

ג) אי אפשר, אנחנו מסתמכים בפייתון על פונקציונליות הניתנת לנו באמצעות `__call__` שאין לה מקבילה בג'אבה, ניתן לממש חלק מועט מהדקורטור, ספציפית לספור כמה פעמים השתמשנו בו – ע"י קידום משתנה סטטי מהקונסטרקטור כאשר פותחים אינסטנס חדש. אך לכל השאר אנחנו מוגבלים מכיוון שבג'אבה דקורטורים הם מחלקות, ולא פונקציות או `callable`.

יש סוג של מימוש מוגבל שאפשר לעשות אבל הוא לא עונה לנו על הצרכים בשאלה; אפשר לממש תבנית עיצוב `Decorator` שעוטפת במקום הפונקציה ממשק בעל חתימה של הפונקציה ואת הפונקציה מכריזים כקלאס אנונימי או כביטוי לאמבדה. כתבתי תוכנית קצרה המדגימה מימוש זה:

```
public class Lambda {
    public static void main(String[] args) {
        EmptyFunc emptyFunc = () -> System.out.println("empty func");
        EmptyFuncDecorator efd = new EmptyFuncDecorator(emptyFunc);
        efd.run();
        System.out.println(efd);

        EmptyFunc emptyFunc_2 = () -> System.out.println("wow");
        EmptyFuncDecorator efd2 = new EmptyFuncDecorator(emptyFunc_2);
        System.out.println(efd2);

        FuncWithParams<String, String, Integer> funcWithParams = (txt, n) -> {
            System.out.println("doing calculations...");
            return txt + ", " + n;
        };
        System.out.println(funcWithParams.apply("func with params", 3));
    }
}

interface EmptyFunc {
    void run();
}

interface FuncWithParams<R, X, Y> {
    R apply(X paramX, Y paramY);
}

class EmptyFuncDecorator implements EmptyFunc {
    int func_calls_num;
    static int decorator_calls_num = 0;
    EmptyFunc func;

    EmptyFuncDecorator(EmptyFunc func) {
        this.func = func;
        decorator_calls_num += 1;
        func_calls_num = 0;
    }

    @Override
    public void run() {
        func_calls_num += 1;
        func.run();
    }

    @Override
    public String toString() {
        return "This func calls: " + func_calls_num + " | Total Decorator Calls: " +
            EmptyFuncDecorator.decorator_calls_num;
    }
}
```

כפי שניתן לראות צריך לפתוח דקורטור נפרד לכל חתימת פונקציה שונה לפי הפרמטרים שהיא מקבלת ולפי מה שהיא מחזירה, כך שזה לא ישים לדרישה שלנו.