

1. Egy osztály több interfészt is megvalósíthat. **IGAZ**
2. Az A osztálynak leszármazottja a nem absztrakt B osztály. B-nek minden olyan metódust meg kell valósítania, ami A-ban abstract. **IGAZ**
3. final osztálynak kell, hogy legyen final módosítójú metódusa. **HAMIS**
4. Egy osztálynak legfeljebb egy őse lehet. **HAMIS**
5. Egy osztálynak több közvetlen őse lehet. **HAMIS**
6. Ha egy s String objektumon meghívjuk a toLowerCase() metódust, akkor nem módosul az s objektum tartalma. **IGAZ**
7. Az A osztálynak leszármazottja a nem final B osztály. B-nek lehet abstract metódusa. **IGAZ**
8. Konstruktornak nem lehet private láthatósága. **HAMIS**
9. Egy osztálynak legfeljebb egy közvetlen őse lehet. **IGAZ**
10. abstract osztálynak nem lehet final módosítójú metódusa. **HAMIS**
11. Lehet olyan osztályt írni, amelyiknek nincs az osztályon kívülről meghívható String toString() metódusa. **HAMIS**
12. Konstruktornak nem lehet láthatósága (private, public, stb). **HAMIS**
13. Az A osztálynak leszármazottja a nem final B osztály. B-nek minden olyan metódust meg kell valósítania, ami A-ban abstract. **HAMIS**
14. Statikus metódusban használható a this változó. **HAMIS**
15. Egy metódus nem lehet egyszerre abstract és final. **IGAZ**
16. Minden osztálynak van az osztályon kívülről meghívható String toString() metódusa. **IGAZ**
17. Minden osztálynak van az osztályon kívülről meghívható boolean equals(Object o) metódusa. **IGAZ**
18. Statikus metódusban nem használható a this változó. **IGAZ**
19. A String osztálynak nincs olyan metódusa, amely a meghívott String objektum tartalmát módosítja. **IGAZ**
20. Tömb létrehozásakor minden tömbelem az alapértelmezett (paraméter nélküli) konstruktorral inicializálódik. **HAMIS**
21. A String osztálynak van olyan metódusa, amely a meghívott String objektum tartalmát módosítja. **HAMIS**
22. Konstruktornak nem lehet protected láthatósága. **HAMIS**
23. Ha egy s String objektumon meghívjuk a toUpperCase() metódust, akkor módosul az s objektum tartalma **HAMIS**
24. Egy osztály lehet akkor is absztrakt, ha nincs absztrakt metódusa. **IGAZ**
25. Egy osztály nem lehet final, ha van absztrakt metódusa. **IGAZ**
26. abstract osztálynak lehet final módosítójú metódusa. **IGAZ**
27. Egy osztály nem lehet absztrakt, ha nincs absztrakt metódusa. **HAMIS**
28. Primitív típus tömbje is a primitív típusok közé számít. **HAMIS**
29. Egy metódus nem lehet egyszerre protected és final. **HAMIS**