

מבוא לתכנות II

הרצאה 4 – OOP – הורשה
סמסטר 2

תכנות מונחה עצמים - OOP

- OOP – Object Oriented Programming.
- OOP היא ארכיטקטורת התכנות הנפוצה ביותר כיום.
- אחד העקרונות החשובים ביותר הוא עקרון ההורשה (לצד כימוס, הפשטה ורב צורתיות)
- הורשה מאפשרת לנו שימוש חוזר בקוד כתוב, כדי שנוכל:
 - להמנע משכפול קוד במחלקות.
 - שימוש במחלקה קיימת כבסיס למחלקה מרחיבה (יורשת) שלה.

הורשה - מימוש

- על מנת לממש הורשה בין מחלקות עלינו להגדיר עבור המחלקה הנגזרת (derived) מהי המחלקה ממנה היא יורשת – מחלקת הבסיס (base).

```
class Base
{

}

class Derived : Base
{

}
```

- כעת, מחלקת *Derived* מכילה את כל מה שקיים במחלקת *Base* ללא צורך לשכפל את הקוד מחדש.

נושאים חשובים

- הורשה היא קשר בין מחלקות אשר מאופיין ע"י "הוא/היא" או "מסוג של" למשל:
 - משולש הוא צורה.
 - מלבן הוא סוג של צורה.
 - המרצה הוא סוג של אדם 😊.
 - סטודנט הוא בן אדם 😊.
- שדות המחלקה מהווים קשר אשר מאופיין ע"י "יש" או "מכיל/ה" למשל:
 - לאדם יש שם.
 - מכונית מכילה 4 גלגלים.
 - לכלב יש זנב.

דברים חשובים

- מחלקה יכולה לרשת רק מחלקה אחת.
- מבנה (סוג של מחלקה אך עם פחות יכולות וזהו סוג ערך ולא הפניה. אנחנו לא נלמד את זה) לא יכול לרשת כלל.
- לא ניתן לרשת ממחלקה חסומה (*sealed*).
- יורשים הכל מהמחלקה לרבות: שדות, תכונות, פונקציות. בנאים הם יוצאי הדופן.
- משתנים פרטיים (*private*) מועברים למחלקה הנגזרת, אך אינם נגישים ממנה.
- משתנים ציבוריים נגישים מכל מחלקה.

בנאי

- במחלקה הנגזרת נצטרך לקרוא לבנאי מתוך מחלקת הבסיס.
- הבנאי של מחלקת הבסיס ייקרא לפני של המחלקה הנגזרת.
- ברירת המחדל לקריאת בנאי מחלקת הבסיס היא הבנאי ללא הפרמטרים (הבנאי הריק/דיפולטיבי).
- קריאה לבנאי של מחלקת הבסיס מתבצעת באמצעות מילת המפתח *base* ועל ידי שימוש שומה לשיטה בה השתמשנו לקריאה לבנאי אחר בתוך אותה מחלקה (`this()`)

```
class Base
{
    public Base(int x)
    {
        //...
    }
}

class Derived : Base
{
    public Derived(int x, int y)
        :base(x)
    {
        //...
    }
}
```

- לשימושינו קיימת הרשאת הגישה "מוגן" (*protected*).
תכונותיה:

- ציבורית עבור המחלקה הנגזרת ועבור כל המחלקות היורשות ממנה...

- פרטית עבור כל השאר.

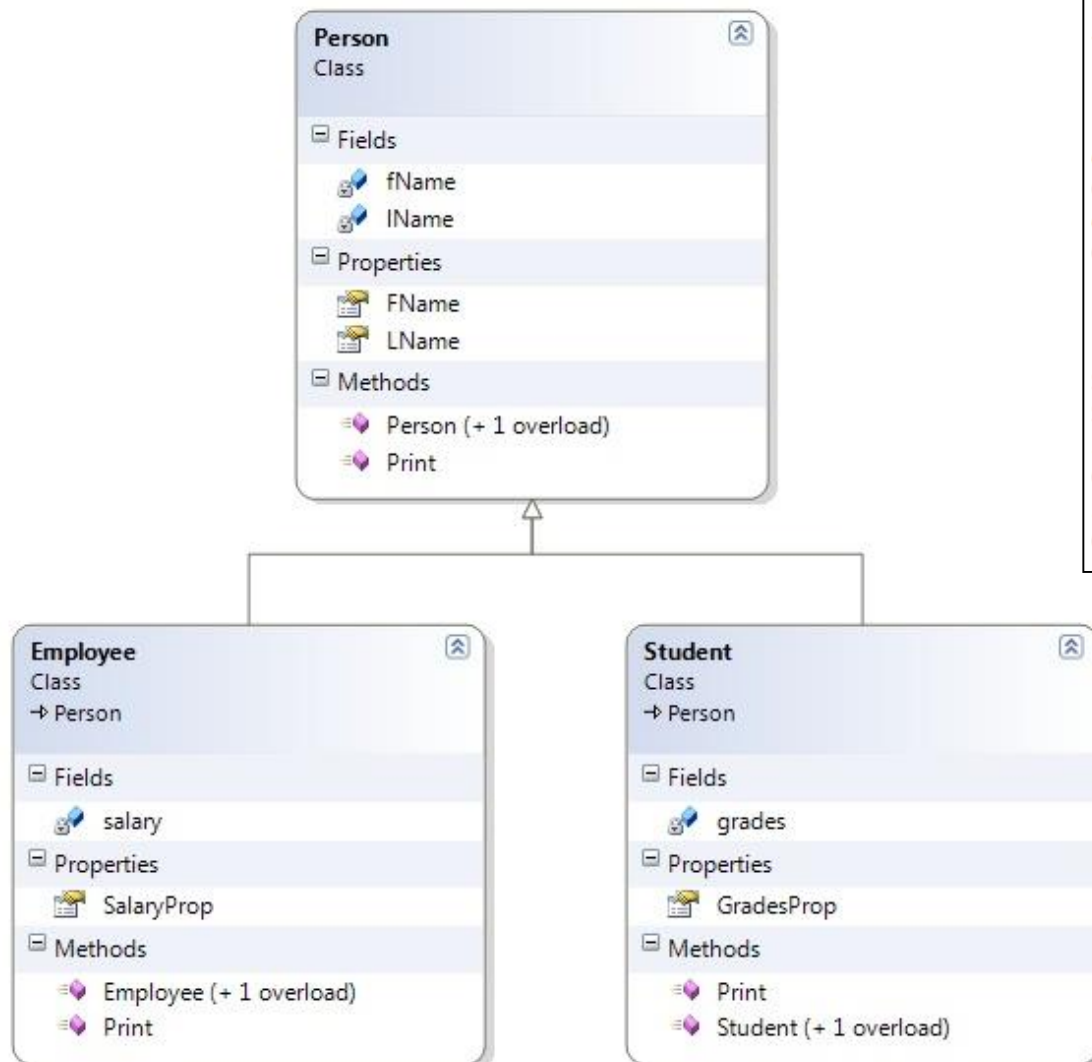
בסיס וחדש

- לעתים במחלקת הבסיס קיימת פונקציה אשר לא עומדת בדיוק בדרישות עבור המחלקה הנגזרת, למשל פונקציית הדפסת פרטים.
- נוכל לרשום את אותה הפונקציה במחלקה הנגזרת, אך ניאלץ להוסיף את מילת המפתח *new* לפני סוג ההחזר (זה יוביל לאזהרה ולא לשגיאה).
- נוכל לקרוא לכל פונקציה אשר קיימת במחלקת הבסיס באמצעות:
base.function()

```
class Base
{
    public void Print()
    {
        //...
    }
}

class Derived : Base
{
    public new void Print()
    {
        //...
        base.Print();
        //...
    }
}
```


דוגמה 1 – אדם וסטודנט



```
Person p1 = new Person("a", "b");
// Person p2 = new Person();

Employee e1 = new Employee();
Employee e2 = new Employee("aaa", "vvvv", 25000);
e1.Print();
Console.WriteLine("-----");
e2.Print();
Console.WriteLine("-----");

int[] grades = new int[] {1,5,7};
Student s1 = new Student();
Student s2 = new Student("asfsd", "safsdf", grades);
s1.Print();
Console.WriteLine("-----");
s2.Print();
Console.WriteLine("-----");
```

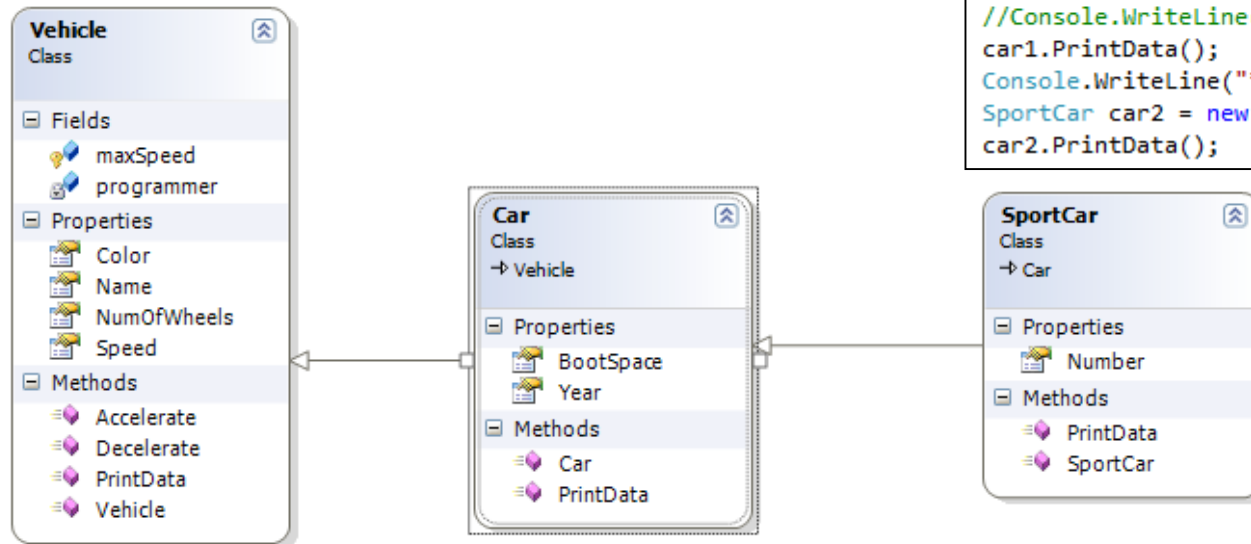
```
FName: a
LName: a
Salary: 0
-----
FName: aaa
LName: vvvv
Salary: 25000
-----
FName: a
LName: a
0, 0, 0, 0, 0.
-----
FName: asfsd
LName: safsdf
1, 5, 7.
-----
Press any key to continue . . .
```

דוגמה 2 - מכוניות

```
0
VEHICLE DATA:
Color: , Name: , Number of wheels: 0, Speed: 0, max speed: 0, programmer: Lady GAGA.
*****
0
VEHICLE DATA:
Color: , Name: , Number of wheels: 4, Speed: 0, max speed: 100, programmer: Lady GAGA.
CAR DATA:
Year: 0, Bootspace: 0.
*****
VEHICLE DATA:
Color: , Name: , Number of wheels: 3, Speed: 0, max speed: 200, programmer: Lady GAGA.
CAR DATA:
Year: 0, Bootspace: 0.
SPORTCAR DATA:
Number: 22
Press any key to continue . . . _
```

```
Vehicle vec1 = new Vehicle();
Console.WriteLine(vec1.NumOfWheels);
//Console.WriteLine(vec1.maxSpeed); //ERROR because protected
vec1.PrintData();

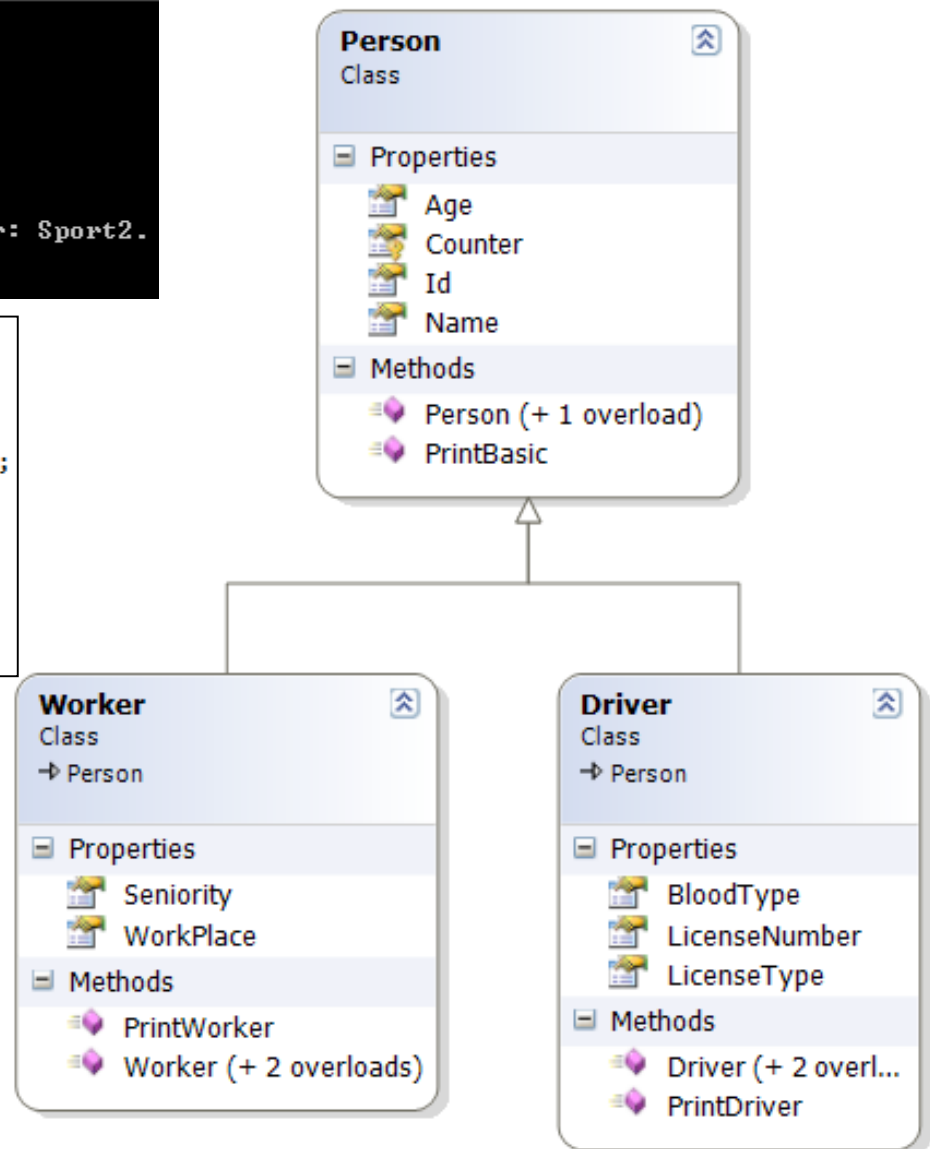
Console.WriteLine("*****");
Car car1 = new Car();
Console.WriteLine(car1.BootSpace);
//Console.WriteLine(vec1.maxSpeed); //ERROR because protected
car1.PrintData();
Console.WriteLine("*****");
SportCar car2 = new SportCar();
car2.PrintData();
```



דוגמה 3 – אדם ועובד

```
i'm the 3 person!  
Name: Joe Doe, Age: 0, ID: 0.  
Workplace: Intel, Seniority: 5.  
  
i'm the 3 person!  
Name: Nir, Age: 33, ID: 123456567.  
Workplace: Hi-Tech, Seniority: 8.  
i'm the 4 person!  
Name: Joe Doe, Age: 0, ID: 0.  
Blood type: A+, License Number: 123456, License Number: Sport2.  
Press any key to continue . . . _
```

```
Person per = new Person(33, "Nir", 123456789);  
//per.PrintBasic();  
  
Worker work = new Worker("Intel", 5);  
Worker work2 = new Worker(33, "Nir", 123456567, "Hi-Tech", 8);  
work.PrintWorker();  
Console.WriteLine();  
work2.PrintWorker();  
  
Driver driver = new Driver("A+", 123456, License.Sport2);  
driver.PrintDriver();
```



דוגמה 4 – קופסאות

```
CanBox can1 = new CanBox("Hamin",5, 2);
//can1.Diameter = 2;
//can1.Label = "Hamin";
//can1.Height = 5;
can1.CalcCapacity();
can1.PrintData();

PlasticBox plastic1 = new PlasticBox("Luf!", 5,6,10);
//plastic1.Height = 5;
//plastic1.Length = 6;
//plastic1.Width = 10;
//plastic1.Label = "Luf!";
plastic1.CalcCapacity();
plastic1.PrintData();
```

```
Box data: Hamin, capacity: 47.1.
Box data: Luf!, capacity: 300.
Press any key to continue . . .
```

