

## ג'אווה- שיעור חמישי

בפרק הקודם: חזרה ודגשים משיעור שעבר

final- שתי נקודות חשובות:

1. משפר ביצועים = נטען בתחילת ההרצה!
2. מתן הגנה = לא ניתן לשינוי לאחר הצהרה.

### **Inheritance- הורשה**

הורשה היא חלק מרכזי משפות תכנות מונחות עצמים. תחביר השיתוף ושימוש במחלקות ע"י שימוש בקוד שלהם הוא פשוט, אולם בהורשה יש לציין את שם המחלקה החדשה, וכן איזו מחלקה המחלקה החדשה יורשת.

תהליך ההגדרה מתחיל בציון שם המחלקה החדשה, המילה extends (הרחבה) ולאחר מכן שם מחלקת העל- המחלקה ממנה יורשים.

דוגמא:

```
public class Engineer extends Employee{
```

המחלקה **Engineer** יורשת את המחלקה **Employee**

מנגנון זה מאפשר יצירת היררכיה של עצמים. כאשר יש לנו עצם ניתן להגדיר עצם חדש אשר "יורש" את תכונות העצם וגם את כל השיטות של העצם ובנוסף יש לו משלו תכונות נוספות ושיטות משלו.

**subclass** - (תת- מחלקה) מחלקה היורשת ממחלקה אחרת, יורשת את כל החברים שיש למחלקה המורשתה.

**super class** - (מחלקת על) המחלקה המורשתה.

**Descendant** - מחלקה בעץ הירושה היורשת מחלקת על, נקראת צאצא של כל **אחת** מהמחלקות שמעליה.

- בג'אווה לכל מחלקה יכולה להיות רק מחלקת-על אחת. אין מגבלה על "עומק" עץ ההיררכיה כלומר ניתן ליצור עץ ירושה עם כמה רמות.
- תת מחלקה יורשת את כל האיברים של מחלקת העל, כולל השיטות של מחלקת העל.
- השיטה הבונה נושאת את אותו השם של המחלקה. למחלקה Creature יש שיטה בונה Creature שיוצרת אובייקטים לפי מפרט התכונות של המחלקה, ואינה יכולה לעבור בירושה לתת המחלקות שלה.
- מילת המפתח **super** מאפשרת בנייה של שיטה בונה עבור תתי מחלקות. מאחר ובג'אווה אין לאף מחלקה, יותר מאשר, מחלקת-על אחת. ההתייחסות למחלקת על של תת מחלקה היא חד חד ערכית, ושפת ג'אווה מנצלת פשוטות זו באמצעות מילת המפתח **super**

**Protected** - מתן הרשאה לתת מחלקות למחלקת האב.

הרשאה זו היא בין ההרשאה **private** להרשאה **public**. הרשאת גישה זו מאפשרת גישה משני מקומות:

1. מתוך המחלקה בה הוגדר האיבר של המחלקה
2. מכל תת-מחלקה (מחלקה יורשת) של המחלקה בה האיבר הוגדר בה.

**This** - אני נגיש בתוך מה שיש בתוכי (פונקציות וכדומה)

**Super** - מייצג את הקונסטרוקטור של האב (מי שמעליי)

## **פולימורפיזם - polymorphism**

פולימורפיזם מאפשר לנו להתייחס לעצמים מטיפוסים שונים באופן אחיד.

פולימורפיזם מאפשר לנו להתייחס אל אובייקט בצורות שונות. האובייקט עצמו אינו משתנה אלא רק נקודת המבט שלנו. באופן הזה אנו יכולים לזמן שיטה המוגדרת מחדש (overriding) במחלקות הגזרות (בתת-המחלקות) ואע"פ שאנו מזמנים את השיטות באופן אחיד כל אובייקט יבצע את השיטה כפי שהיא מוגדרת בתת המחלקה בה הוא נוצר.

### עקרונות הפולימורפיזם:

**עיקרון 1:** reference בתורשה – reference למחלקת בסיס יכול בפועל להצביע על עצם ממחלקה נגזרת.  
**עיקרון 2:** נוכל ליישם שיטות וירטואליות, כלומר כאשר נקרא לשיטה שהיא קיימת ונקראת בכל אחת מהמחלקות הנגזרות, השיטה הנקראת היא שיטה וירטואלית, אולם בעת הקריאה לשיטה, היישום המעשי יהיה קריאה לשיטה בהתאם לסוג המחלקה היורשת.

## **גישה לשיטה מוסתרת באמצעות super**

שימוש באותה שיטה (שם זהה) במחלקת האב ובתת המחלקות.  
ההגדרה בתת המחלקות אינה מבטלת את השיטה שבמחלקת האב והיא נגישה באמצעות שימוש באובייקטים של מחלקת האב, אפילו דרך תת המחלקות באמצעות super.  
מנגנון זה מאפשר לאובייקטים של תת מחלקה לפעול באופן שונה ממחלקת האב.  
אם ברצוננו לממש את ההגדרה מחדש עלינו לוודא שחתימות השיטות זהות, והמימוש יהיה שונה.  
**Overriding (=דריסה)** - הגדרה מחדש של שיטה אינה מבטלת את הגדרת השיטה במחלקת – העל לאחר ההגדרה מחדש עדיין ניתן להשתמש בשיטה המקורית: או דרך האובייקטים של המחלקה המקורית או בתוך הקוד של המחלקה היורשת באמצעות המילה השמורה super.

## **עבודת כיתה-**

צור 4 קלאסים: Waiter, Manager, Accountant, HumanResources, Main  
בכל אחד מהם להגדיר משתנים, גטרים, סטרים, קונסטרוקטורים.  
הורשה

ליצור מערכים ב"משאבי אנוש" 1. מנהלים 2. חשב שכר 3. מלצר == 10 לכל אחד. כולם פרייבט.  
פונקציה שתכניס בלולאה לכל אחד.  
אחרי זה פונקציה נוספת שתדפיס את כל העובדים- לולאה שתדפיס את כל המשתנים שהורשו.  
• הדפסה עבור כל הנתונים- אלו שהורשו והאישיים לכל קלאס.

- אחרי עבודת הכיתה, נסיק מספר מסקנות:

- 1- שם הפונקציה תהיה זה אצל כולם.
  - 2- נשים את הפונקציה שאנחנו רוצים ליישם בפולימורפיזם בכל הפונקציות.
  - 3- ראשית מדובר במערך של אימפלויי ולכן הפונקציה תהיה במחלקת אב ואז נראה מי האובייקט. בפולימו מותר לשנות את הצורה של האב לאחד הבנים שלו, אם במערך יש את הבן צריך לרדת ולבדוק דור דור שבכולם יש את הפונקציה. בדור שבה זה נעצר היא המקום שבה תרוץ הפונקציה.
- חייבת להיות הפונקציה אצל האב כי היא מוגדרת על אימפלויי וכך הוא יורד מטה דור אחרי דור.

## שיעורי בית:

- להפוך למערך אחד, להוסיף פונקציה פרינט אול ואד אול, אד ניו וורקר- ואז הוא שואל איזה עובד? כשבחרים אחד מהם הוא עושה ניו לסוג הנבחר. בהוספת עובד הוא יוסיף את הפונקציה.
- להוסיף עוד קלאס שנקרא "מזכירה" והוא צריך לעבוד מתוך ההורשה.
- 1. הכינו מחלקות לעובדים
- 2. צרו מחלקת משאבי אנוש מערך בגודל 100 מסוג אימפלויי
- 3. ממשו בכל מחלקה, אבות ובנים פונקציות "הוסף עובד" ו"הדפס הכול"
- 4. הריצו בראשי באמצעו ראן
- 5. שיפורים:
  - 1. הוסיפו תפריט הכולל-
    - קליטת עובד
    - עריכת עובד
    - דוח עובדים
    - יציאה
  - תת תפריט – הוסף מנהל, הוסף מלצר, הוסף מנ"חש
  - 2. הוסיפו פתיח לאלקציה.
  - 3. הקפידו על הוואלידציות הבאות:
    - כל הערכים הקשורים בחישוב שכר יהיו חיוביים
    - אין לאפשר הכנסת שמות ריקים.
    - הגנו על התוכנה מהכנסת ערכים שגויים באמצעות try & catch.
    - 4. וודאו שכל המשתנים במחלקות הבנים יהיו מסוג פרייבט.
  - וכל המשתנים במחלקת האב מסוג פרוטקטד.

## תקציר שלבי עבודה:

1. פתיחת מחלקות רצויות.
2. הגדרת המשתנים בכל מחלקה.
3. גטרים וסטרים ברוחב לב.
4. קונסטרוקטורים לכל מחלקה- אחד ריק ואחד מלא.
5. להודיע לבנים שזה אבא שלהם: נוסיף בשורה של `public class` את המילים:  
**`extends ShemMachleketAv`**
6. לאחר הוספת `extends` (=הרחבה) כאשר נצהיר על אובייקט ממחלקה יורשת (תת-מחלקה) נראה כי היא מקבלת את הגטרים וסטרים של מחלקת האב.
7. מגדיר קונסטרוקטור נוסף ולמעלה בחץ נפתח נבקש את מחלקת האב. אחר כך תהיה גישה גם אליו בגטרים וסטרים. ואז הוא מאפשר לאתחל גם את המשתנים המקומיים של המחלקה וגם את אלו של האב.

## **קיצורים במקלדת:**

`Ctrl + shift + o` - מייבא את כל האימפורטים הנדרשים אוטומטית.