

פרויקט סיום – תיכון מונחה עצמים

Builder (1)

קטע קוד:

הוספנו מחלקה חדשה (Truck).

קטע קוד חדש:

יצרנו ב model את המחלקה truck.
ב model המימוש ל builder בשורה 19.

מצב קיים:

יצירת משאית על ידי בנאי ברירת מחדל ללא תכונות והצגתה על המסך.

הסבר:

בחרנו להרחיב את הפונקציונליות של המשאית על ידי הוספת תכונות, עשינו זאת על ידי שימוש של builder בדרך שבה בכל פעם שתיווצר משאית (במידה וירצו ליצור יותר מאחת) יוכלו להוסיף לה תכונות בצורה יעילה יותר ונוחה יותר ולא יהיה צורך במספר רב של בנאים.

Singleton (2)

קטע קוד:

קובץ model שורה 13.
קובץ stackOnTruck שורה 7, 11.

קטע קוד חדש:

קובץ model שורה 13.
קובץ stackOnTruck שורה 7, 12, 19-32.

מצב קיים:

ניתן היה ליצור כמה מחסניות של מכולות על משאית אחת.

הסבר:

בחרנו להשתמש בתבנית singleton מכיוון שהיא מאפשרת לנו ליצור אובייקט אחד בלבד מסוג stackOnTruck, ודבר זה הכרחי היות שלא יכול להיות בפועל שתי מחסניות על אותה המשאית.

Prototype (3)

קטע קוד:

הוספנו מחלקה חדשה (Ship).

קטע קוד חדש:

קובץ model שורה 28-36, 14.

קובץ Ship חדש.

מצב קיים:

הספינה הייתה אובייקט פשוט שכולל רק ציור בחלון ללא שום תכונות.

הסבר:

על מנת לממש תבנית עיצוב זו יצרנו ספינת ברירת מחדל ובחרנו להוסיף לה תכונות מסוימות, כך נוכל לשכפל אותה בכל פעם מבלי ליצור מופע של המצב החדש. בפועל הספינה שאנו משתמשים בה היא clone של ספינת ברירת המחדל, עם עדכון של שתי תכונות.

Iterator (4)

קטע קוד:

קובץ ListOfStack שורה 47-53.

קטע קוד חדש:

קובץ ListOfStack שורה 57-104.

מצב קיים:

מעבר על array list בעזרת לולאת for רגילה.

הסבר:

כעת נוכל לגשת ברצף לכל stack ב list מבלי לחשוף את המבנה הפנימי של listn. כמו כן, נוכל בעתיד לשנות את סוג מבנה הנתונים מבלי לשנות את לולאות החיפוש, או להוסיף לקוד לולאות נוספות ולהשתמש באותו Iterator.

Null object (5)

קטע קוד:

הוספנו פונקציות חדשה.

קטע קוד חדש:

קובץ NullContainer, AbstractContainer.

קובץ StackOfConts שורות 80-88.

מצב קיים:

לא קיימת פונקציה לקבלת מכולה מסומנת.

הסבר:

על מנת לממש null object הוספנו פונקציה אשר עוברת על stack עד אשר מוצאת מכולה מסומנת, המימוש בא לידי ביטוי כאשר לא נמצאה מכולה מסומנת ובמקרה זה תחזור למשתמש NullContainer.

State (6)

קטע קוד:

קובץ Container שורות 10, 22, 28, 33, 78.

קטע קוד חדש:

קבצים IContState, MarkOn, MarkOff.

קובץ Container שורות 11, 37-47, 45, 89-103.

מצב קיים:

מצב המכולה מתואר ע"י משתנה בוליאני והתוכנית לא פתוחה להרחבות.

הסבר:

המכולה תכיל בדיוק אחד משני המצבים: מסומנת/ לא מסומנת, ולא תהיה אפשרות לבחור עבורה מצב אחר. כל הפעולות החיצוניות על המכולה/ מחסנית המכילה מכולות יהיו תלויות במצבה הפנימי של המכולה. כמו כן, נוכל בעתיד להוסיף/ לשנות את מצבי המכולה האפשריים בקלות.

Command (7)

קטע קוד:

קובץ StackOfConts שורות 45-52.

קטע קוד חדש:

קבצים ContainerCargo, SetCargoCapacityCommand, Command.
קובץ Container שורות 12, 27-34.

מצב קיים:

אין מידע לגבי תכולת כל קונטיינר.

הסבר:

בחרנו להוסיף מידע לגבי תכולת כל מכולה (תכולה מלאה/ ריקה) על ידי דפוס העיצוב Command, וכך הפכנו בקשה/ התנהגות (עדכון לגבי תכולת מכולה מלאה) לאובייקט נפרד הכולל את כל המידע לגבי אותה הבקשה או ההתנהגות ועל ידי כך ביצענו אינקפסולציה של כל המידע שצריך על מנת לבצע את אותה הפעולה. בעזרת שיטה זו יהיה יותר נוח בעתיד להוסיף תכונות חדשות או פונקציות נוספות למכולה או לאובייקטים שונים אשר בעלי אותה התכונה.

Facade (8)

קטע קוד:

קובץ ListOfStacks שורות 35-45.

קטע קוד חדש:

קבצים SetMarked, SetMarkedService.
קובץ ListOfStacks שורות 37-41.

מצב קיים:

קיימת לוגיקה מסובכת לסימון מכולה. כמו כן, הלוגיקה שייכת לאובייקט של מחסנית המורכבת ממכולות בלבד.

הסבר:

יצירת ממשק פשוט לשימוש ש"מחביא" את הלוגיקה המורכבת. ע"י השימוש בממשק זה, הקוד יהיה ברור ופשוט יותר. כמו כן, ניתן יהיה להשתמש בממשק זה גם במידה ונשתמש במבנה נתונים אחר/ המחסנית תכיל אובייקטים אחרים.

Decorator (9)

קטע קוד:

קובץ ShipView שורה 17.

קטע קוד חדש:

קבצים ShapeColor, RedColor, YellowCollor, ColorDecorator
קובץ ShipView שורות 22-25.

מצב קיים:

צבע הספינה אינו דינמי ואינו חלק מתכונות הספינה.

הסבר:

הוספנו ממשק המגדיר תכונה של צבע (בתכנית שלנו לאובייקט ShipView) באופן דינמי תוך שמירה על הממשק. בדרך זו, לא פגענו בתכונות הספינה, ונוכל לשנות את תכונות הצבע/ להסיר אותה בקלות בכל עת. כמו כן, כעת נוכל לממש את ממשק הצבע באופן פשוט על כל אחד מהאובייקטים בתוכנית.

Visitor (10)

קטע קוד:

קובץ Truck שורה 5.

קטע קוד חדש:

קבצים ITruckInfo, TruckVisitor, TruckDriverVisitor
קובץ Truck שורות 5, 64- 69.

מצב קיים:

שם נהג המשאית מוגדר כתכונה קבועה לאותה המשאית, למרות שבפועל יכולים להיות נהגים שונים לאותה המשאית.

הסבר:

כדי שנוכל לשנות את פרטי נהג המשאית באופן חיצוני ודינאמי מבלי לשנות את המשאית עצמה בכל פעם, הוספנו visitor למחלקה Truck. זאת כדי שנוכל לשנות תכונה מסוימת באובייקט (בתכנית שלנו- את פרטי הנהג TruckDriver) באופן חד פעמי/עתי מבלי לשנות את האובייקט Truck כולו. המחלקה מאשרת את ההתערבות החיצונית הזו ותיתן ל-visitor כפרמטר את עצמו כך שיוכל לטפל בשינוי בצורה חיצונית, ובכך המשאית תוכל להישאר אותה המשאית אך עם פרטי הנהג שונים בכל פעם לפי הצורך.