1 - מחלקה היא התיאור של מבנה הנתונים ואובייקט הוא המימוש שלו, כלומר מחלקה היא המתכון בזמן שהאובייקט הוא מה שמכינים בעזרת המתכון.
2 - אפשר להגיד שירושה היא הכלה כיוון שהמחלקה היורשת מכילה את כל המאפיינים והמתודות של מחלקת האם.
3 - רכב חשמלי יורש מהמחלקה רכב. אחד היתרונות שקיבלנו זה שכאשר אנחנו יוצרים את המחלקה רכב חשמלי, אנחנו מקבלים את כל מה שכבר יש במחלק רכב, אנחנו נצתרך לשנות חלק מהדברים אבל רוב המאפיינים הבסיסיים כמו כמות דלתות, חלונות, גלגלים וכו אנחנו נקבל חינם רק בגלל הירושה.
4 - המשפט הזה לא נכון כיוון ששולחן וכיסא לא ממשים את היחס is a. שולחן הוא לא כיסא ולהפך. שניהם רהיטים כלומר שניהם ממשים את היחס is a עם המחלקה רהיטים, הגיוני ששניהם ירשו מאותה מחלקה אבל שני מחלקות שיורשות מאותה המחלקה לא ממשות את יחס הירושה אחת כלפי השניה.
5 - כדי שנוכל למצוא בקלות שדות מידע ולהפריד אותם משדות מקומיים. שימוש ב _ גם חוסך מאיתנו להשתמש ב this כשאנחנו מאתחלים משתנים (בבנאי למשל).
6 - בנאי הוא מתודה בעלת אותו השם כמו המחלקה שמשתמשים בה כדי ליצור מופע חדש של המחלקה. נדרוס את הבנאי הדיפולטי אם אנחנו רוצים שהמשתמש יממש את שדות המידע שאנחנו צריכים בעת יצירת האובייקט.
7 - זהו בנאי שאנחנו מקבלים חינם כשאנחנו יוצרים מחלקה. הוא לא מאתחל אף שדא.
8 - בנאי יוצר מופע חדש, ויכול לאתחל אותו במידת הצורך. Object initializer זאת פשוט דרך נוחה יותר לאתחל משתנים של אובייקט קיים - הוא לא יכול ליצור מופע חדש.
9 - העמסה זה כאשר אנחנו יוצרים עוד פונקציה עם אותה החתימה כמו פונקציה קיימת (היא יכולה להיות private אם הפונקציה המקורית היא public) והיא צריכה לקבל משתנים שונים מפונקציית המקור. אנחנו נעמיס בנאי כשאנחנו רוצים להביא למשתמש את האפשרות לממש מחלקה על ידי שימוש במשתנים שונים. למשל שתהיה למשתמש את האפשרות ליצור אובייקט בלי לאתחל משתנים וגם את האפשרות לאתחל את כל המשתנים בעת מימוש המחלקה.
10 - פונקציית ToString היא פונקציה שמדפיסה את הערכים של כל המאפיינים של המחלקה כשאנחנו קוראים לה או כאשר אנחנו מדפיסים את האובייקט שמממש את המחלקה שדרסה אותה. נשתמש בא כאשר אנחנו רוצים לראות את ערך השדות של האובייקט, בזמן דיבאגינג למשל.
11 - אנחנו נשתמש במילה השמורה override כאשר אנחנו נותנים מימוש חדש של פונקציה שירשנו. אנחנו חייבים להשתמש במילה השמורה הזאת כדי שתהליך הדריסה ימומש.
12 - במידה ולא נדרוס את ToString אנחנו נקבל את המימוש של Object - הדפסה של ה namespace ושם המחלקה.
13 - מחלקה אבסטרקטית במקרה שלנו היא מחלקה שיש לה מימוש מינימלי, מתודה אבסטרקטית היא מתודה ללא מימוש - רק החתימה. אנחנו נשתמש במחלקה אבסטרקטית כשנרצה לחפות את המימוש על המשתמש וכאשר המחלקה עצמה היא כללית מידי ודורשת ירושה של מחלקות נוספות - המחלקה בעל חיים היא מאוד כללית (לא הגיוני שיהיה מופע של בעל חיים כללי) לכן הגיוני שתהיה אבסטרקטית. למרות זאת, הגיוני שיהיו לה מאפיינים שמשותפים לכל בעליי החיים כמו bool mamal, int legs וכו..
14 - פונקציה אבסטרקטית היא חתימה בלבד ()public abstract int doSomething כאשר פונקציה רגילה, היא בעלת מימוש - במקרה הזה אנחנו מחוייבים למימוש המינימלי של להחזיר int לפחות. פונקציה אבסטרקטית לא יכולה להיות private - זה לא יהיה הגיוני כיוון שהיא קיימת כדי לכפות מימוש.
15 - המחלקה עצמה תהיה חייבת להיות אבסטרקטית.
16 - כן.
17 - לא, היא תהיה חייבת להפוך לאבסטרקטית.
18 - אי אפשר ליצור מופע של מחלקה אבסטרקטית.
19 - אי אפשר באמת ליצור מופע של מחלקה אבסטרקטית, אבל כאשר אנחנו ממשים מחלקה שיורשת ממחלקה אבסטרקטית אנחנו נקרא לבנאי של המחלקה האבסטרקטית. בעת מימוש כל מחלקה שיורשת ממחלקה אחרת,

קודם כל יווצר מופע של מחלקת האם (הקומפיילר קורא לבנאי של מחלקת האם) ורק אז המחלקה הרלוונטית. במקרה ומחלקת האם היא מחלקה אבסטרקטית, קודם יווצר מופע של מחלקה אבסטרקטית - אבל לתווך זמן כל כך קצר שזה לא באמת נחשב ואנחנו לא באמת יכולים להשתמש בזה.

20 - זאת מילה שמורה שמאפשרת לנו לגשת למימוש של מחלקת האם, זה שימושי במיוחד כשיש מתודות במחלקה היורשת שדרסו מתודות במחלקת האם וזה נוח במיוחד כשמדובר על בנאים. אנחנו חייבים להשתמש ב base כאשר אנחנו רוצים לקרוא לבנאי ספציפי במחלקת האם מהמחלקה היורשת.
21 - נשתמש ב base.ToString כשאנחנו רוצים להרחיב את הToString של מחלקת האם במחלקה היורשת. זה יחסוך מאיתנו כפילות קוד - נשתמש במה שכבר נכתב במחלקת האם ונוסיף לו את מה שחסר במחלקה היורשת.
22 - + מסמל public
23 - חץ מהמחלקה היורשת למחלקת האם.
24 - מחלקה טלפון מחילה את המחלקה איש קשר
25 - זאת שפת ביניים בין שפות .net לשפת המכונה
26 - נשתמש בILDASM
27 - נוכל לפתוח את הקוד באמצעות ILDASM לעשות Dump לתוך קובץ ביניים, לפתוח אותו, לשנות את הקוד כרצונינו ולקפמל חזרה באמצעות ILASM.
28 - פולימורפיזם זה כאשר באמצעות ירושה אנחנו מממשים מתודה אחת במספר צורות. למשל במחלקת בסיס בעל חיים יש מתודה להשמיע קול, המחלקה כלב תממש את המתודה בלהדפיס "האו האו" והמחלקה חתול תממש את אותה המתודה בלהדפיס "מיאו". אנחנו יכולים לעשות מערך של חיות ולהדפיס את כולם בלולאה - המימוש של המתודה להשמיע קול יהיה שונה פר מחלקה והמימוש הזה יקבע רק בזמן ריצה.