**Protected & super Constructor**

1. **Protected :**

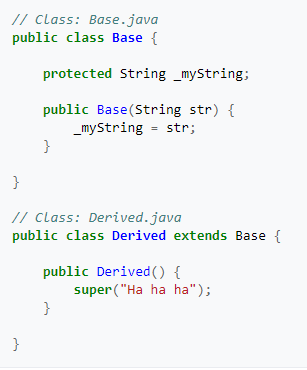
כמו שתי ההרשאות המוכרות לנו כבר, public ו-private, גם ההרשאה protected ניתנת לשימוש עם משתנים ועם שיטות. ההבדל הוא שהגישה מותרת רק למחלקות היורשות. אם כך:

* **Private -** לא מאפשרת גישה לאף מחלקה חיצונית, ובכלל זה גם לא למחלקות יורשות.
* **Protected -** מאפשרת גישה למחלקות יורשות ולמחלקות באותה החבילה, לא מאפשרת גישה למחלקות חיצונית אחרות.
* **Public-** מאפשרת גישה לכל מחלקה.

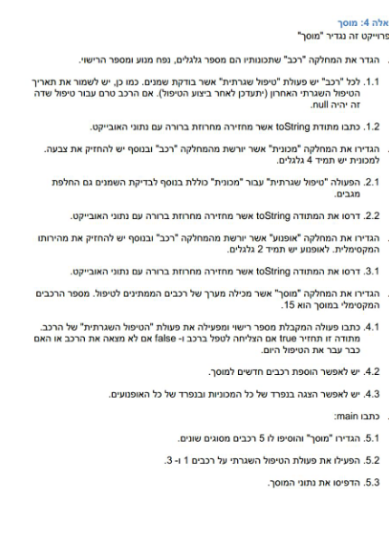
**הכלל**, כמו שהכרנו עד עכשיו, הוא לשמור על עיקרון הכימוס באמצעות הגבלה של הגישה, ככל האפשר

1. **בנאים(constructor):**

נניח כי יש לנו מחלקה בשם Base, ומחלקה בשם Derived, שיורשת אותה (כלומר - Derived extends Base. אובייקט של מחלקה יורשת (Derived) למעשה מכיל, בין השאר, אובייקט של המחלקה אותו הוא יורש (Base). לכן, בכל פעם שיוצרים בעזרת הבנאי אובייקט כזה, צריך לדאוג גם ליצירת האובייקט שמתחתיו בעזרת פנייה לבנאי שלו, מה שאומר במקרה שלנו שהבנאי של המחלקה Derived חייב להכיל פנייה לבנאי של המחלקה Base. מקרה יוצא דופן הוא כאשר המחלקה אותה הוא יורש מכילה בנאי ריק (כלומר, בנאי שלא צריך לקבל פרמטרים) - המהדר מוסיף בצורה אוטומטית בתחילת הבנאי של המחלקה היורשת פנייה לבנאי של המחלקה אותה הוא יורש. זה המקרה בדוגמה למעלה, ולכן שם לא היה צורך בפנייה לבנאים. הפנייה עצמה מתבצעת בעזרת המילה השמורה super, בצורה דומה לשימוש במילה this. כמו עם המילה this, גם כאן הפנייה חייבת להיות בשורה הראשונה של הבנאי.

**דוגמא :**  קריאה ממחלקה שיורשת ל-constructor ממחלקת "אב".

**תרגיל לחזרה על החומר**

****