מגישים: יוסי לנדה 208705863 , רועי אש 308007749

עבודה #2 יסודות הנדסת תוכנה:

שאלה 1:

בחרנו בבדיקות שלנו בכל המקרים האפשריים שיש לכסות אותם, כלומר מחלקות השקילות שבחרנו מכסות הכל:

- מקרה של Fizz, כלומר מספר שמתחלק ב-3.
- מקרה של Buzz, כלומר מספר שמתחלק ב-5.
- מקרה של FizzBuzz, כלומר מספר שמתחלק ב-3 וב-5 כלומר ב-15.
- מקרה שבו אנחנו מקבלים מספר לא תקין, כלומר לא בטווח התקין של 1 ל-100.

שאלה 2:

בשיטת הטריאנגולציה מטרתנו היא להגיע למקרים בוחנים ככל הניתן עבור הקוד שאנו רוצים לבדוק. ובשיטת מניעת ההכפלות מטרתנו למנוע קוד משוכפל. ולכן כאשר אנחנו נשלב בין שתי השיטות הללו אנחנו בו זמנית נבדוק את הכמות המקסימלית של המקרים שניתן לבחון בכמות המינימלית של קוד משוכפל עבור המקרים שאנו בוחנים ולכן נייצר קוד גנרי מצוין עבור כל הקוד שאנחנו בוחנים.

שאלה 3:

לעומת הסרטון שראינו אנחנו לא כתבנו בדיקות חדשות תוך כדי כתיבת הקוד או ביצענו שינוי כל שהוא לבדיקות בשלב כתיבת הקוד, כל הבדיקות שכתבנו נכתבו לפני שלב כתיבת הקוד. לעומת זאת ההפך מתקיים בסרטון כי רואים תהליך איטרטיבי של כתיבת קוד בעקבות התקלות בבדיקה שאינה עוברת, כלומר ממש כתיבת הקוד מונעת (Driven) מקוד טסט שאינו עובר שזה ממש TDD.

שאלה 4:

בשיטת TDD: בונים טסטים עבור הפונקציונאליות הנדרשת מהתוכנית ואז מתאימים את התוכנית בשלבים כך שתעבור את הטסטים.
שיטת Unit-testing: לעומת זאת היא שיטה לבניית טסטים. בה עבור כל יחידה, עצם או פונקציה קטנה בונים טסט מתאים.