is used to know the status of photo such as accepted, crop or upload other
  def setProfilePicture(pictureLength: Int, photoWidth: Int, photoHeight: Int): String = {
    if (photoWidth < pictureLength || photoHeight < pictureLength) {
      return "Upload another photo"
    }
    else if (photoWidth >= pictureLength && photoHeight >= pictureLength && photoWidth == photoHeight) {
      return "Accepted"
    }
    else {
      return "CROP IT"
    }
  }
}

```

Program

```
18  //singleton object
19  object Main extends App {
20    print("Enter the Length of Picture :")
21    private val pictureLength = readInt()
22    print("Enter the Width of Photo :")
23    private val photoWidth = readInt()
24    print("Enter the Height of Photo :")
25    private val photoHeight = readInt()
26    private val obj = new ProfilePicture
27    try {
28      if (pictureLength <= 0 || photoWidth <= 0 || photoHeight <= 0) {
29        throw new ArithmeticException
30      }
31      val result: String = obj.setProfilePicture(pictureLength: Int, photoWidth: Int, photoHeight: Int)
32      print(result)
33    }
34    catch {
35      case ex: ArithmeticException => {
36        print("Number should be greater than zero")
37      }
38    }
39  }
40
```

Test Case

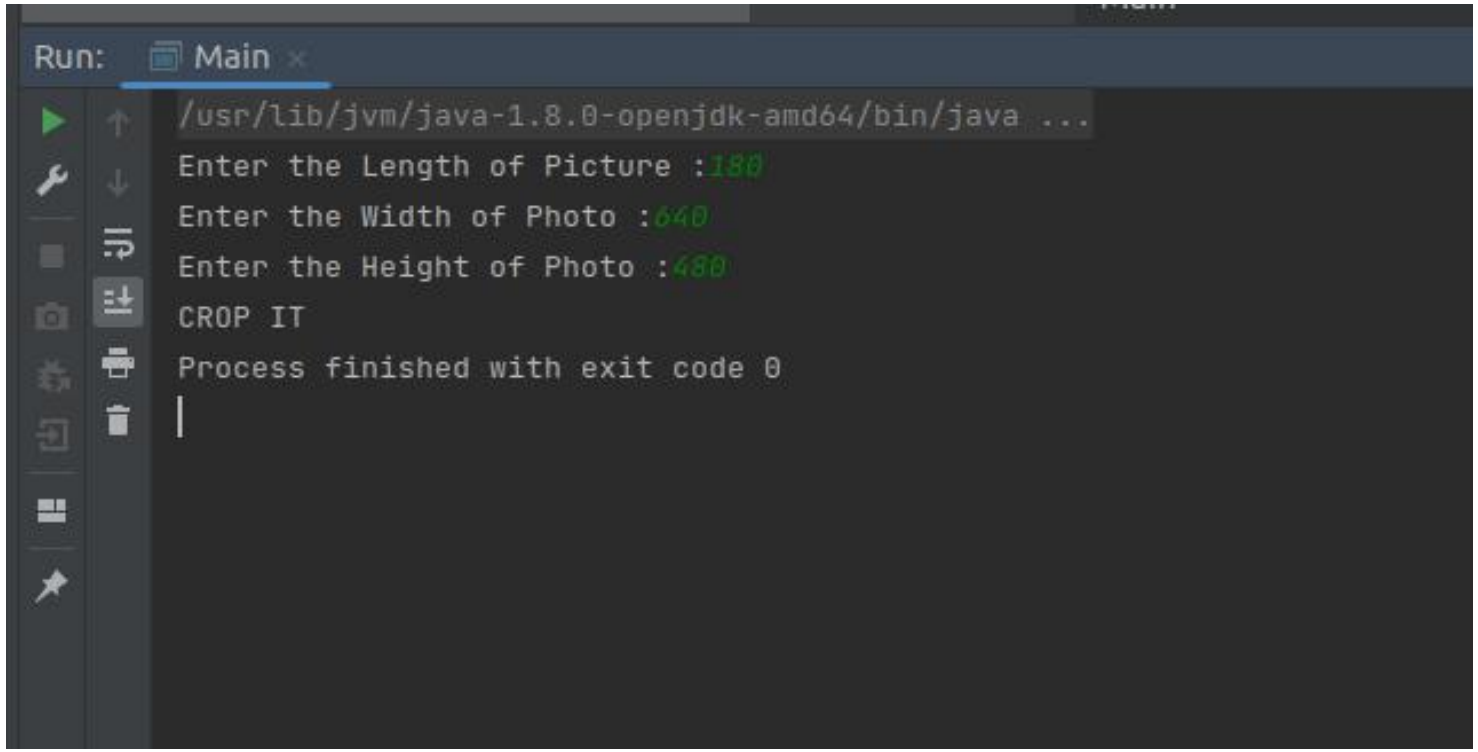
Main.scala | ProfilePictureTest.scala | build.sbt

```scala
import org.scalatest.flatspec.AnyFlatSpec

class ProfilePictureTest extends AnyFlatSpec{
  val obj =new ProfilePicture
  it should "match with width and height grater than length of picture " in{
    assert( condition = "CROP IT"==obj.setProfilePicture( pictureLength = 180, photoWidth = 640, photoHeight = 480))
  }
  it should "match with width and height less than length of picture" in{
    assert( condition = "Upload another photo"==obj.setProfilePicture( pictureLength = 180, photoWidth = 90, photoHeight = 90))
  }
  it should "match with same width and height of photo" in{
    assert( condition = "Accepted"==obj.setProfilePicture( pictureLength = 180, photoWidth = 200, photoHeight = 200))
  }
}

```

Output


Run:
Main
/usr/lib/jvm/java-1.8.0-openjdk-amd64/bin/java ...
Enter the Length of Picture :180
Enter the Width of Photo :640
Enter the Height of Photo :480
CROP IT
Process finished with exit code 0

knóldus

THANK YOU

knólˣ