

הנדסת תוכנה שיעור 12

עוד על שימוש בגרסאות:

Git revert - חוזר אחורה אבל שומר את השינויים שמהם חזרנו.

ככה נחזור אחורה מגרסאות שונות במקרה שקיבלנו תקלה.

העלים - השלב הכי רחוק בו אין תנאי קדם והוא כבר תקין – זה העלים.

תמיד נשמור גרסה עובדת שממנה אפשר לעשות דברים למקרה שמשהו יתפקשש בדרך ונרצה להראות ללקוח.

נניח ששיש לנו גרסה טובה ללקוח קיים אבל אנחנו רוצים לעשות גרסה חדשה ללקוח אחר, אנחנו בעצם נרצה לשמור את הגרסאות הטובות – זה מה שנעשה בתרגיל היום.

עקרונות הקוד הטוב:

- לא לחזור על עצמנו בקוד – אם אני אשנה במקום אחד אני אולי לא אזכור לשנות במקום שני, כמו כן, יכול להיות שיש לי בעיה בdesign כיוון שאם אותו קוד עושה אותו דבר אין סיבה שיופיע פעמיים.
- עיצוב שיש בו צמידות נמוכה ולכידות גבוהה
- Immutability – לנסות לכתוב מחלקות שלא תלויות בstate
- להשתמש בפונקציות עזר

איך מגיעים לקוד הטוב עפ"י העקרונות ?

- למחוק קודים שחוזרים פעמיים ולצמצם אותם לפונקציה משותפת אחת
- OCP - אנחנו צריכים לעשות את המחלקות שיהיו פתוחות להרחבה אך סגורות לשינויים (כמו ע"י ירושה)
- ISP - ליצור ממשקים ממוקדים
- DIP – להיות תלויים בממשקים אך לא במימוש
- לא לעשות קוד ספגטי

בתרגיל בית:

המטרה: רוצים לעשות גרסה חדשה של קוד קיים עבור לקוח חדש.

הדרך שנעשה את זה: ליצור פרויקט חדש עבור באקליפס ולהעתיק את הקבצים.