Code refactoring

רון ירובוי 205591142 דורון שפיצר 313262594

Refactoringהוא תהליך של שיפור מבנה הקוד הקיים מבלי לשנות את התנהגותו החיצונית(ללא שינוי מהותי של הפונ').

המטרה שלו זה לשפר את איכות הקוד ולהקל על התחזוקה והפיתוח העתידי.

התהליך כולל פעולות כמו פישוט קוד, שיפור קריאות, וארגון מחדש של הקוד(שחזור), כך שיהיה קל יותר להבין ולתחזק אותו.

ביצוע ארגון מחדש מאפשר גילוי ותיקון בעיות פוטנציאליות בקלות רבה יותר גם בעת תיקון באגים ושדרוגים, יכול להקל על התחזוקה ולהפחית סיכונים לטעות.

התהליך הזה חשוב במיוחד בפרויקטים גדולים ומורכבים שבהם הקוד נוטה להתבלגן ולעיתים קרובות נדרשת קריאה ותחזוקה מרובה.

לאחר הארגון מחדש הקוד הופך למבנה ברור ומאורגן יותר, מה שמפשט את העבודה השוטפת ומאפשר לבצע שינויים ותוספות בקלות רבה יותר.

ובאופן כללי הוא חיוני לשיפור איכות הקוד ושמירה על סדר וארגון בפרויקטים לטווח הארוך.

**יתרונות מרכזיים:**

**שיפור הקוד** - הקוד הופך לקריא יותר, מובן יותר ומסודר יותר, מה שמקל על המפתחים להבין אותו ולעבוד איתו.

**תחזוקה** - קוד מסודר וברור קל יותר לתחזק ולבצע בו שינויים, מה שמפחית את הסיכון לטעויות.

**איתור באגים** - קל יותר לזהות ולתקן בעיות בקוד כאשר הוא מאורגן ומובן.

**שיפור ביצועים -** במקרים מסוימים, שיפור מבנה הקוד יכול להוביל לשיפור בביצועים הכלליים של המערכת.

**סכרון צוותים –** שכתוב הקוד יכול להפוך את הקוד למובן עבור מפתחים אחרים ולהוביל לעבודה טובה ויעילה יותר בין המפתחים.

**שיטות ליישום:**

**(חילוץ\קיבוץ) Extract Method**

שיטה זו מאפשרת לחלץ קטע קוד מתוך פונקציה ולהעבירו לפונקציה חדשה עם שם ברור ומשמעותי. היא משמשת להורדת רמת המורכבות של הפונקציה המקורית ולהגדלת הקריאות של הקוד.

נשתמש בה על מנת:

לפשט את הפונ' - כאשר פונקציה נעשית מורכבת מדי, ניתן לחלץ ממנה קטע קוד לפונקציה חדשה ובכך לפשט אותה.

להסיר קוד כפול - אם קטע קוד מופיע מספר פעמים במקומות שונים בקוד, ניתן לחלץ אותו לפונקציה חדשה ולהשתמש בה במקום לשכפל את הקוד.

קריאה והבנה עמוקה יותר - כאשר קטעי קוד מועברים לפונקציות עם שמות משמעותיים, הקוד הכולל נעשה ברור יותר ומובן יותר

איך זה עובד בפועל?

ניצור מטודה\פונ' וניתן לה שם מובן לפעולה שהיא עושה.

נעתיק את קטע הקוד הרלוונטי מטודה החדשה שלנו, נמחק את הקטע מהמיקום ונקרא שוב למטודה החדשה.

נראה איזה משתנים רלוונטים לקוד, אם הם מוצהרים ולא נעשה בהם שימוש מחוץ למטודה נשאיר אותם בתור משתנים מקומיים עבור המטודה החדשה.

אם המשתנים מוצהרים לפני הקוד שאנו מחלצים אנחנו צריכים להעבירו אותם(המשתנים) אל המטודה החדשה.(יותר פשוט להשתמש ב Replace Temp with Query שזוהי טכניקה המחליפה משתנים).

A screenshot of a computer program

Description automatically generated

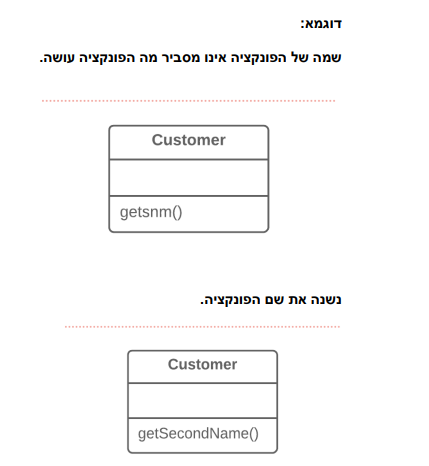
**Rename Methud**

טכניקה זו משתמשת לשינוי שמות ישויות תוכנה קיימות, כגון פונקציות, משתנים, קבצים ועוד, כדי לשפר את הבנה והתחזוקה של הקוד ושיפור פונ'.

איך זה עובד? נבדוק אם המטודה קיימת במחלקה או בתת מחלקה, וניצור מטודה חדש עם שם חדש.

נעתיק אליו את הקוד הישן ונמחק את הקוד מהמטודה הישנה שהקוד ממנה כבר העותק. (כמובן שלא נשכח לשים קריאה למטודה החדשה)

לאחר העתקה נחליף את הקריאות למטודה מהישנה לחדשה ולאחר שבדקנו שהכל הוחלף אפשר למחוק את המטודה הישנה לגמרי.



מקורות:

<https://refactoring.guru/refactoring/techniques>

https://www.techtarget.com/searchapparchitecture/definition/refactoring