

פתרון לבחינה לדוגמא בתוכנה 1

שאלה 1:

1.

b1.equals(null) /**/ מוחלף ב-

i. יודפס true ii. יודפס false iii. תתקבל שגיאת קומפילציה iv. תתקבל שגיאה בזמן ריצה

הסבר: המימוש של equals שנקרא הוא המימוש הדורס של B. בשורה השלישית מנסים לגשת לשדה של obj, ומכיוון שהוא null, בזמן ריצה ייזרק חריג מסוג NullPointerException.

2.

b1.equals(b2) /**/ מוחלף ב-

i. יודפס true ii. יודפס false iii. תתקבל שגיאת קומפילציה iv. תתקבל שגיאה בזמן ריצה

הסבר: המימוש של equals שנקרא הוא המימוש הדורס של B. מימוש זה מקדם את השדה f של b1 ל-1 בעוד השדה של b2 נותר 0. לכן, ההשוואה של 0 ל-1 מחזירה true.

3.

b3.equals(b3) /**/ מוחלף ב-

i. יודפס true ii. יודפס false iii. תתקבל שגיאת קומפילציה iv. תתקבל שגיאה בזמן ריצה

הסבר: המימוש של equals שנקרא הוא המימוש הדורס של B (לפי הטיפוס הדינאמי). מימוש זה משווה את השדה f של b3 לעצמו פחות 1, ולכן ההשוואה בהכרח מחזירה false.

4.

b2.eq(b2) /**/ מוחלף ב-

i. יודפס true ii. יודפס false iii. תתקבל שגיאת קומפילציה iv. תתקבל שגיאה בזמן ריצה

הסבר: eq קוראת למתודת equals של super, כלומר, של Object. מתודה זו מחזירה true על שתי הפניות לאותו עצם בזיכרון, כמו במקרה הנ"ל.

5.

b1.eq(b3) `/**/ מוחלף ב-`

.i יודפס true .ii יודפס false .iii תתקבל שגיאת קומפילציה .iv תתקבל שגיאה בזמן ריצה

הסבר: eq מצפה לקבל כארגומנט עצם מטיפוס B אך b3 הוא מטיפוס Object, לכן הקומפיילר מדווח על שגיאה.

שאלה 2:

6.

- a. לא יתקמפל בגלל חלוקה באפס
- b. יתקמפל אבל התוכנית תזרוק חריג כאשר היא תנסה לחלק באפס
- c. יודפס X
- d. יודפס Y

הסבר: double מאפשר חלוקה ב-0 של מספר חיובי והתוצאה היא `Double.POSITIVE_INFINITY` ומספר זה גדול מ-1.

7.

- a. לא יתקמפל
- b. יתקמפל אבל יזרוק חריג מסוג `NullPointerException`
- c. ידפיס 0
- d. ידפיס i

הסבר: לא ניתן לגשת למשתנה מקומי שלא אותחל בערך.

8.

- a. B
- b. BC
- c. BCD
- d. לא יתקמפל

הסבר: הריצה מתחילה מה-case המתאים וממשיכה לכל הבאים אחריו אלא אם עוצרים אותה ע"י, למשל, `break`.

9. d - יזרוק חריג אחר

הסבר: מערך הקלט הוא בגודל 2 כך שאין לו תא באינדקס 2 וניסיון לגשת אליו ייכשל בזמן ריצה
ויזרוק חריג `ArrayIndexOutOfBoundsException`.

10. b - יודפס 4

הסבר: אין שגיאת קומפילציה בשורה 3 - גם אם `a` הוא `final`, ניתן לשנות שדות בתוכו, ואסור רק
לשנות את המצביע. לכן גם לא יזרק `FinalModificationException`. אין שגיאה בשורה 7 מכיוון
שאנחנו נמצאים בתוך אותה מחלקה ואין בעית נראות.

11. c - לא יתקמפל

הסבר: `Outer.Inner` היא מחלקה פרטית, לא ניתן לגשת אליה מ-A

12. c - 1 4 3 2

הסבר: הלולאה מחלקת את המספרים רנדומית לשני רצפים. זהו הרצף היחיד שלא ניתן לחלק לשני
רצפים עולים.

13. a - 6

הסבר: `i` נוסף ל-`sum` ואח"כ מקודם פעמיים בכל לולאה, ויוצא ש `sum=0+2+4` ואז יוצאים מהלולאה
עבור `i==6`.

14. a - לא יתקמפל

הסבר: צמצום נראות של `f`

15. a - לא יתקמפל

הסבר: חריג שאינו נתפס או נזרק

16. c - ידפיס `in B in A`

הסבר: יש העמסה של `f`, הקומפילר בוחר עבור כל קריאה את החתימה המתאימה ביותר.

17. b - ידפיס A in A

הסבר: יש העמסה של f, הקומפיילר בוחר עבור כל קריאה את החתימה המתאימה ביותר - החתימה עם ארגומנט A (יותר ספציפית מ-object)

18. a - Clicked! Clicked! Clicked!

19. b - Clicked! Clicked! ?

20. d - Clicked! ? ?

הסבר: כל לחיצה על כפתור משנה אותו פעם אחת ל-clicked

21. e - Clicked! ? ?

22. e - Clicked! ? ?

23. c - ? ? ?

הסבר: לחיצות על 2 ו-3 משנות את כפתור 3 ל-clicked

24. a - Clicked! Clicked! Clicked!

25. c - ? ? ?

26. c - ? ? ?

הסבר: השדה num מאותחל ל-0. אם לוחצים על הכפתור באינדקס num הוא הופך ל-clicked והאינדקס מקודם. אם לוחצים על כפתור clicked שוב הוא הופך חזרה ל-? (סעיף 26)