Care este rezultatul in urma rularii codului urmator?

public class Person {  
  
 private static Person *person* = null;  
 public String s;  
 private Person(String name)  
 {  
 s = "Hello, my name is" + " " + name;  
 }  
 public static Person getInstance(String name)  
 {  
 if (*person* == null)  
 *person* = new Person(name);  
  
 return *person*;  
 }  
}

class Main  
{  
 public static void main(String args[])  
 {  
  
 Person x = Person.*getInstance*("Ana");  
 Person y = Person.*getInstance*("Andrei");  
 Person z = Person.*getInstance*("Maria");  
  
  
 x.s = (x.s).toUpperCase();  
  
 System.*out*.println(x.s +"; "+ y.s +"; "+ (z.s).toLowerCase()+"; "+ x.s);  
   
 }  
}

Variante de raspuns:

1. HELLO, MY NAME IS ANA; Hello, my name is Andrei; hello, my name is maria; Hello, my name is Ana
2. HELLO, MY NAME IS ANA; HELLO, MY NAME IS ANDREI; hello, my name is maria; hello, my name is ana
3. **HELLO, MY NAME IS ANA; HELLO, MY NAME IS ANA; hello, my name is ana; HELLO, MY NAME IS ANA**
4. HELLO, MY NAME IS ANA; Hello, my name is Ana; hello, my name is Ana; Hello, my name is Ana

Justificare raspuns:

Cand se apeleaza prima data metoda getInstance(“Ana”) din clasa Person este creat un obiect al clasei cu numele person, fiind schimbat din null. Apoi, cand este apelat getInstance(“Andrei”), deoarece obiectul person nu mai este null, este returnat variabilei in loc sa fie instantiat iar.

In main, clasa Person este instantiata cu trei obiecte (x,y,z) prin apelul la metoda statica getInstance(nume). Dupa crearea primului obiect (x), celelalte variabile (y si z) primesc un pointer catre obiectul deja creat (x). Astfel, daca este schimbata variabila obiectului x, schimbarea se reflecta si asupra celorlalte variabile: y si z.

Referinte web:

https://refactoring.guru/design-patterns/singleton

Dificultate:

(Easy, **Medium**, Difficult)