**Student propunător:** Viski Paul Darius,

grupa 227/2, e-mail vpir2639@scs.ubbcluj.ro

***Ce se afișează la rularea codului următor? Explicați.***

**class** Audi {  
 **int speed**;  
  
 **public** Audi(**int** speed) { **this**.**speed** = speed; }  
  
 **public int** getSpeed() { **return this**.**speed**; }  
  
 **public** String toString(){ **return speed** + **""**; }  
}  
  
**class** tunedAudi **extends** Audi {  
 **public** tunedAudi(**int** speed) { **super**(speed + 30); }  
  
 **public int** getSpeed() { **return this**.**speed**; }  
}  
  
**public class** Application {  
 **public static void** main(String[] args) {  
 Audi a1 = **new** Audi(200);  
 Audi a2 = **new** tunedAudi(200);  
 tunedAudi a3 = **new** tunedAudi(200);  
 Audi a4 = **new** Audi(200);  
  
 **long** topSpeed200 = Arrays.*asList*(a1.toString(), a2.toString(), a3.toString()).stream().filter((s) -> s.matches(**"200"**)).count();  
 System.***out***.println(topSpeed200);  
 }  
}

**Variante de răspuns:**

1. true
2. eroare
3. 2
4. **1**

Explicații: Doar obiectul a1 din listă verifică condiția (speed să fie egal cu 200, printr-o comparație de string-uri). Se creează și obiectul a4 care ar trebui să verifice, dar acesta nu apare în listă, și automat nu se ia în considerare. Iar obiectele de tipul tunedAudi au speed=230. La final apare count() care numără aparițiile, deci nicidecum aplicația nu ar trebui să afișeze o valoare boolean.