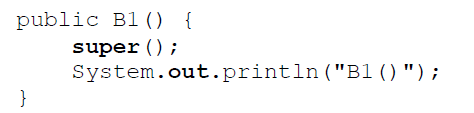
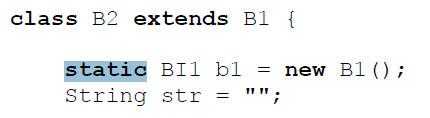
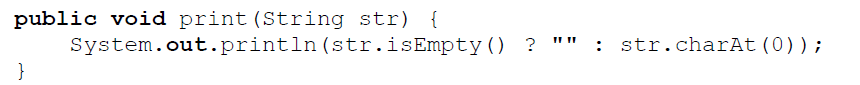
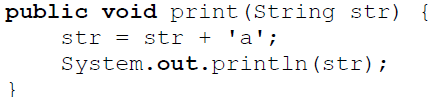
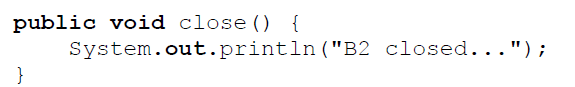
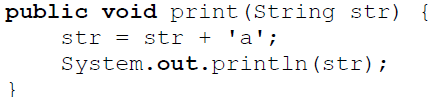
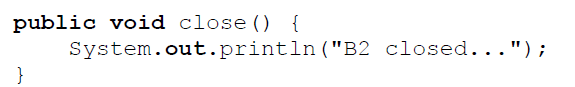
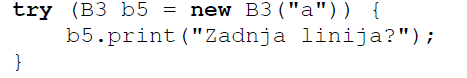
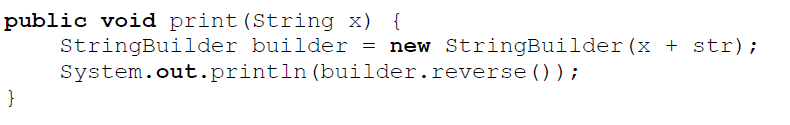
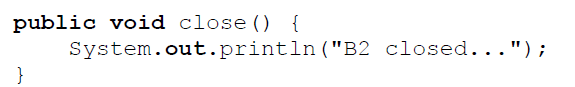
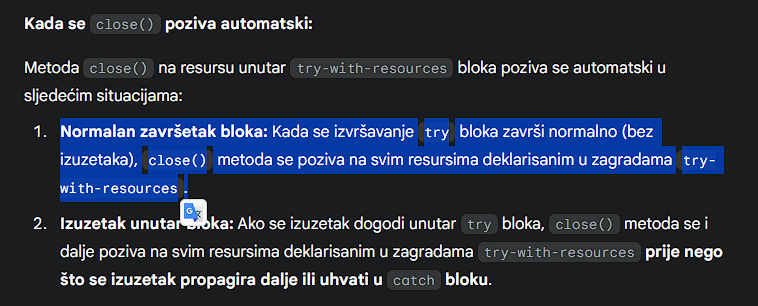
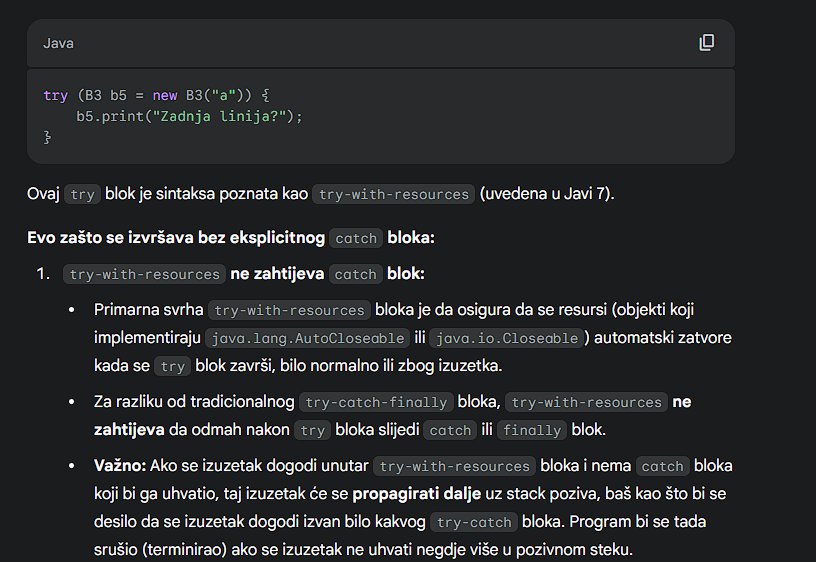
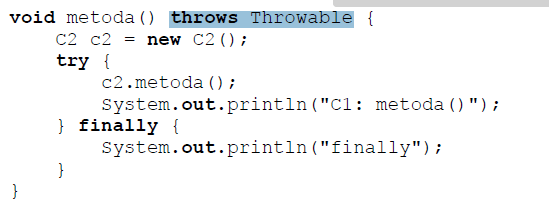
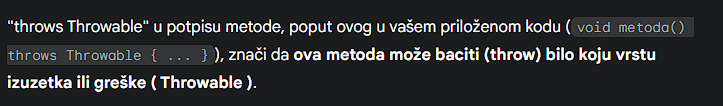
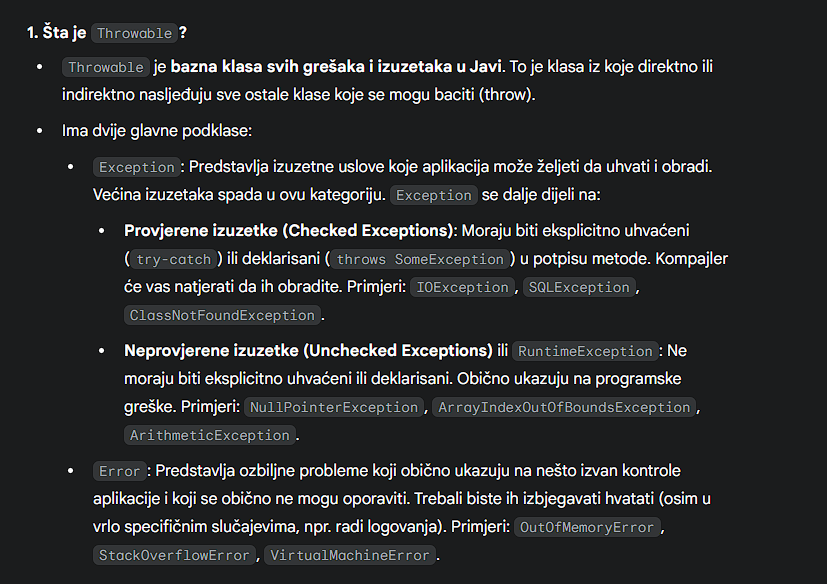
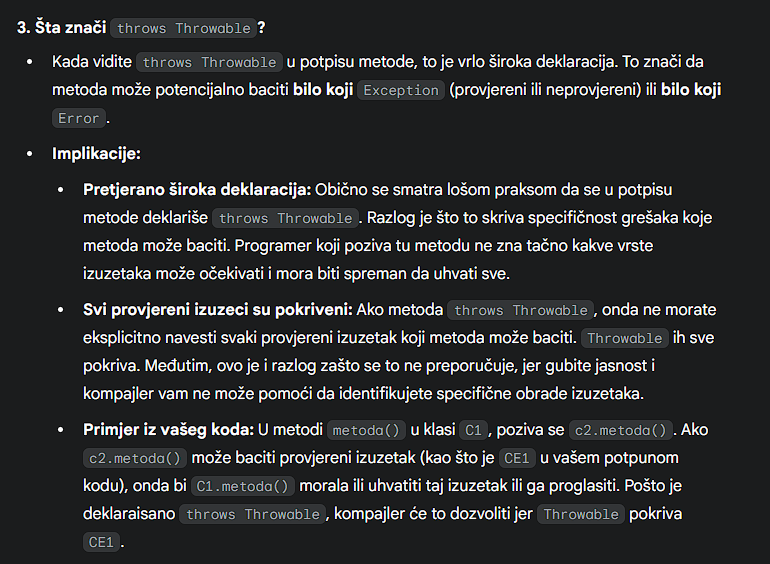
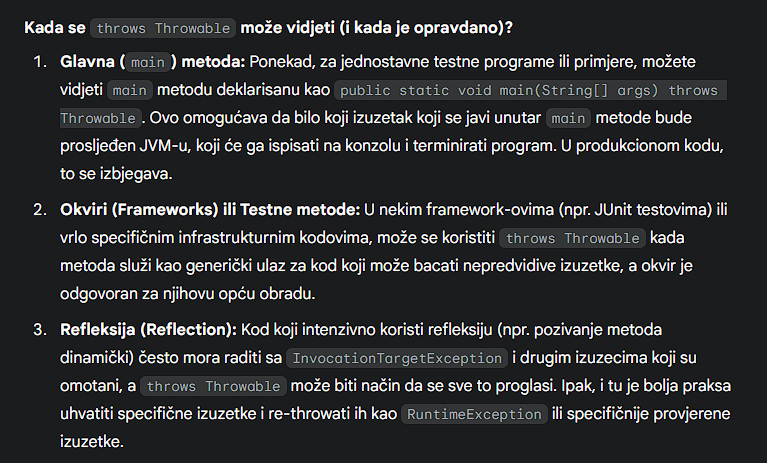
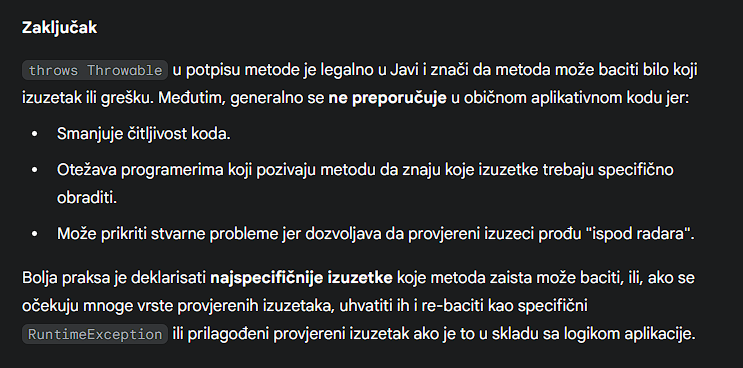
1. b)

B1 b1 = new B1();  
  
  
B1 b2 = new B2();  
  
Zatim se pozivaju redom super() -B1, a nakon toga B2()!!  
Znači, zbog “b1” imamo B1() prije super unutar B2 Constructor-a imamo B1() te onda imamo još B2() zbog B2 Constructor-a!!  
***b1 ima priroritet nad super() jer on okarakterisan sa static!!***  
  
BI1 b3 = **new** B3("b3");  
Poziva se super() unutar B2 u kojem se poziva super() od B1 tako da je ispis redom:  
B1() B2() ali pazi, “b1” UNUTAR B2 JE “static” TAKO DA SE ON KREIRAO JEDNOM I TO JE TO..  
  
BI1 b3 = **new** B3("b3");  
Poziva se super() unutar B2 u kojem se poziva super() od B1 tako da je ispis redom:  
B1() B2() ali pazi, “b1” UNUTAR B2 JE “static” TAKO DA SE ON KREIRAO JEDNOM I TO JE TO..  
  
  
b1.print(b1.j);  
  
  
  
  
b2.print("av");  
  
ZBOG POLIMORFIZMA!  
U klasi B2, imamo implementiranu metodu imena “print”!!   
Iako je referenciramo iz njoj roditeljske klase!!  
  
b2.close();  
  
  
b4.print("b3");  
  
  
((BI1) b3).close();  
  
jer B3 extends B2, a metoda close() u B2 je public, tako da se desio inheritance!!

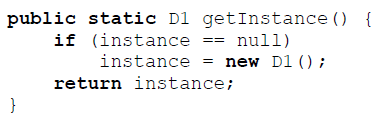
  
Zbog Constructor Call imamo redom B1() te B2() onda  
  
ali PAZI, ZBOG AUTOCLOSABLE!! IMAMO:  
  


Jer  
  


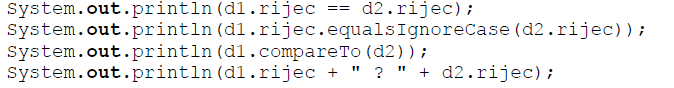
c)

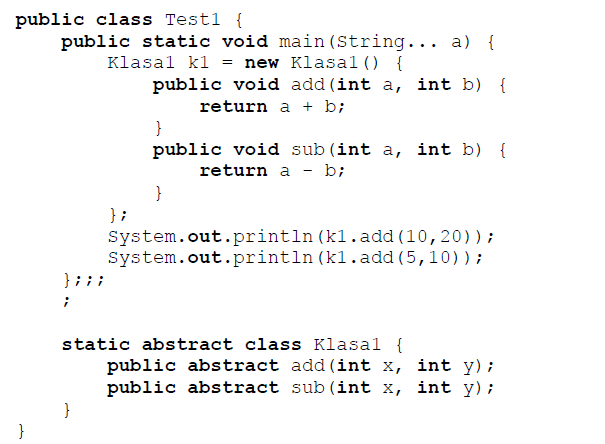
d)

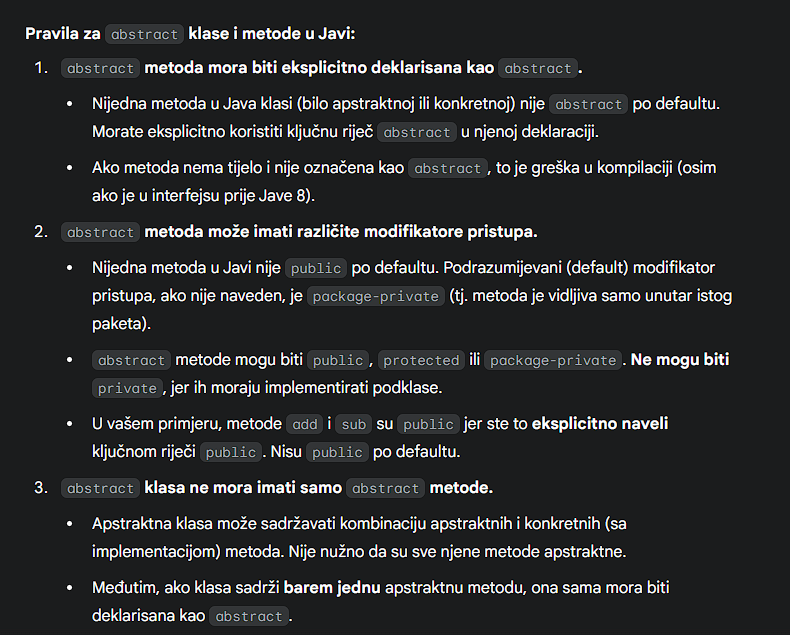
D1 d1 = D1.getInstance();  
  
UVIJEK OBRATI PAŽNJU NA KLJUČNU RIJEČ “static”!!  
Elem, na hep-u imamo objekat klase D1, u klasi D1, statička referenca upućuje na taj objekat..  
  
D1 d2 = D1.getInstance();  
Sad i d1 i d2 upućuju na taj isti objekat..  
  
ZNAČI!! -instace je referenca koja pripada klasi D1 i ona ukazuje na objekat klase D1 koji se nalazi na heap-u!!

d1.rijec = "hello";

d2.rijec = new String("HELLO");  
  
  
true –jer d1.rijec i d2.rijec referenciraju isti objekat!!   
Jer d1 i d2 takođe referenciraju isti objekat!!  
  
true -jer “Hello”.equalsIgnoreCase(“Hello”)  
0 – jer rezultat poređenja je 0, identičan je sadržaj..   
HELLO ? HELLO

1. Zadatak..

a) 



b)

Uvijek obrati pažnju:  
 -potpis main metode  
 -privatni članovi/metode klase  
 -statički članovi/metode klase  
  
Ovde konkretno, uočiš privatni-x, i vidiš da se on poziva unutar statičke metode!!  
  
c) Moraš znati šta se nalazi na heap/stack/data-egment!!