

Drikkepenge

Skriv en metode, som præsenterer brugeren for følgende menu på skærmen:

1. Land
2. Service
3. Beløb
4. Beregn drikkepenge

Metoden skal:

- udskrive menuen på skærmen
- læse brugerens valg (fra tastaturet)

Skriv derpå en metode, der ud fra brugerens valg returnerer hvor mange drikkepenge man bør give. Et data-sæt kunne være "Danmark;Restaurent;10", "Danmark;Tankstation;0", "USA;Tankstation;15". Find selv på flere.

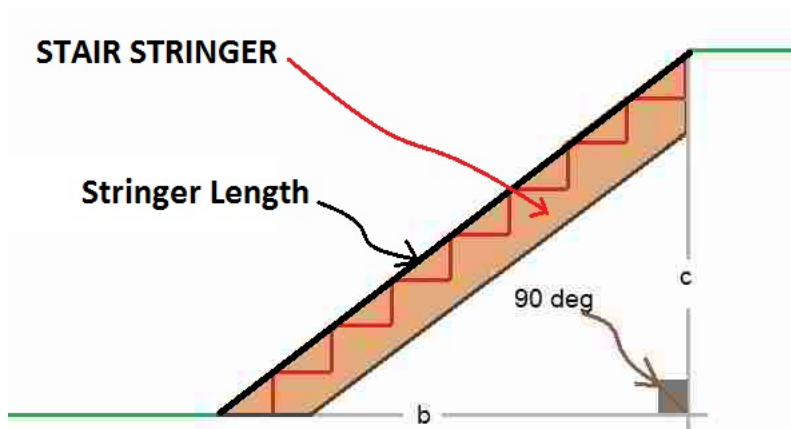
Metoden skal kaldes ved punkt 4 ovenfor.

Trappeligning

Skriv en metode, som kan beregne hvor mange trin en trappe skal have ud fra afstanden fra trappens start ind til væggen (b) og højden (c). Et idealtrin har en højde på ca 15 cm og en bredde på ca 31 cm. Brug Pythagoras til at udregne ideal-hypotenusen.

Metoden skal minimum kunne returnere det antal trin der kommer tættest på en trappe bestående af x antal idealtrin.

Dernæst kunne man lægge en tolerance ind så metoden kaster en exception hvis forholdet mellem b og c gør det umuligt at bygge en trappe med idealtrin der holder sig indenfor tolerancen.



Gåden

Skriv en klasse Riddle som har nedenstående testklasse.

Læg mærke til at du skal skrive din egen exception - AlreadySolvedException.

```
import org.junit.Before;
import org.junit.Test;

import static org.junit.Assert.assertTrue;

public class RiddleTest {
    Riddle myRiddle, myRiddle2;
    int solution;
    int level;
    String message;
    String name;
    String userGuess;

    @Before
    public void setUp() throws Exception {
        solution=8;
        level=1;
        name = "sequence of numbers ..";
        myRiddle = new Riddle(level, name);
        message = "What is the missing number: 1 1 2 3 5 _ 13 ";
        myRiddle.setMessage(message);
        myRiddle.setSolution(Integer.toString(solution));
        myRiddle2 = new Riddle(level, name);
        myRiddle2.setMessage(message);
        myRiddle2.setSolution(Integer.toString(solution));
        userGuess="8";
        myRiddle2.solveRiddle(userGuess);
    }

    @Test(expected = AlreadySolvedException.class)
    public void solveRiddleException() throws AlreadySolvedException {
        userGuess="8";
        myRiddle2.solveRiddle(userGuess);
    }

    @Test
    public void solveRiddle() throws AlreadySolvedException {
        userGuess="8";
        boolean expected = myRiddle.solveRiddle(userGuess);
        assertTrue(expected);
    }
}
```