שאלה 1:

תשובה:

נימוק:

שאלה 2:

תשובה:

נימוק:

שאלה 3:

תשובה:

נימוק:

שאלה 4:

תשובה: ו – רק טענות 3+2 נכונות.

נימוק:
טענה 1 – הטענה לא נכונה. אמנם אין שום משתנה שמצביע אליו, אבל אין לנו דרך לדעת בוודאות מתי האובייקט שנוצר בשורה # ישוחרר ע״י ה GC.
טענה 2 – הטענה נכונה. האובייקט הוא האובייקט ש myInteger (השדה הסטטי) מצביע אליו.
טענה 3 – הטענה נכונה. ה finalize יכול, למשל, להוסיף אותם לרשימה מקושרת סטטית, ואז ישאר מצביע לאובייקטים אלה.

שאלה 5:

תשובה:

נימוק:

שאלה 6:

תשובה: ג. b1.equals(b2)

נימוק: כשמפעילים את הקריאה b1.equals(b2), f של b1 שווה ל 1 (עושים לו ++ בשורה הראשונה), ו f של obj (שמצביע ל b2) שווה ל 0. ה if מתקיים ולכך מופעלת פונק׳ equals שנורשה מ Obejct (בקריאה ל eq) ומתבצעת השוואה לפי כתובות.

שאלה 7:

תשובה: ו. מלבד תשובה זו, כל התשובות לא נכונות.

נימוק: האצלה והכלה (אופציות א׳ ו ב׳) אינן תחליפיות לירושה. למרות שהן מאפשרות חיסכון בקוד, מחלקה B שתממומש ללא ירושה לא תקיים את יחס is-a עם A ולכן לא נוכל לשלוח אותה לפונק׳ שמקבלות את A (לדוגמא). ג לא נכון כי מחלקה פנימית לא תחליפית לירושה אפילו בהיבטים של מניעת שכפול קוד. ד לא נכון מהסיבות ש א ו ב׳ לא נכונות.

שאלה 8:

תשובה: רק f2+f3 מתקמפלות.

נימוק: f1 לא מתקמפלת מהסיבה הבאה. המשתנה l1 יכול להצביע רק לאוסף של Number. הרשימה l2 יכולה להצביע לאוסף של כל דבר שמרחיב את Number, למשל, אוסף של Double-ים. לו ההשמה היתה מתקמפלת, l1 יכול היה להצביע לרשימה של double-ים, מתוך מחשבה שזו רשימה של Number-ים. זה היה, למשל, מאפשר לבצע הכנסה של אובייקט מסוג Integer (שהוא סוג של Number) לרשימת Double-ים.

שאלה 9:

תשובה:

נימוק:

שאלה 10:

תשובה:

נימוק:

שאלה 11:

תשובה: ד׳ – רק טענה 4 לא נכונה.

נימוק: טענה 4 לא נכונה – על מנת להדפיס 5, ה set צריך לחשוב שיש לו 5 איברים שונים. זה לא אפשרי כשהוכנסו רק 4 איברים.
לגבי השאר הטענות – טענה 3 נכונה עבור המימוש של equals שנורש מ Object (השוואה לפי כתובת).
טענה 2 נכונה עבור מימוש equals אשר משווה לפי ערך ה i (סה״כ 3 ערכים שונים), וטענה 1 נכונה עבור מימוש המשווה לפי ערך ה j (ואז יש 2 איברים שונים).

שאלה 12:

תשובה: ב. בריצה של תוכנית יתכנו מופעים של המחלקה A, אך לא יתכנו מופעים של מחלקות שיורשות מ A.

נימוק: מכיוון של הבנאי של A הוא פרטי, שום מחלקה מחוץ ל A לא יכולה לרשת מ A, שכן מחלקה יורשת חייבת לקרוא לבנאי של מחלקת הבסיס. מכיוון שבשאלה נתון שלא נוצרות מחלקות פנימיות ב A, הרי שאין מחלקה שמרחיבה את A. לעומת זאת, יכול להיות קוד בתוך A שמייצר אובייקטים מטיפוס A.

שאלה 13:

תשובה: ד. רק # ו @.

נימוק:
שורה * ו $ - הפונק׳ f1 ו f2 צריכות לקבל רשימה של טיפוס התלוי ב T (T בדיוק או טיפוס שמרחיב את T). ב b ה T הוא מחרוזת, ולכן שתי הקריאות ששולחות רשימת שלמים לא מתקמפלות.
שורה # ו @- הקריאות מתקמפלת. הקריאה ל f1 ו f2 תקינה מבחינת התאמה לסוג הרשימה. הקריאה ל comp תקינה גם היא. comp דורשת לקבל שני איברים מטיפוס זהה שהוא גם Comparable, מה ש String מקיימת.

שאלה 14:

תשובה: ו. מלבד תשובה זו כל התשובות לא נכונות.

נימוק: שורה 1 לא מתקמפלת. יש פה דריסה לא חוקית של f של A כיוון שיש הורדה בניראות, וזה גם מימוש לא חוקי של המנשק בגלל אותה הסיבה בדיוק – הקטנת הניראות. אם נמחק את השורה הראשונה, הקוד עדין לא יתקמפל כי יחסר מימוש בניראות public ל f (המימוש שנורש מ A הוא בניראות package).

שאלה 15:

תשובה:

נימוק:

שאלה 16:

תשובה: ב. רק מימוש 2.

נימוק:
הפונק׳ printDecision צריכה לקבל אובייקט מסוג Decidable, שהוא בעצם מנשק פונק׳ שמגדיר פונק׳ יחידה – משני char-ים ל int. לכן, ניתן לממש אותו באמצעות ביטוי למבדא יחיד שמממש את אותה הפונקציה.
מימוש 1 – שולח ל printDecision פונק׳ משני char-ים ל Decidable, ולכן לא תקין.
מימוש 2 – תקין, כיוון ששולח Decidable (המימוש של Decidable עצמו תקין).
מימוש 3 – המימוש של Decidable אינו תקין בגלל החתימה של decide שלא מתאימה למה שמוגדר במנשק (כלומר, למעשה, המנשק לא מומש כי חסר מימוש לפונק׳ decide שהמנשק מגדיר).

שאלה 17:

תשובה: ב. רק טענה 1 נכונה

נימוק:
טענה 1 – נכונה. אם שורה 2 מתקמפלת, הרי ש C יורש מ A (ב upcasting לא חייבים לעשות casting מפורש). לכן, שורה 3 תתקמפל גם היא.
טענה 2 – לא נכונה. אם A יורשת מ B ו C יורשת מ A, הקוד יתקמפל. יחד עם זאת, b יכול להיות אובייקט מטיפוס D שגם הוא יורש מ B.
טענה 3 – לא נכונה. עבור המבנה הקודם (C יורשת מ A שיורשת מ B), נוכל להגדיר ש b הוא מטיפוס A, ו c הוא מטיפוס C. שורה 3 תתקמפל ותרוץ (upcasting). שורה 2 תתקמפל אך תיפול בזמן ריצה כי אובייקט מטיפוס A ולכן אי אפשר לעשות לו casting ל C.

שאלה 18:

תשובה:

נימוק:

שאלה 19:

תשובה: ד. 14_133442

נימוק:
הזרם מתחיל באיבר 1 שיודפס ב filter ולא ימשיך הלאה.
לאחר מכן, יגיע האיבר 3 שיודפס ב filter, ימשיך ל map, יודפס שם שוב ואחריו יעבור האיבר 6 ויגיע ל sorted. מכיוון שה sorted צריך להמתין לכל האיברים, חוזרים לתחילת הזרם.
האיבר הבא יהיה 4. הוא יודפס, יגיע ל map, יודפס שוב, וממנו ה map יחזיר 8.
האיבר האחרון הוא 2. הוא יודפס ב filter ולא ימשיך האלה. פה יסתיימו האיברים.
ה sorted ימיין את 6 ו 8, וה reduce יחזיר את הסכום שלהם, שזה 14, וזה הערך שיודפס בסוף.

שאלה 20:

תשובה:

נימוק: