

OOP

Object Oriented Programming – OOP – תכנות מונחה עצמים

זו שיטה תכנותית המאפשרת לבנות תבניות עבור אובייקטים, ולאחר מכן לייצר אובייקטים מתבניות אלו.

תבנית מתארת ישות כלשהי. בקוד אנו בונים תבנית ע"י class (מחלקה).

ליישות יש שני דברים מרכזיים:

א. מאפיינים – אלו המשתנים שמחזיקים את המידע

ב. פעילויות – אלו פונקציות שהישות יודעת לבצע

class - מחלקה

מילה שמורה שיוצרת תבנית.

Object – אובייקט (מילה נרדפת: instance – מופע בזיכרון)

אזור בזיכרון המכיל שדות (תאים) ויש לו שם. ע"י השם ניתן לפנות לתאים שלו.

מ-class יוצרים אובייקטים. האובייקטים נראים באותה הצורה שמוגדרת ב-class.

Access Modifiers – הרשאות גישה

הרשאת גישה זו מילה המגדירה מהיכן ניתן לגשת למשתנה או לפונקציה. במילים אחרות – מה ה-Scope שלהם.

א. public – נגיש מכל מקום.

ב. private – נגיש רק מתוך ה-class.

ג. protected – נגיש רק מתוך ה-class ומהירושים.

Constructor

פונקציה מיוחדת – מופעלת אוטומטית ברגע יצירת אובייקט מהמחלקה.

הייעוד שלה הינו לבצע פעולות אתחול על האובייקט (שכרגע נוצר).

Inheritance – הורשה

העברת כלל התוכן של מחלקה A למחלקה B ללא שכפול קוד. נקרא שמחלקה B יורשת את A.

המחלקה הראשונה נקראת מחלקת הבסיס – Base Class או Super Class.

המחלקה היורשת נקראת מחלקה נגזרת – Derived Class או Sub Class.

המטרה: יצירת מחלקות הדומות למחלקות אחרות ללא שכפולים של קודים.

Override – דריסה

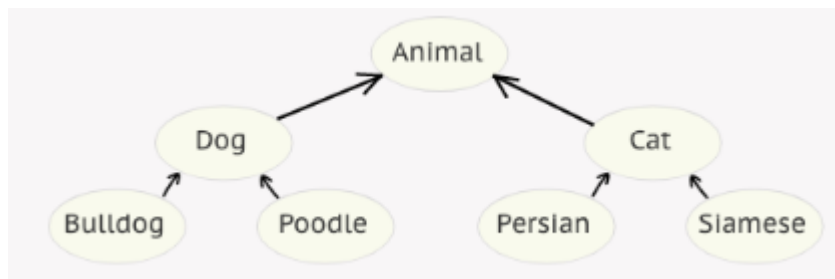
זה המצב בו המחלקה שאנו כותבים קיבלה פונקציה ממחלקת בסיס כלשהי אך הפונקציה לא מתאימה לנו לגמרי, ולכן אנו כותבים פונקציה בעלת אותו השם אך מתאימה יותר. תופס גם ל-Constructor.

Inheritance Tree – עץ הורשה

סכמה של המחלקות וקשרי ההורשה ביניהם.

מחלקות בסיס נמצאות למעלה בעץ, מחלקות יורשות נמוך יותר.

דוגמה:



חוק ההורשה:

זהו חוק המראה היכן מותר והיכן אסור לבצע הורשה.

החוק: ניתן לבצע הורשה אך ורק אם אפשר לומר (במילים) שהמחלקה היורשת היא גם מחלקת הבסיס.

מחלקה אבסטרקטית

מחלקה המיועדת להורשה בלבד ולא ליצירת אובייקטים ממנה. ממחלקה אבסטרקטית לא ניתן להגדיר אובייקט.

פונקציה אבסטרקטית

חוק: אם פונקציה דרושה בכל המחלקות היורשות, הגיוני לשים אותה בבסיס. זאת מפני שבעבודה עם יכולת ה-Polymorphism – אם יש לנו Reference מסוג בסיס – נוכל דרכו לקרוא לפונקציה מבלי הצורך לבצע שום בדיקות instanceof ושום Down Cast.

אם מחלקת הבסיס הינה אבסטרקטית, יכול להיות שאין לנו יכולת לכתוב תוכל לפונקציה (כי אנו עדיין לא יודעים בדיוק מה הסוג המלא/השלם/הסופי של המחלקה). לכן במקרה כזה, נגדיר את הפונקציה כאבסטרקטית – אין לה תוכן.

- פונקציה אבסטרקטית יכולה להיות אך ורק במחלקה אבסטרקטית.
- מחלקה רגילה (שאינה אבסטרקטית) היורשת מחלקה אבסטרקטית חייבת לדרוס את כל הפונקציות האבסטרקטיות שקיבלה.

Reference (ייוחס)

זה תא זיכרון לרוב בגודל של number המיועד להכיל בתוכו כתובת של אובייקט.

לכל אובייקט יש כתובת. Reference מחזיק כתובת של אובייקט.

האופרטור new מבצע:

- יוצר אובייקט בזיכרון
- מחזיר את הכתובת שלו

Reference Type

סוג נתונים שעבודה איתו מצריכה גם אובייקט וגם Reference.

סוגים כאלו הם:

- מחרוזות
- אובייקטים
- פונקציות
- מערכים

Value Type

סוג נתונים שלא דורש Reference אלא מחזיק בתוך עצמו את ה-Value

סוגים כאלו הם:

- number
- boolean
- undefined

Polymorphism – רב צורתיות

זוהי יכולת של Reference מסוג בסיס להחזיק כתובת של אובייקט מסוג נגזרת.