Build a Polygon Area Calculator Project

In this project you will use object oriented programming to create a Rectangle class and a Square class. The Square class should be a subclass of Rectangle, and inherit its methods and attributes.

Rectangle class

When a Rectangle object is created, it should be initialized with width and height attributes. The class should also contain the following methods:

- set_width
- set_height
- get_area: Returns area (width * height)
- get_perimeter: Returns perimeter (2 * width + 2 * height)
- get_diagonal: Returns diagonal ((width ** 2 + height ** 2) ** .5)
- get_picture: Returns a string that represents the shape using lines of '*'. The number of lines should be equal to the height and the number of '*' in each line should be equal to the width. There should be a new line (\n) at the end of each line. If the width or height is larger than 50, this should return the string: 'Too big for picture.'.
- get_amount_inside: Takes another shape (square or rectangle) as an argument. Returns the number of times the passed in shape could fit inside the shape (with no rotations). For instance, a rectangle with a width of 4 and a height of 8 could fit in two squares with sides of 4.

Additionally, if an instance of a Rectangle is represented as a string, it should look like: 'Rectangle(width=5, height=10)'.

Square class

The Square class should be a subclass of Rectangle. When a Square object is created, a single side length is passed in. The __init__ method should store the side length in both the width and height attributes from the Rectangle class.

The Square class should be able to access the Rectangle class methods but should also contain a set_side method. If an instance of a Square is represented as a string, it should look like: 'Square(side=9)'.

Additionally, the set_width and set_height methods on the Square class should set both the width and height.

Usage example

```
rect = Rectangle(10, 5)
print(rect.get_area())
```

```
rect.set_height(3)
print(rect.get_perimeter())
print(rect)
print(rect.get_picture())

sq = Square(9)
print(sq.get_area())
sq.set_side(4)
print(sq.get_diagonal())
print(sq)
print(sq.get_picture())

rect.set_height(8)
rect.set_width(16)
print(rect.get_amount_inside(sq))
```

That code should return:

```
50
26
Rectangle(width=10, height=3)
**********
**********
**********

81
5.656854249492381
Square(side=4)
****
****
****
****

8
```

Tests

1. The Square class should be a subclass of the Rectangle class.
2. The Square class should be a distinct class from the Rectangle class.
3. A square object should be an instance of the Square class and the Rectangle class.
4. The string representation of Rectangle(3, 6) should be 'Rectangle(width=3, height=6)'.
5. The string representation of Square(5) should be 'Square(side=5)'.
6. Rectangle(3, 6).get_area() should return 18.
7. Square(5).get_area() should return 25.
8. Rectangle(3, 6).get_perimeter() should return 18.

9. Square(5).get_perimeter() should return 20.
10. Rectangle(3, 6).get_diagonal() should return 6.708203932499369.
11. Square(5).get_diagonal() should return 7.0710678118654755.
12. An instance of the Rectangle class should have a different string representation after setting new values.
13. An instance of the Square class should have a different string representation after setting new values by using .set_side().
14. An instance of the Square class should have a different string representation after setting new values by using .set_width() or set_height().
15. The .get_picture() method should return a different string representation of a Rectangle instance.
16. The .get_picture() method should return a different string representation of a Square instance.
17. The .get_picture() method should return the string 'Too big for picture.' if the width or height attributes are larger than 50.
18. Rectangle(15,10).get_amount_inside(Square(5)) should return 6.
19. Rectangle(4,8).get_amount_inside(Rectangle(3, 6)) should return 1.
20. Rectangle(2,3).get_amount_inside(Rectangle(3, 6)) should return 0.