סיכום כללי פולימורפיזם

1. הכלל הבסיסי – מופע של מחלקה יורשת (צד ימין) יכול להיכנס למצביע מסוג בסיס (צד שמאל).

Person p = new Student();

צד שמאל – מצביע של מחלקת בסיס

צד ימין – מופע של מחלקה יורשת

2. אפשר להפעיל רק פונקציות של צד שמאל. (ושל האבות שלו)

p.setName(….)

אבל לא – p.setMark(….)

3. הפונקציה שתופעל בזמן ריצה – הפונקציה הקרובה ביותר למופע בצד ימין.

כלומר הדורסת. אם אין דורסת – תופעל הבסיסית או הקרובה ביותר בהיררכיה.

לדוג':

Person - > Employee - > Manager

Person p=new Manager();

p.func();

אם יש פונקציה דורסת ב-Manager, היא תופעל.
אם אין פונקציה דורסת ב-Manager, תופעל הדורסת של Employee,
ואם גם אצלה אין, תופעל הבסיסית של Person.

4. המרה כלפי מעלה/מטה לממשק – יש לשים לב אם האובייקט מממש את הממשק בצורה ישירה או עקיפה.

אם כן ניתן לבצע המרה לממשק.

סיכום כללי פולימורפיזם

5. השמות בין אובייקטים (מסתכלים רק על המצביע- צד שמאל) – ניתן לבצע בין:

- אותו סוג.
- אובייקט של מחלקה יורשת. (המרה כלפי מעלה)
- אובייקט שביצענו לו המרה כלפי מטה.

לדוג':

```
Person p = new Student();

Manager m=new Manager();

Person m2=new Manager();
```

p=m; ✓
m=m2; //רק בהמרה כלפי מטה זה יהיה תקין ☒
m2=m; ✓
p=m2; ✓

6. שליחה לפונקציה (מסתכלים רק על המצביע-צד שמאל) – פונקציה המקבלת כפרמטר אובייקט יכולה לקבל:

- אותו סוג.
- אובייקט של מחלקה יורשת. (המרה כלפי מעלה)
- אובייקט שביצענו לו המרה כלפי מטה.

לדוג':

```
Person p = new Student();

Manager m=new Manager();

Person m2=new Manager();


public void func(Manager m){}
```

func(p);// שגוי בכל מצב אין אפשרות לשלוח ☒
func(m); ✓
func(m2); //רק בהמרה כלפי מטה זה יהיה תקין ☒