

1. Μία διαδικτυακή εφαρμογή τρέχει σε τρεις εξυπηρετητές: (α) στον εξυπηρετητή ιστού (web server), (β) στον εξυπηρετητή εφαρμογών (application server) και (γ) στον εξυπηρετητή βάσεων δεδομένων (database server). Η εφαρμογή είναι διαθέσιμη εκτός του εσωτερικού δικτύου μέσω ιστοσελίδων που παρέχονται από τον εξυπηρετητή ιστού. Προκειμένου οι εξωτερικοί χρήστες να επικοινωνήσουν με την εφαρμογή θα πρέπει να χρησιμοποιήσουν έναν browser (thin client). Οι αιτήσεις τους περνάνε από το τοίχος προστασίας (firewall) πριν φτάσουν στον εξυπηρετητή ιστού. Τέλος για τους χρήστες στο εσωτερικό δίκτυο υπάρχει απευθείας πρόσβαση στον εξυπηρετητή εφαρμογών μέσω εφαρμογών (rich client) και όχι μέσω του browser. Οι αιτήσεις των εσωτερικών χρηστών δεν περνάνε από το τοίχος προστασίας. **Δώστε ένα διάγραμμα ανάπτυξης (deployment diagram) για την πιο πάνω περιγραφή δείχνοντας τους διακριτούς κόμβους του συστήματος και τους συνδέσμους μεταξύ τους (5 μονάδες).**
2. Δίνεται η κλάση Square για ένα τετράγωνο και η κλάση SquareTest. **Συμπληρώστε τις μεθόδους ελέγχου της SquareTest ώστε να περνούν όλοι οι έλεγχοι.** Χρησιμοποιείστε σε όλους τους ελέγχους το Square square που δημιουργείται πριν από κάθε έλεγχο στην μέθοδο setUp με την επισήμανση BeforeEach. Για τον έλεγχο της setSide θέστε την πλευρά σε μία τιμή της αρεσκείας σας. Για τον έλεγχο της area() η αναμενόμενη τιμή του εμβαδού είναι 25 ακριβώς. Για τον έλεγχο της perimeter η αναμενόμενη τιμή της περιμέτρου είναι 20 ακριβώς. Για τον έλεγχο της diagonal() η αναμενόμενη τιμή της διαγωνίου είναι 7.07 με ακρίβεια 2 δεκαδικών ψηφίων (η πραγματική τιμή είναι 7.0710678118654755). **(5 μονάδες)**

```

public class Square {
    private double side;

    public Square(double side) {
        this.side = side;
    }

    public double getSide() {
        return side;
    }

    public void setSide(double side) {
        this.side = side;
    }

    public double area() {
        return side*side;
    }

    public double perimeter() {
        return 4.0*side;
    }

    public double diagonal() {
        return side * Math.sqrt(2.0);
    }
}

import static
org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.*;

class SquareTest {
    Square square;

    @BeforeEach
    void setUp() {
        square = new Square(5.0);
    }

    @Test
    void getSide() {
        ...
    }

    @Test
    void setSide() {
        ...
    }

    @Test
    void area() {
        ...
    }

    @Test
    void perimeter() {
        ...
    }

    @Test
    void diagonal() {
        ...
    }
}

```