

פרק 8 - ממשקים

עמ' 244 - דף עבודה מס' 1
תרגיל תיאורטי

לפניכם קטע קוד ללא שגיאות:

1. כתבו את A במלואו בהנחה שהוא ממשק interface
2. כתבו את המחלקות B ו-C

```
public class TargilTheoretY
{
    /**
     * פרק 8 - ממשקים - תרגיל תיאורטי עמ' 244
     */
    public static void main(String[] args)
    {
        int x;
        String y;

        x = 2;
        y = "bbb";

        C b1 = new B();
        System.out.println("b1: "+b1.toString());

        A[] arrA = new A[3];

        arrA[0] = new B(x, y);
        arrA[1] = new B();
        arrA[2] = new C();

        arrA[0].f();
        arrA[2].g();
        arrA[1].h();
    }
}
```

ההוראות המוזחות ב-tab אינן מופיעות בתרגיל המקורי

המרה (אוטומטית) כלפי מעלה: האובייקט מתת-המחלקה מומר להפנייה מטיפוס מחלקת על

לא ניתן ליצור מופע מטיפוס הממשק. מותר ליצור מערך מטיפוס הממשק, וזאת מפני שההוראה יוצרת מערך של הפניות ולא של מופעים מטיפוס הממשק.

```
/*
b1: x: 5
    y: aaa

f() from class C
g() from class C
h() from class B
*/
```

שיאו ♥: פתרון זה אינו הפתרון האפשרי היחיד לתרגיל

```
public interface A
{
    public final int num = 5;

    public void f();
    public void g();
    public void h();
}
```

קבוע המוגדר בממשק, מוכר בכל המחלקות המממשות את הממשק, או יורשות מהמחלקה המממשת את הממשק

```
public class C implements A
{
    public C() //--- פעולה בונה ---
    {
    }

    public void f()
    {
        System.out.println("f() from class C");
    }
    public void g()
    {
        System.out.println("g() from class C");
    }
    public void h()
    {
        System.out.println("h() from class C");
    }
}
```

```
public class B extends C
{
```

```
    private int x;
    private String y;
```

```
    public B()
    {
        this.x = num;
        this.y = "aaa";
    }
```

המחלקה B יורשת מ-C המממשת את הממשק A. קבוע המוגדר בממשק A מוכר גם במחלקה היורשת B.

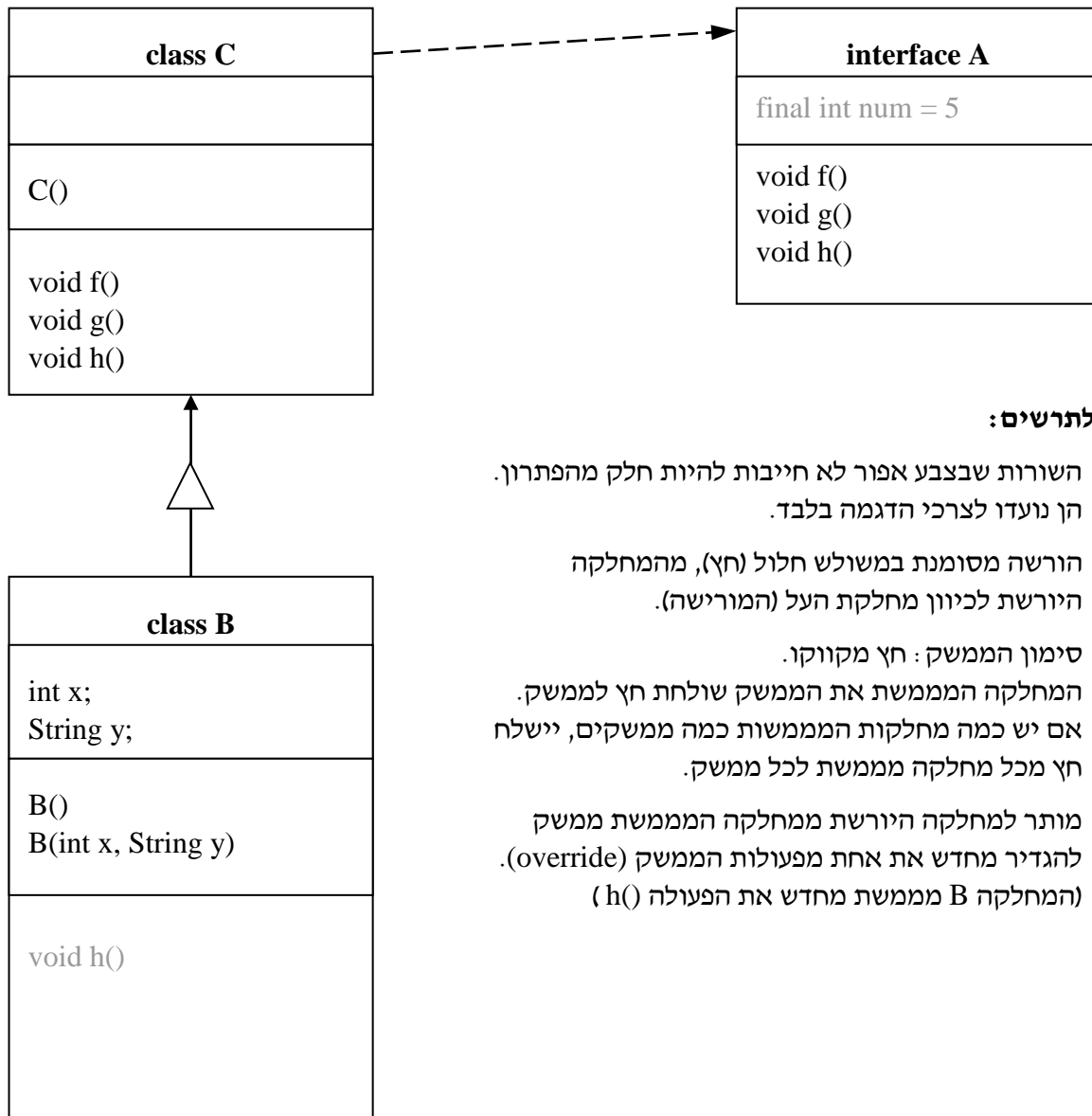
```
    public B(int x, String y)
    {
        this.x = x;
        this.y = y;
    }
```

פעולה של הממשק הממומשת במחלקה B (שאינה מממשת את A), דורסת (override) ומגדירה אותה מחדש.

```
    public void h()
    {
        System.out.println("h() from class B");
    }
```

```
    public String toString()
    {
        return "x: " + this.x + "\n" + "\t y: " + this.y + "\n";
    }
}
```

3. שרטטו תרשים UML המציג את הקשרים המדויקים בין A, B ו-C :



הערות לתרשים:

- השורות שבצבע אפור לא חייבות להיות חלק מהפתרון. הן נועדו לצרכי הדגמה בלבד.
- הורשה מסומנת במשולש חלול (חץ), מהמחלקה היורשת לכיוון מחלקת העל (המורשה).
- סימון הממשק: חץ מקווקו. המחלקה המממשת את הממשק שולחת חץ לממשק. אם יש כמה מחלקות המממשות כמה ממשקים, יישלח חץ מכל מחלקה מממשת לכל ממשק.
- מותר למחלקה היורשת ממחלקה המממשת ממשק להגדיר מחדש את אחת מפעולות הממשק (override). (המחלקה B מממשת מחדש את הפעולה h())

4. האם C יכול להיות ממשק ?

תשובה: C לא יכולה להיות ממשק, מכיוון שאנו יוצרים אובייקט מסוג C. לא ניתן ליצור אובייקטים מטיפוס הממשק (אבל מותר ליצור הפנייה מסוג הממשק).