

הרצאה #7

בכל תחום צריך מספר מונחים כדי שכולם ידעו מה הנושא שאנו מדברים עליו. בעולם התכנות קוראים לרעיונות אלו Design Patterns.

בהרצאה זו נעבור על מספר תבניות עיצוב שכאלו כפי שהם מופיעים בספר Pattern-Oriented software architecture. Vol 2: (להלן POSA וגם GoF)

- Singleton

תבנית עיצוב המיועדת ליצירת אינסטנט יחיד. נבדלת ממשתנה גלובלי כלשהי בגלל שעובד עם מיוטקס (Mutex), על מנת ליצור מצב ששני טרדים לא יכולים ליצור בו זמנית.

נלמד במסגרת קורס מונחה עצמים, על כן פחות רלוונטי.

- Factory Method

תבנית העיצוב "שיטת המפעל" היא תבנית עיצוב יוצרת (creational design pattern) שמספקת ממשק ליצירת אובייקטים במחלקה עליונה (super class), אך מאפשרת לתת-מחלקות לשנות את סוג האובייקטים שיווצרו. זה מתבצע על ידי הגדרת מתודה שמשתמשת ליצירת אובייקטים, שבדרך כלל מכנים "מתודת המפעל". תת-המחלקות יכולות לדרוס את המתודה זו כדי לשנות את מחלקת האובייקטים שיווצרו.

תבנית "שיטת המפעל" משמשת כאשר מחלקה לא יכולה לחזות את סוג האובייקטים שהיא צריכה ליצור ורוצה לאפשר לתת-מחלקות שלה לציין את האובייקטים הללו. זו דרך להעביר את תהליך היישום לתת-מחלקות, מקדמת אינטגרציה חופשית ושימוש חוזר בקוד.

התבנית מכילה את החלקים הבאים:

- מוצר
- זה האינטרפייס של טיפוס האובייקטים שהתבנית מייצרת
- מוצר מסוים
- זהו האובייקט שמממשת את האינטרפייס לעיל
- יצרן
- מחלקה אבסטרקטית בעלת ההצהרה של התבנית. היצרן מחזיר אובייקט מסוג המוצר
- יצרן מסוים
- מחלקה זו מרחיבה את היצרן ומחליפה את תבנית העיצוב למופע (אינסטנס) של מוצר מסוים

בעצם זה האש (או מיפוי אחר) הממפה בין מפתח (לדוגמא סוג הסוקט) לבין הבנאי שלו (בונה את סוג הסוקט הנכון)

דוגמא לתבנית עיצוב זו שראינו בהרצאה זוהי הקריאה ל-socket(2)

נלמד במסגרת קורס מונחה עצמים, על כן פחות רלוונטי.

- Adapter

תבנית עיצוב זו היא כגשר בין שני אינטרפייסים. תבנית זו מיועדת להתאמת אינטרפייס אחד לחברו, גם כאשר אין בהכרח קשר כלשהו בין שני האינטרפייסים ובתבנית זו הם יעבדו יחדיו.

לדוגמא אם רוצים – טרד שממתין לקונד וכאשר הקונד מגיע – שולח הודעה לסוקט – כדי לאפשר לנו לעשות select על קונד.

נלמד במסגרת קורס מונחה עצמים, על כן פחות רלוונטי.

• Façade

התבנית היעילה ביותר לניהול ופישוט מערכות תכנה מורכבות.

מעין מחלקה עוטפת, שמטרתה לקחת מערכת מורכבת ולעטוף אותה ב-API פשוט להבנה. נוח לנטר בעזרת תבנית זו איך מערכת מורכבת יותר עובדת.

דוגמא – FD אלו למעשה Façade כי הפונק' שלו מאימות גם לקבצים, גם לסוקטים וכדומה.

• Reactor

תבנית עיצוב התנהגותית (behavioural design pattern) המיועדת למערכות המונעות ע"י events. בפרט, מיועד במקרים כאשר אפליקציות צריכות להתמודד עם בקשת שירותים ממספר לקוחות במקביל.

הרעיון המרכזי של תבנית זו הוא לטפל בבקשות שירות מרובות המועברות לאפליקציה. כל שירות באפליקציה עשוי לדרוש מתודה נפרדת לטיפול בבקשות, והאפליקציה לא יודעת אילו בקשות יגיעו ומתי. תבנית זו מאפשרת תזמון יעיל של שירות אלו על ידי פילוג (demultiplexing) של הבקשות הנכנסות וטיפול בבקשות החמות בלבד על ידי טרד יחיד באופן סדרתי. (למשל על ידי select(2)) התבנית מכילה את החלקים הבאים:

○ Reactor

המתזמן שדואג לפילוג הבקשות לevent handlers המתאימים

○ Sync Event Demultiplexer

חלק זה הוא שיושב על מספר מקורות לאירועים אפשריים ומחכה שאחד מהם יתעורר ויהיה פעיל (יגיש בקשה). ברגע שאחד מהם מתעורר, הוא מעביר את השליטה ל – Reactor

שיריץ את ה- event handler המתאים

○ Event Handler

אלו הם האובייקטים מותאמי האפליקציה אשר משויכים לאירועים ספציפיים. הם מעבדים

את האירועים אשר מועברים אליהם על ידי ה-Reactor

נשים לב שreactor מטפלת בבקשות סידרתיות. בטרד יחיד (כמו בדוגמאות של beej select(2) וpoll(2)). יתכנו תבניות אחרות שמטפלות בכל בקשה בטרד חדש או קבוצת טרדים שמטפלות בבקשות שלא התעמקנו בהם.

ריאקטור מתאים לטיפול מהיר יחסית בכל בקשה.

• Active Object

טרד עם תור עבודה.

הטרד נמצא בביז' לופ כאשר מה שהוא מבצע – ממתין שיש בקשה בתור (למשל על cond) ברגע שמגיעה בקשה מוציא אותה ומטפל בה.

• Pipeline

תבנית עיצוב זו מארגנת מערכת לסדרה של שלבים, שכל אחד מבוצע על ידי active object בודד שמעבד נתונים ומעביר את הנתונים המעובדים ל(AO (Active Object הבא הנתונים עוברים דרך ה-pipeline כמו שמים עוברים דרך צינור. הפלט של שלב אחד הופך לקלט של השלב הבא. לכל שלב יש תפקיד ומשימה מוגדרים מראש וכל אחד מהם עובד ללא תלות משבליים האחרים.

דוגמא לתבנית עיצוב pipeline:

מערכת באינטרנט שמקבלת בקשות מוצפנות
כל בקשה עוברת שרשרת פעולות ב5 טרדים שונים
טרד א – מבצע פיענוח של הצפנת הבקשה.
טרד ב – מבצע פיענוח של קידוד הבקשה
טרד ג – עונה על הבקשה

טרד ד – בונה את התשובה
טרד ה – מצפין את התשובה ושולח אותה.