תרגול 01072020 תשובות לשאלות

1. מחלקה : היא בעצם התבנית המכילה את הפונקציות ושדות הנתונים וסוגם מופע: הכנסת התבנית למשתנה ובכך ניתן להכניס נתונים לשדות הנתונים והפעלת הפונקציות בהתייחסות לנתונים הנמצאים באותו המופע.
2. כיוון כשמחלקת הבן יורשת מהאבא , מחלקת הבן מכילה את כל שדות המידע והפונקציות שיש לאבא וניתן להגיד שהבן מכיל את האבא.
3. מחלקה שנקראת צורות , ומחלקות יורשות שנקראות מרובע ומשולש , מחלקת האב כולל פונקציה אבסטרקטית לחישוב שטח ועוד מספר שדות נתונים ברגע שאני יורש , כל שדות הנתונים והפונקציות הקיימות במוריש עוברות גם ליורש מה שנותר לי זה להוסיף את השדות והפונקציות המבדילות בניהם.

4. משפט זה אינו נכון כיוון ששימושם אינו לאותה המטרה משכך המחלקה היורשת לא מסוגלת לממש את כל שדות המידע והפונקציות מהמחלקה המורישה.

5. כדי להבדיל בין שדות מידע של המחלקה ולא פונקציות.
6. בנאי הינה פונקציה המאתחלת (מגדירה) פרמטרים התחלתיים אותם ירשנו או הגדרנו במחלקה והגדרנו אותם כפרמטרים הכרחים לצורך יצירת המופע.
7. בנאי ברירת מחדל הינו בנאי שלא מאתחל אף פרמטר ובעצם מוגדר גם מבלי שכתבנו אותו בקוד .
8. הבדל בין בנאי לבין "מאתחל אובייקט" , בבנאי אנו משתמשים ברגע "לפני" יצירת המופע ובעצם מאתחלים איתו שדות והמופע נוצר עם השדות המאותחלים , ב"מאתחל אובייקט" אנו נשתמש כדי לאתחל שדות לאובייקט קיים בצורה נוחה יותר מבחינת סדר הריצה תמיד אבל תמיד הבנאי ירוץ לפני מאתחל האובייקט .
9. Overloading - אנו יכולים ליצור פונקציה עם אותו השם בעלת חתימת משתנים שונה וכל אחת מהן ניתן לממש באופן שונה על מנת להגיע לתוצאה הרצויה בכך נוכל ליצור מספר אפשרויות לפתרון אותה הבעיה.
Constructor overloading - יצירת חתימה שונה לבנאי בוא אנו יכולים לבקש לאתחל שדות שונים לפני יצירת המופע – הוא חייב להיות בעל חתימה שונה.
10. פונקצית ToString הופכת למחרוזת כל סוג מסויים של מידע , מהדוגמאות שניתנו לנו כאשר מדובר באובייקט / מופע אנו נשכתב מחדש את הפונקציה כדי להציג את כל שדות המידע של האובייקט.
11. Override בשונה מ Overloading , אנו נשתמש בא כאשר יש לנו פונקציה וירטואלית או אבסטרקתית בכדי לשכתב את הפונקציה הקיימת בכדי שבתוך האובייקט החתימה הזו תתפקד כפי שאנו רוצים , לדוגמא toString בכל מחלקה אנו נשכתב אותה כך שתציג לנו את שדות המידע הרלוונטים מאותה מחלקה ושדות אלו בהכרח ישתנו בין המחלקות השונות בעוד ש Overloading מביאה לנו האפשרות ליצור מספר פונקציות בעלות שם זהה עם חתימת משתנים שונה ואופן מימושם גם יכול להיות שונה בכדי להשיג את אותה המטרה.
12. נקבל את שם המחלקה , וה space name שלה .
13. המשמעות של המילה abstract היא שנהיה חייבים לממש את הקוד של הפונקציה במחלקה היורשת , נרצה לממש מחלקה אבסטרקטית כמחלקת בסיס כאשר יש לנו תכונות משותפות רבות בין דברים שונים , לדוגמא מחלקת צורות , המחילה פונקציה אבסטרקטית לחישוב שטח , משולש ועיגול ומרובע יכולות לרשת ממחלקת צורות , אך הנוסחה לחישוב השטח שונה בין הצורות ונממש בבנים את הנוסחה הרלוונטית לצורה .
14. פונקציה אבסטרקטית הינה פונקציה שלא ממומשת במחלקה בא היא נוצרה אלה רק במחלקה שיורשת ממנה , לכן נשים רק את החתימה של הפונקציה ,

פונקציה רגילה אנו מחוייבים להשים גוף לפונקציה במחלקה בא הצהרנו על הפונקציה (גם אם לא כתובה בא שורת קוד אחת)

15. עם הצהרת פונקצה אבסטרקטית במחלקה באותו הרגע אנו צריכים להצהיר על המחלקה כאבסטרקטית .
16. כן
17. לא
18. לא
19. .
20. משמעות המילה base היא לבצע שימוש בפונקציה המוגדרת במחלקה שממנה ירשנו ולא בפונקציה הכתובה אצלנו. נהיה חייבים להשתמש בא כאשר אנו יורשים ממחלקת אב בא יש בנאי ובמחלקת הבן יש לנו בנאי נוסף שם נצטרך להתשתמש ב base בכדי שנקבל את אתחול השדה מהבנאי של מחלקת האב .
21. כשאנו רוצים להכיל את הפונקציה של האבא לפני הדריסה שלה לצורך התאמה של הפונקציה למחלקה המתאימה , חוסך לנו לכתוב שוב את המימוש שכתבנו במחלקת האב ולנו נשאר רק להוסיף את השדות שנרצה להדפיס בנוסף למה שכבר הגדרנו בפונקצית האב ולא נצטרך לכתוב את הכל מחדש.
22. + מציין את המילה public
23. באמצעות חץ כאשר ראש החץ פונק למחלקה שממנה אני יורש , והזנב של החץ פונה למחלקה היורשת
24. ניתנו מספר דוגמאות לכך לדוגמא בתרגיל שלנו סלט פירות מכיל מערך של מחלקת פירות דוגמא נוספת , רכב מכיל מחלקה של מנוע שבו כלל הפרטים הדרושים על המנוע וכו'
25. IL זו שפת בניים אליה ממורות כל שפות ה DOT NET למינהן בוא אנו נוכל לראות כיצד בחר הקומפיילר לממש את מה שכתבנו ויכולה לגרום לנו להבנה טובה יותר או להסיר סימני שאלה על המימוש שלנו כמו איזה פקודה נעשת לפני ומהו סדר הפעולות שלנו , כמו כן משפה זו המחשב ממיר לשפת מכונה.
26. ILDASM
27. באמצעות ILDASM נוכל לפתוח את הקובץ ולמחוק / לשנות את הדבר הרלוונטי לנו
28. פולימורפיזם – בתרגום חופשי רב צורות או רב צורתיות , בתוכנה אנו משתמשים בכך לשם יצירת מימוש שונה לפונקציה בין הפונקציות היורשות , זה יתבצע על פונקציות אבסטרקטית או על פונקציה ווירטואלית בלבד כך שהפונקציה שתרוץ תתאים למחלקה המתאימה ראה דוגמא בשאלה 13 , בא אנו ממשים את הפונקציה לחישוב שטח לכל צורה (משולש ריבוע ועיגול ) עם נוסחה שונה אך קוראים לאותה הפונקציה בדיוק.