תשובות למבחן 25.10.2020

חלק 1

1. cat ,dog, bird יורשים מAnimal Seimicat יורש מ Cat Buldog יורש מDog. כן אם אנו ל א רוצים ליצור אובייקט של Animal אלא רק של סוג החיה- cat ,dog, bird (או Seimicat , Buldog)

2. לא גכעעכיעכעוע

3. כן

4.לא

5.כן

6.לא

7. כל המתודות באינטרפייס הם public לא ניתן לכתוב modifier access באינטרפייס

8.ההבדל הוא ברמת הנגישות השונה:

Private- מותר לגשת מאותה מחלקה באותו פרוייקט

Intranal- אותו פרוייקט

Protected- אותה מחלקה, מחלקה ירושת מאותו פרוייקט ומחלקה יורשת מפרוייקט אחר.

Public- לכולם מותר לגשת.

9. לא חובה לממש פונקציה וירטואלית בבנים היורשים. חובה לממש פונקציה אבסטרקטית בבנים היורשים.

10. אם כתבתי מתודה וירטואלית באבא ומימשתי בבן – זה override

אם כתבתי שתי פונקציות הזהות בשמותיהם אך שונות בפרמטרים האם זה overload

אם כתבתי שתי פונקציות הזהות בשמותיהם ובפרמטרים אך שונות בטיפוס הערך המוחזר האם זה overload

11.ג.

12. ב.

13. במחלקה סטטית הכל סטטית. לא יוצרים ממנה מופע. במחלקה לא סטטית עם פונקציות ומשתנים סטטים לא צריך מופע כדי לגשת למשתנים והפונקציות הסטטיות אך ניתן ליצור מופע.

14. הבנאי הסטטי נקרא לפני שנוצר מופע של המחלקה הוא שפונים לאחד מהמשתנים הסטטים בבנאי הסטטי אפשר להתחל משתנים סטטים או לבצע פעולה שצריך לעשות רק פעם אחת לפני שמתחילים להתשמש במחלקה

15.
```
public int MyProperty { get; set; }
```

16. כדאי להשתמש ב property בגלל שזה חוסך את הצורך להשתמש ב getו set

לא ניתן לעשות public get ניתן לעשות -private set -

17. דוט נט יצור שדה ושני פונקציות get ו set עבור ה property

18. ניתן בעזרת ה ildsm יוצרים קוד IL מבצעים בקוד זה את השינויים הדרושים. ובונים את התוכנית מחדש מהקוד הIL.

19. בעניין. stack, queue, list, dictionary במקרים הבאים, במה הכי כדאי להשתמש - :

אני כותב אפליקציית חייגן ואני מעוניין לשלוף את השיחות האחרונות שחוייגו – stack

אני כותב תוכנית המטפלת בבקשות שהגיעו למערכת ואני מעוניין לשלוף את הבקשות בסדר שהם הגיעו ולטפל בהם אחת -אחת – queue

אני מנהל אוסף של רשומות ואני מעוניין בצורה אקראית לגשת לאיברים ספציפיים ברשימה עפ"י מיקומם – list

אני מנהל מאגר לקוחות ואני מעוניין לשלוף לקוח לפי שם המשפחה שלו)בהנחה שאין כפילויות) בצורה המהירה ביותר - dictionary

20. כי אנחנו רוצים להוסיף אופציה לאופרטור לעבוד על סוג נוסף של משתנה לא לשנות את המשתנה (ולבטל את הקודם)

21. הסבר את הפונקציות הבאות במחלקת : Object מתי נשתמש בהם? מתי נעשה להם ? override ToString –נשתמש בהם כשנרצה להפוף את האובייקט למחרוזת ולהציג פרטים ממנו. נעשה לו override כשמוסיפים למחלקה שדות ורוצים ליצור מחרוזת מהשדות האלה שתוכל להציג אותם בצורה נוחה

GetType - נשתמש בו שנרצה לדעת את סוג האובייקט.לא עושים לו override

Equals – שנרצה להשוות בין שני אוביקטים. עושים לו override כאשר רוצים לשנות את הלוגיקה של השוויון למשל אם יש אובייקט שרוצים להשוות אותו ע"י השדות שלו ו כאשר עושים overloading לאופרטור =

GetHashCode- מחזיר Hash Code עושים לו override בדר"כ יחד עם האופרטור =.

22. כי ל ToString עושים override ולGetType לא

23. ב icompreable

24. IComparer

25. אם הייתה חריגה וב catch כתוב הסוג exception התואם/ מכיל את סוג exception שקרה הקוד בcatch יתבצע finally אמור להתבצע גם אם יש שגיאה שמקריסה את התוכנית.

חלק 2

1. 18618181.8181818

2. בgitub

3. בgitub