

נושאים מתקדמים בתכנות מונחה עצמים: תרגיל מעבדה 28/06/23

התרגיל עוסק בתבניות תכן של מבנים ושל התנהגות (*Structural + Behavioral DPs*) – על בסיס החומר שנלמד לאחרונה.

1. נתון המודול `DoubleDispatchDemo_TODO.cpp` – ובו המחלקות `Processor`, `Event` וכן דרייבר ופונקציית עזר ליצירת *Events* אקראיים – הדורשים שניהם מימושים (ראו כיתוב סמסד לאורך קוד המקור).

על הדרייבר ליצור רשימה אקראית של *Events* ולהריץ סימולציית **Double-Dispatch**, כלומר - לצמד כל *Event* לכל *Processor*.

בסה"כ, הפיתרון נדרש לכלול את המימושים הבאים:

- מחלקת בסיס `Processor` (**TODO1**)
- שתי מחלקות קונקרטיות (**TODO2+TODO3**) עם מימושים התומכים בביקורים הנדרשים
- פונקציית `main` שבה מתרחשים לפחות ארבעה ביקורים (אינטראקציה בין שני מבקרים קונקרטיים לשני אלמנטים קונקרטיים) (**TODO4**)

2. נתון המודול `CompositeStructure.java` – ובו המחלקות `{Component, Leaf}` וכן דרייבר בשם `CompositeStructure`.

ממשו מחלקה בשם `Node` כמימוש תבנית *Composite* של `Component`.

מומלץ לעשות שימוש באוסף הגנרי `ArrayList<>`.

3. נתון לכם פרוייקט המגדיר מנשק "אסטרטגיית תן-לי" של ילד וכן מחלקות קונקרטיות המייצגות פניות אל אבא/אמא/סבא/סבתא. מטרת התרגיל היא ליצור טיפוס קונקרטי "תן-לי" בשם `Gimme`, היורש גם הוא מ-"אסטרטגיית תן-לי" כמימוש של תבנית **Chain-Responsibility**, באופן שיתמוך בפונקציונליות של הדרייבר הנתון באותו קובץ.

אסטרטגיה זו מכילה את סדרת הפניות כברירת-מחדל – והרעיון הוא לממש זאת באמצעות אוסף כללי, ולא דווקא כרשימה מקושרת כפי שהוצג בהרצאה.

הגדירו מחלקה בשם זה, ממשו אותה בתקן `C++0x`, והריצו את הדרייבר הנתון.

4. שכתבו את הפתרון שלכם לשאלה-2 בשפת `C++0x`.

בהצלחה רבה!