* 1. מי יורש ממי?

Animal -> Dog -> Buldog

Animal -> Cat -> SiamiCat

Animal -> Bird

* 1. כי מחלקה אבסטרקטית כל התכונות חייבים להיות ממומשים במחלקות היורשים ממנה. ויש תכונות כלליות של כל החיות.

2. לא, אי אפשר. כי יכול להיות מצב סטירה בן התכונות: אם נירש יותר מאחת, אז פונקציות דומות בתוך המחלקה היורשת לא נדע ממי לירש.

3. כן, זה אפשרי.

4. כן, ניתן להצהיר על מתודה אבסטרקטית במחלקה אבסטרקטית.

5. כן. ניתן לכתוב פונקציה רגילה או Virtual .

6. לא כי אינטרפייס זה רק חוזה - הגדרה מה חייבים לממש במחלקות היורשות ממנו.

7. כן. כל המתודות באינטרפייס הם רק public. לא ניתן להגדיר אחרת.

* 8. **Public** – רמת הנגישות הגבוהה ביותר. מחלקות ומידע שיסווגו כ-Public יהיו נגישים לכולם.
* **Private** – רמת הנגישות הנמוכה ביותר. מידע זה יהיה נגיש רק למחלקה עצמה.
* **Protected** – המידע יהיה נגיש רק למחלקה עצמה וליורשות שלה **וגם** ליורשות שב-Assemblies אחרים.
* **Internal** – המידע יהיה נגיש בכל מקום באותו Assembly

9. פונקציה virtual היא פונקציית שמאפשרת, אך לא מחייבת, שיממשו אותה במחלקת הבן. כל הפונקציות האבסטרקטיות המופיעות בקוד חייבים להיות ממומשים אך במחלקות היורשות.

10.1 זה נקרה overriding. Overriding – או "דריסה" היא האפשרות לדרוס פונקציות שירשנו למחלקה והן הוגדרו כ-Virtual אוAbstract במחלקת האב. בחתימת הפונקציה ה"דורסת" נכתוב את המילה override.

10.2 זה נקרה Overload. ניתן לייצר מספר פונקציות עם אותו השם, שעושות את אותה הפעולה, אך מקבלות ארגומנטים שונים או שממשות את פעולתן בדרכים שונות.

10.3 לא זה ולא זה.

11. const לא מאפשר להכניס ערך לאחר הקומפילציה. readonly כן מאפשר.

12. למשתנה Readonly ניתן לתת ערך רק בתוך הבנאי או בעת ההצרה (Declaration).

13. מחלקה סטטית (Static Class) – היא מחלקה שלא ניתן לייצר ממנה מופע או לרשת ממנה, וכל ה-Members שלה חייבים להיות סטטיים.

מחלקה לא סטטית יכולה ולהכיל גם Static Members וגם Members רגילים. במקרה זה, ישנה הפרדה בין השניים. זה כמו שתי מחלקות – אחת רגילה, ואחת סטטית. לאובייקט הנוצר מן המחלקה לא תהיה גישה לפונקציות והשדות הסטטיים, ולפונקציות הסטטיות לא תהיה גישה למידע ולפונקציות הרגילות.

14. בנאי סטטי מתחיל לרוץ כאשר התכנית מתחילה לרוץ. תפקידו רק לאתחל את שדות המחלקה. בנאי סטטי לא יכול לקבל פרמטרים.

15. public int Student\_ID {get; set;}

16. במקרה שבו מוסרים המידע (שדות) למחלקות אחרות ניתן לייצר הגבלות על כך בעזרת Properties.

גם ניתן לעשות public get ו- private set. זה היתרון של properties. וזה חוסך המון עבודה. וגם מייצר DATA (backing field). וחוסך זמן בזיכרון.

17. המימוש של הProperty הוא באמצעות getter וגם setter. ניתן לקבל ערך בעזרת return, ולשנות בעזרת value =.

18. כן, אפשר לשנות קובץ exe באמצעות assembly מחדש של הקובץ . ניתן לשנות את הקוד באמצעות פירוק בILDASM ואחרי שסיימת להרכיב את הקוד מחדש באמצעות ILASM.

19.

1. במקרה של שיחות אחרונות – תשתמש בQUEUE

2. במקרה של בקשות לפי הסדר – תשתמש בSTACK

3. במקרה של אוסף רשומות – List

4. מאגר לקוחות – Dictionary

20. במקרה של overloading אנחנו לא דורסים שום דבר, אלה מימוש של כמה פרמטרים.

21.

1. ToString – נממש כשנרצה להדפיס במחלקה ש נתונים או שדות שנרצה.

2. GetType – מחזיר שם הnamespace והמחלקה שממנה הגיע המשתנה. שם לTYPE.

3. Equals – נממש כשנרצה להשוות ערכים אחרים במחלקה כשמשווים בין מופעים של מחלקות.

4. GetHashCode – נממש כשנרצה לשלוט במספרים שחוזרים.

22. כדי למנוע שינוים בGETTYPE.

23. IComparable

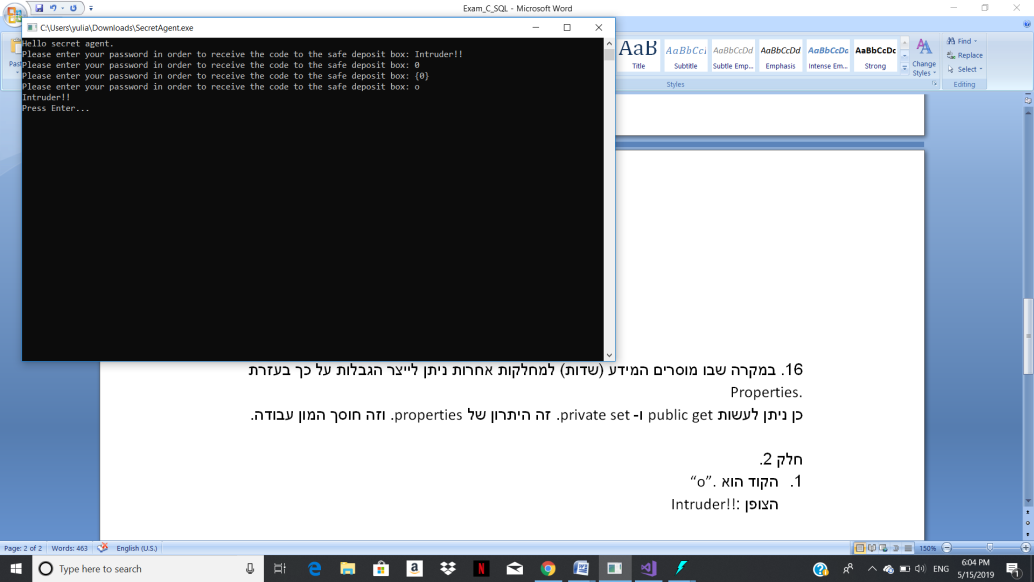
24. IComparer

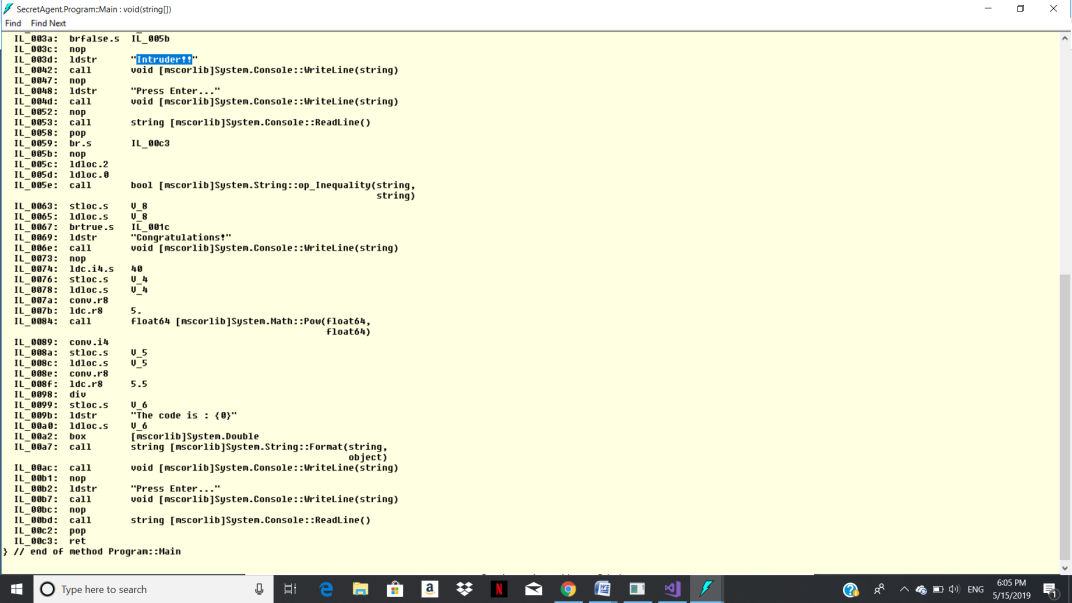
25. כן. ה-Finally יופעל ברגע שהאקספשן החדש יזרק.

חלק 2.

1. הקוד הוא “o”.

הצופן : 10618181.8181818





1. בGIT
2. בGIT

**חלק 3**

1. Stored Procedure היא שיטת כתיבה של שאילתות SQL המאוחסנות בשרת של בסיס הנתונים. כמו בפרוצדורה רגילה, הפרוצדורה המאוחסנת כוללת קוד של התחלה, קוד של סיום, פרמטרים וגוף הפרוצדורה, שבו נכתבות הפעולות המתבצעות בפרוצדורה.
2. במידה ואם יש לנו מספר מחלקות ADO, יותר נוח לשנות נתיב לBD במקום אחד כמו למשל App.config. כמו כן אין צורך לקמפל שוב.
3. ג SQLITE.
4. ב Firebase.
5. א MSSQL.
6. ב Firebase.
7. 1:1 (א בן אדם אחד - אישה אחת. n:1 (ב לקוח אחד - מספר הזמנות. n:m (ג לכל הזמנה יש מספר מוצרים וגם כל מוצר קיים במספר הזמנות.
8. א) inner join
9. ב) left join
10. ד) full outer join
11. ב) right join
12. ה) cross join
13. POCO הינה מחלקה רק עם מידע, ללא התנהגות ופונקציונאליות. כדאי להשתמש בה כשרוצים לשמור מידע או לסדר שמירת מידע.

כדי להשוות בין אובייקטים, חייבים לממש לממש ,Equals וגם GetHashCode. אחרת המחשב ישווה את הכתובות בזיכרון של האובייקטים, ולא את המידע שבהם.

1. מחלקת DAO היא מחלקה איפה רושמים פרמטרים לחיבור בין בסיס נתונים וגם שפת LINQ (למשל) כדי לרשום שאילתות. זה לא מחלקה לכתיבת לוגיקה.

**חלק 5 SQL.**

**ב GIT**