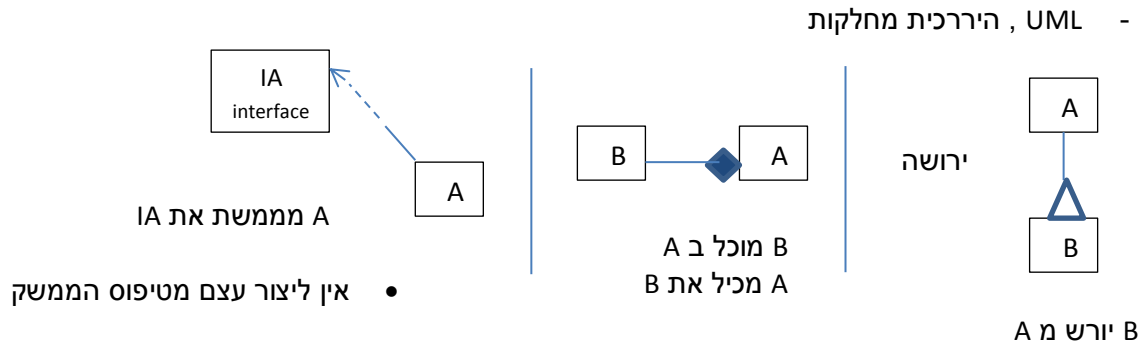


דגשים תמ"ע



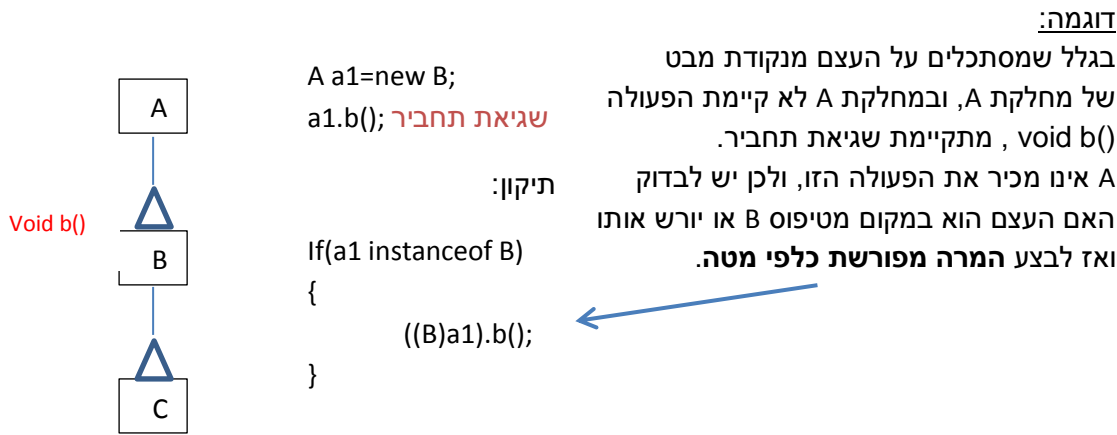
- מחלקה מופשטת

- המחלקות היורשות חייבות לממש את כל הפעולות המופשטות במחלקה המופשטת, ואם הן לא, אז המחלקות הופכות גם הן למופשטות.
- אין ליצור עצם ממחלקה מופשטת.
- במחלקה המופשטת יש פעולה בונה ותפקידה היא שבמחלקות היורשות יהיה ניתן להשתמש בsuper.

- בג'אווה, מחלקה יכולה לרשת רק מחלקה אחת, אך מחלקה יכולה לממש יותר מממשק אחד.

- פולימורפיזם והמרות

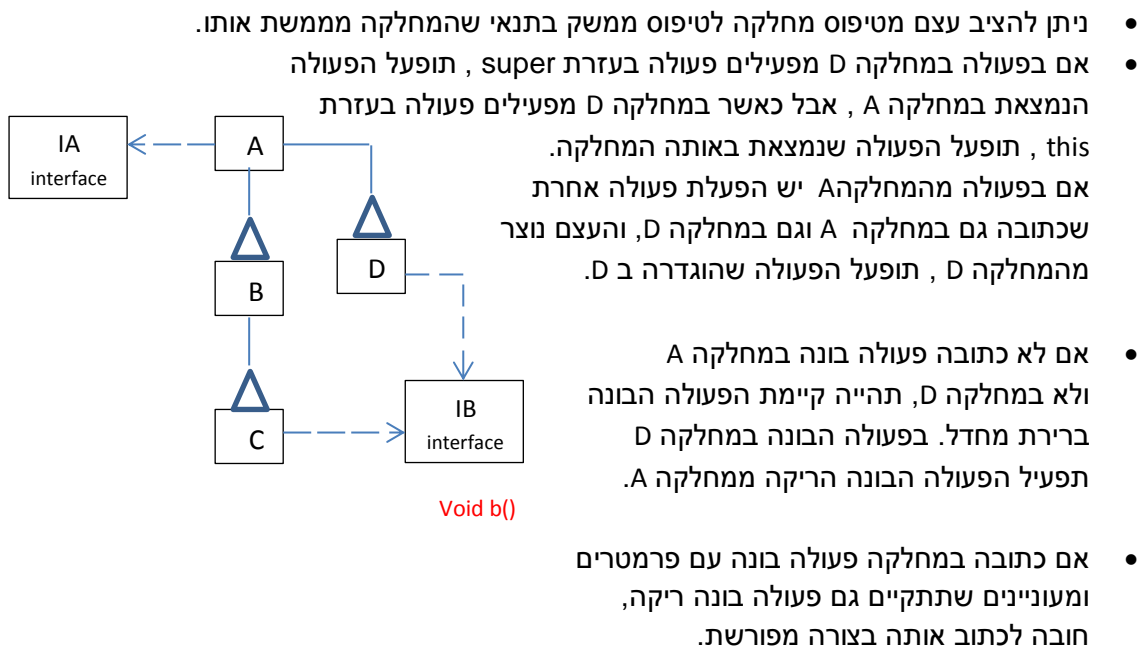
- כל המחלקות יורשות ממחלקת Object
- הפולימורפיזם קשור לעובדה שאפשר במחלקת הבת לדרוס פעולה ממחלקת העל, ועל ידי כך, כשנסתכל על עצמים שונים ממחלקת העל, כל עצם יפעיל את הפעולה הייחודית לו.
- במידה ודרסנו פעולה, העצם יפעיל את הפעולה שבמחלקה שלו. במידה ולא דרסנו את הפעולה, הוא יפעיל את הפעולה שקיבל בירושה ממחלקת העל. לא משנה מהי נקודת המבט או אם בוצעה המרה, משנה הטיפוס ממנו נוצר העצם.
- מתי תתעורר שגיאת תחביר בהקשר של הפעלת פעולות במחלקות יורשות? כאשר במחלקת הבת קיימת פעולה שלא קיימת במחלקת העל, כלומר ביצענו המרה לא מפורשת כלפי מעלה. במידה וננסה להפעיל את הפעולה אשר נמצאת רק במחלקת הבת, תתבצע שגיאת תחביר.
- אומנם למחלקת הבת קיימת פעולה ואומנם העצם נוצר מטיפוס מחלקת הבת, אבל ברגע שמסתכלים על העצם מנקודת מבט של מחלקת העל לא ניתן להפעיל עליו פעולות שלא ומגדרות בו.



- זוהי שגיאת תחביר, מכיוון ש B אינה מכירה את המחלקה C ולכן לא ניתן להציב בה עצם מטיפוס B.
- שגיאת ריצה היא כאשר אנו ממירים עצם לטיפוס אשר לא נוצר ממנו או לא יורש ממנו. לדוגמא ← זוהי אינה שגיאת תחביר מכיוון שזהו לג'אווה, אך ההמרה אינה נכונה.

נימוק נוסף: לא ניתן לעשות המרה כלפי מטה C מכיוון שהעצם עצמו לא נוצר מ C. נימוק נוסף: יש שגיאת ריצה, המרות מתבצעות בזמן ריצה, בזמן הריצה התוכנה תנסה להמיר את העצם לטיפוס C ותגלה שלא ניתן מכיוון שהעצם לא נוצר מהמחלקה הזו.

- המרות וממשקים



- ניתן לבדוק האם עצם הוא מטיפוס ממשק ולא רק אם הוא מטיפוס מחלקה.
- ```
for(.....)
{
 if(arr[i] instanceof IB)
```
- כאשר יש תרשים היררכית מחלקות וממשקים, ומבקשים לכתוב את כל הפעולות שמחייבות במחלקות על פי קוד הממשק והיררכית המחלקות, יש לרשום אך ורק את הפעולות הנדרשות, כלומר אם מחלקה A מממשת ממשק IA ומחלקה B יורשת מ A, במחלקה A נכתוב את כל הפעולות שכתובות בממשק IA ולא נסיף אותן למחלקה B

### הרשאות גישה

מאפשרות לממש את עקרון הסתרה המידע, encapsulation, הכנסה. במחלקה שיורשת נתן לתכונה מידע מוגן protected ומחלקות שאין מהן ירושה private הגישה לפעולות תהייה ציבורית public

### תכונות מחלקה ופעולות מחלקה

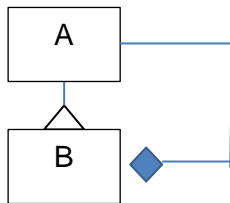
לתכונות מחלקה ולפעולות מחלקה נוסף את המילה static. זה אומר, שיש הקצאה פעם אחת בזיכרון לתכונה, והתכונה משותפת לכל העצמים מהטיפוס. יש לצייר את התכונה מחוץ למלבן של העצם בתרשים העצמים ולעדכן אותו בהתאם לקוד שבמחלקות.

גישה לתכונה של עצם counter שהוגדר במחלקה A, תהייה: A.counter. לתכונה של מחלקה, הפעולה המתאימה תהייה גם היא פעולת מחלקה, כלומר בכותרת הפעולה תופיע המילה static והפעלתה תהייה בעזרת ()שם פעולה.שם מחלקה

### ירושה והכלה

במידה וקיימת מחלקה יורשת וגם מכילה עצם ממחלקה העל.

יש לשים לב, אם החזרת התכונה מטיפוס A במחלקה B היא על ידי השמה או על ידי הפעלת פעולה בונה מעתיקה.



קוד א

```
public class B extend A
{
 A a;

 public B(A a)
 {
 this.a = a;
 }
 public A getA()
 {
 return this.a
 }
}
```

קוד ב

```
public class B extend A
{
 A a;

 public B(A a)
 {
 this.a = new A(a);
 }
 public A getA()
 {
 return new A(this.a);
 }
}
```

בקוד א

תהינה הפניות לעצם שהועבר כפרמטר, אין היווצרות של עצם חדש.

בקוד ב

תהינה הפניות לעצם חדש שהוא העתק של העצם שהועבר כפרמטר.