**מטלה 4 :**

**מגישים : אילונה גפטר ועבדאללה מטר**

**הפרויקט נמצא בגיט האב :** [**https://github.com/Ilonagef/FinalProject**](https://github.com/Ilonagef/FinalProject)

**דו"ח על הבדיקות :**

**א.**

2.1.הקריטריון כיסוי הוא לפי משפטים (P1 בהרצאות)

2.2.החריגות שעלולות להתעורר :

א. אם הcount של InputFile וגם של outputFile יהיה שווה ל 0

(כלומר אין TC או TNC בקבצים בכלל) ->תיהיה חריגה

(זריקת EXCEPTION עם הודעה על כך שהקבצים לא מכילים TC ו TNC כלל).

ב. אם במקום ה-i ב list אשר מכיל את קובץ ה input וגם במקום ה-i בlist שמכיל את קובץ ה output , אין את אותו הערך של מילות המפתח TC ו TNC -> תהיה חריגה.

(זריקת EXCEPTION עם הודעה על כך שצריכה להיות התאמה בין מילות המפתח בשני הקבצים)

(לדוגמא : אם ברשימה של ה Input במקום השני מופיע TC אז גם ברשימה של ה output במקום השני צריך להופיע TC .אם לא יהיה את ההתאמה הזו אז תיהיה חריגה)

ג.בשני הlist של הקבצים במקום ה i==0 צריכה להופיע מילת המפתח TC . אם במקום ה i==0 לאחת מהרשימות יש ערך ששונה מTC אז תיזרק חריגה.

ד.אם ה count של קובץ הinput והcount של קובץ ה output לא שווה כאשר ה count של ה Input יהיה שווה ל 0אז תהיה חריגה.

(זריקת EXCEPTION עם הודעה על כך שקובץ ה Input ריק . כלומר לא מכיל TC או TNC )

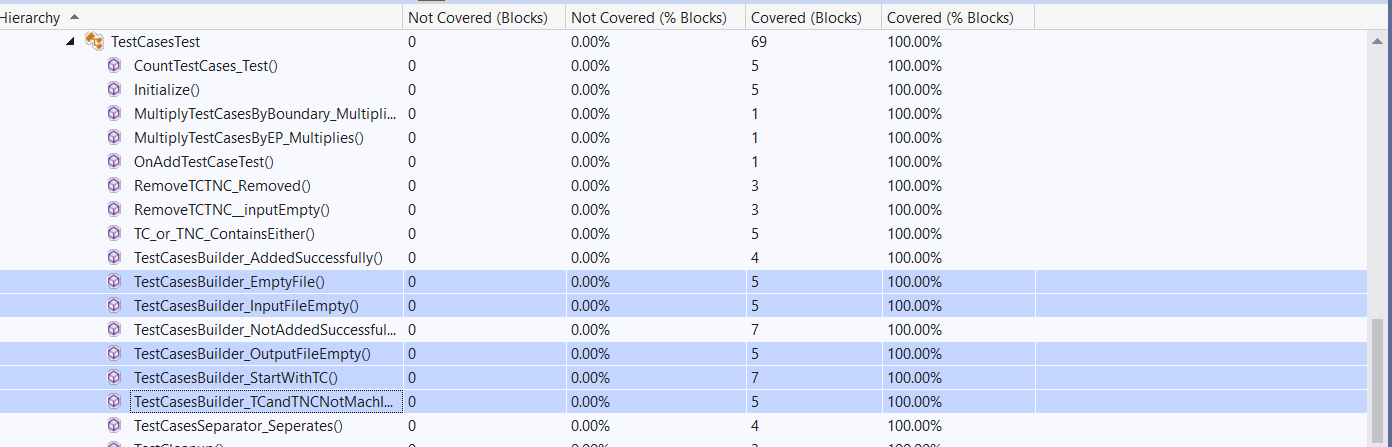
ה. אם ה count של קובץ הinput והcount של קובץ ה output לא שווה כאשר ה count של קובץ ה Output יהיה שווה ל 0 אז תהיה חריגה.

(זריקת EXCEPRION עם הודעה על כך שקובץ ה Output ריק. כלומר לא מכיל TC או TNC ).

החריגות הללו נבדקו בבדיקות יחידה ע"י :

Assert.ThrowsException<Exception>(() =>TestCases.TestCasesBuilder(inputFiletxt, outputFiletxt))

2.3. אחוז כיסוי :

covered in TestCasesTest(in Unitest):

100% הוא TestCasesTest אחוז הכיוסוי של

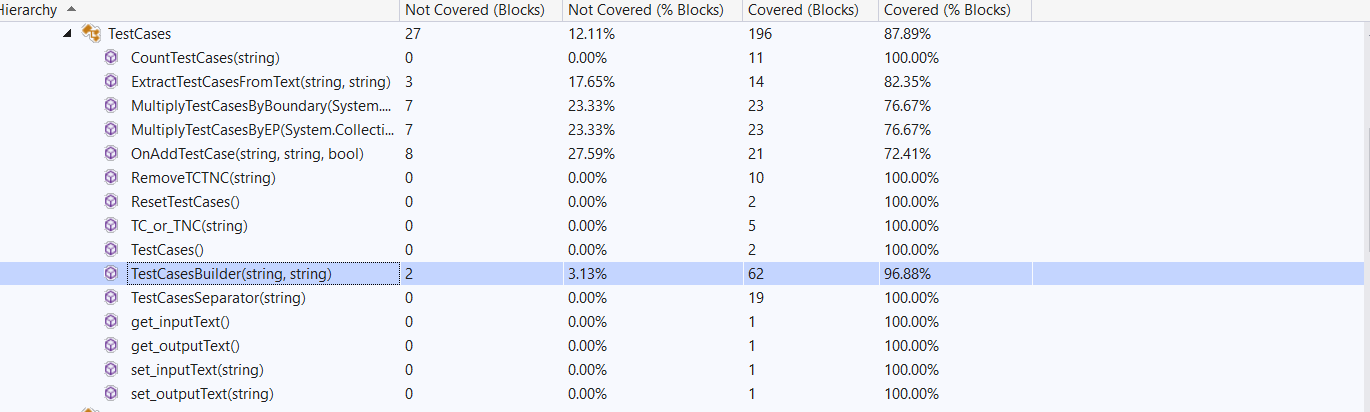
הוספנו חמש בדיקות חדשות למחלקה הזו .

TestCasesBuilderחמשת הבדיקות האלו בודקות את השינויים שביצענו בשיטה

.(TestCasesבמחלקה :)

אחוז הכיסוי של כל אחת מהבדיקות הוא 100% (מוצג בתמונה למעלה . השיטות מסומנות בצבע כחול).

Covered in TestClass (the class that checked in unitTest with TestCasesTest)



(מוצג בתמונה מעל . השיטה הנבדקת מסומנת בכחול)96.88% : TestCasesBuilderאחוז הכיסוי של

2.4. במהלך הבדיקות גילינו על שגיאות של קבלת Path בצורה לא נכונה בתוך הקוד .

כלומר , רצינו לשמור Path של קובץ בתוך משתנה בשביל ליצור את הקובץ בהמשך, אך השתמשנו בשיטות הלא נכונות וכתוצאה מכך קבלנו path לא תקין .

כאשר התוכנית רצתה ליצור את הקובץ , נזרקה שגיאה והתוכנית ישר עצרה .

3.התיקונים שביצענו בקוד הם בשיטה TestCasesBuilder שנמצאת במחלקה TesCases .

שיטה זו מטפלת בקבלת קבצי קלט פלט וכעת לאחר התיקונים ניתן לקבל **רק** קבצי קלט ופלט תקינים , כלומר כתובים בפורמט הנכון : לא ריקים , מתחילים ב \_\_[TC] ויש התאמה בין הקבצים.

בדיקת המערכת המתוקנת :

* בחרנו קובץ Input (ע"י לחיצה על הכפתור Browser ) ריק מתוכן (לא כתוב בו כלום כלל).

בחרנו קובץ output (ע"י לחיצה על כפתור Browse ) ריק מתוכן(לא כתוב בו כלום כלל)

עם בחירת קובץ הoutput קפצה הודעה :

TestCases Number in input and output File is empty!

לאחר לחיצת אישור על ההודעה שקפצה , הנתיבים (textbox ) יתאפסו ועלינו לבחור קבצי קלט ופלט אחרים בשביל להמשיך את הבדיקה .

* בחרנו קובץ Input (ע"י לחיצה על הכפתור Browser ) אשר מתחיל במילת המפתח \_\_[TNC]
* בחרנו קובץ output (ע"י לחיצה על הכפתור Browser ) אשר כתוב כפורמט הנכון .

עם סיום הבחירה קפצה הודעה:

The Test File must stat with \_\_[TC]

לאחר לחיצה על אישור ההודעה שקפצה , הנתיבים (textbox ) יתאפסו ועלינו לבחור קבצי קלט ופלט אחרים בשביל להמשיך את הבדיקה .

- בחרנו קובץ Input (ע"י לחיצה על הכפתור Browser ) וקובץ output

(ע"י לחיצה על הכפתור Browser ) כאשר אין התאמה בין מיליון המפתח במקומות זהים בקבצים.

(לדוגמא בקובץ Input בשורה השלישית רשום \_\_[TC] כאשר בקבוץ הoutput בשורה השלישית רשום \_\_[TNC]. כלומר אין תאמה בין מילות המפתח בקבצים .

ולכן לאחר בחירת הקבצים תקפוץ הודעה :

matching! TC and TNC in input and output file must be

לאחר לחיצה על אישור ההודעה שקפצה, הנתיבים (textbox ) יתאפסו ועלינו לבחור קבצי קלט ופלט אחרים בשביל להמשיך את הבדיקה .

-בחרנו קובץ input (ע"י לחיצה על הכפתור Browser ) שהוא ללא מילות מפתח כלל(אין TC וגם לא TNC )

בחרנו קובץ Output (ע"י לחיצה על הכפתור Browser ) שכתוב בפורמט הנכון .

לאחר בחירת הקבצים תקפוץ הודעה :

The number of TC and TNC in input file is zero!

לאחר לחיצה על אישור ההודעה שקפצה ,הנתיבים (textbox ) יתאפסו ועלינו לבחור קבצי קלט ופלט אחרים בשביל להמשיך את הבדיקה .

את אותה הבדיקה ביצענו עבור קובץ output ריק וקפצה הודעה מתאימה .

בדיקת תקינות לקבצים תקינים :

-בחרנו קובץ Input שרשום בפורמט הנכון (הקובץ לא ריק ומתחיל ב \_\_[TC] ).

-בחרנו קובץ Output שרשום בפורמט הנכון (לא ריק ,בכל מקום שיש בו TC או TNC בקובץ ה input יש TC או TNC בהתאמה בקובץ ה Output ).

הקבצים אכן נבחרו , לא קפצו הודעות והנתיבים (textbox ) של הקבצים התמלאו ב Path שלהם בהתאמה .

ב.

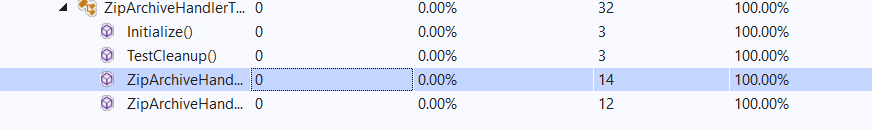
**פונקציונאליות 1**: ניתן לבחור בתיקייה בודדת (לא בפורמט ZIP )

פונקציונאליות זו נכתבה בתוך המחלקה ZipArchiveHandler .

החריגות שעלולות להתעורר זהות לאלו שנכתבו בסעיף א ונבדקות באותה הצורה .

אחוז כיסוי :

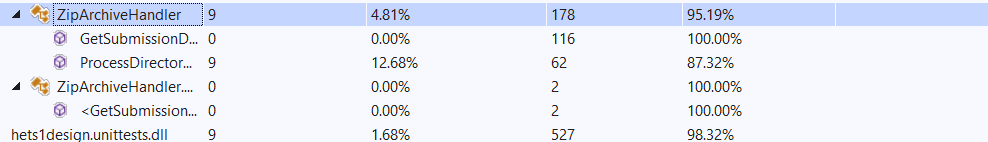
covered in ZipArchiveHsndlerTest(in Unitest)



אחוז הכיסוי של המחלקה אשר בודקת את הפונקציונאליות החדשה הוא : 100%

יצרנו את הבדיקה ZipArchiveHandler\_NotZip אשר תבדוק שהתוכנית יודעת לטפל במצב שבו נבחרה תיקיה אחת לבדיקה (לא מפורמט ZIP ) .

Covered in TestClass (the class that checked in unitTest with TestCasesTest)



אחוז הכיסוי של המחלקה ZipArchiveHandler הוא 95.19%.(במחלקה זו הוספנו את הפונקציונאליות שלנו )

פונקציונאליות 2 : בדיקת תעודות זהות תקינה בתייקית ההגשה של הסטודנט.

בתוך המחלקה : SingleSubmission

covered in SingleSubmissionTest(in Unitest)

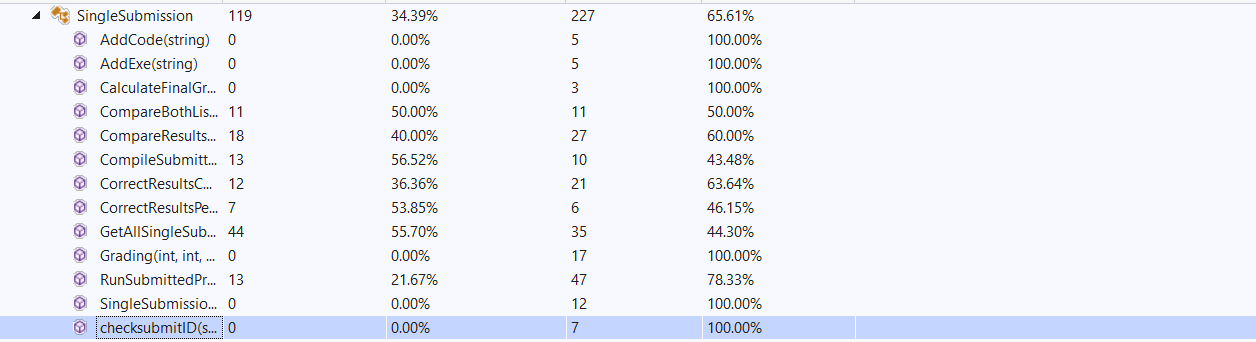
תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

אחוז הכיוסי של מחלקת הבדיקה הוא : 100%

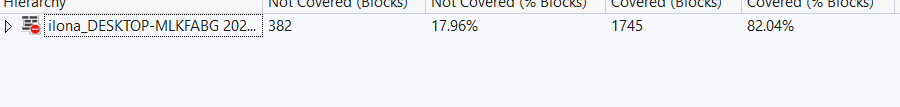
השיטה שמבצעת את הבדיקה היא checksubmitID() . שיטה זו תבדוק אם התעודת זהות (subID) היא חוקית , כלומר בנויה מ 9 ספרות . במידה ואינה חוקית יחזור FALSE.

Covered in SingleSubmission (the class that checked in unitTest with checksubmitID() )



במחלקה זו הוספנו את הפונקציה checksubID() , זאת הפונקציה אשר תבדוק שהתעודת זהות קיימת ותקינה (בתיקיית ההגשה).

אחוז כיסוי הקוד של כל הפרויקט : 82.04%



3. הדרך שבה בדקנו את המערכת המתוקנת והמשופרת:

ע"י לחיצה על הכפתור Browser-Folder -> תיפתח תיקיית הקבצים -> נבחר תיקיה לבדיקה (לא ZIP).

בחרנו בתיקיה : D:\שנה ג\סמסטר ב\אימות ובדיקת תוכנה\project2\_1\Project Code\HETS - Azo\Assets\New Test\_ilonaAbdala\OpenFolderTest\אחמד סכראן\_709245678\_assignsubmission\_file\_

לאחר מכן הכנסנו קובץ input ו וקובץ output (נוכל להכניס רק קבצי קלט ופלט שהם תקינים וכתובים בפורמט הנכון כפי שהסברנו בסעיף א – 3 ).

כעט ניתן לבצע קומפילציה ע"י הכפתור : compaile -> הרצה ע"י הכפתור : Run Programs .

ניתן לצפות בתוצאות הבדיקה ע"י Results ובנוסף ניתן ליצור קובץ של הבדיקה עם התוצאות ע"י DetailedResults .

בבדיקה זו בדקנו את המערכת עם הפונקציונאליות החדשה : בחירת תיקיה בודדת לבדיקה וגם בדיקת תעודות זהות בהגשה.

\*ע"י לחיצה על הכפתור Browser-Folder נוכל לבחור אך ורק תיקיה בודדת שבתוכה יש עבודת הגשה .

לא נוכל לבחור תיקיה שמכילה קבצי .txt או קבצי .ZIP

\*במידה והתעודת זהות אינה קיימת או אינה תקינה: השדה של ה ID יהיה עם שם העבודה במקום

(בקובץ של תוצאות הבדיקה Detailed Results HETS - Azo) .

ג. בדיקות קבלה אוטומטיות :

\*בשביל לייצר בדיקות קבלה אוטמטיות יצרנו פונקציות נוספרות במסך ה MAINSCREEN עפ"י ההסבר שקבלנו במודול .

פונקציות אלו מופעלות מתוך התוכנית CSHCONSOLEAPI ע"י כך שהגדרנו פקודות של בדיקות שיש לבצע בתוך המחלקה AppCommands .

הגדרנו את שם הפקודה ואת הפונקציה שלה בתוך AppCommands , הוספנו את הפקודה להיות חלק מהפקודות ע"י CommandsApi.CommandAdd(…….) מתוך המחלקה CONSOLEAPLICATION .

הפקודה שנכתבת במסך ה CMD מופעלת מתוך המחלקה AppCommands והיא תקרא לפונקציות המתאימות במחלקה MAINSCREEN וכך יתבצעו בדיקות הקבלה באופן אוטומטי .

\*חריגות שעלולות להתעורר :

-במידה ולא נכניס קובץ ZIP וננסה לבצע קומפילציה (הפקןדה : COMPILE())תיהיה חריגה , זריקת EXCEPTION והודעה על כך שצריך לבחור נתיב לקובץ ZIP.

-במידה ולא נכניס קובץ קלט(הפקודה INPUT()) וננסה לבצע קומפילציה (הפקודה COMPILE()) , זריקת EXCEPRION והודעה על כך שצריך לבחור נתיב לקובץ קלט.

אותו דבר לגבי קובץ OUTPUT .

-במידה וננסה להזין פקודה שלא הוגדרה בקוד , תיהיה זריקת EXCEPTION והודעה על כך שהפקודה לא מוכרת . התוכנית תעצר במצב כזה.

פקודות ל CMD אשר מפעילות את בדיקות הקבלה באופן אוטומטי:

gui() : פתיחת חלון האפליקציה

zip()-מכניס באופן אוטומטי את הנתיב של קובץ הארכיב

folder()-מכניס באופן אוטומטי את הנתיב של התיקיה הבודדת לבדיקה

Input()-מכניס באופן אוטומטי את הנתיב לקובץ הקלט.

Output()-מכניס באופן אוטומטי את הנתיב לקובץ הפלט.

Compile()-מבצע קומפילציה באופן אוטומטי .

Run()-מבצע הרצה באופן אוטומטי.

Results()-מציג את התוצאות של הבדיקה בחלון האפליקציה.

fileResults()-מייצר תיקיה אשר תכיל את קובץ תוצאות הבדיקה.

bit32()-בוחר את סוג הקומפיילר להיות 32 ביט. כתוצאה מביצוע הפקודה הנ"ל כפתור הרדיו בחלון האליקציה יבחר.במידה וכפתור הרדיו הזה לא יבחר אז באופן אוטומטי יבחר כפתור הרדיו של 64 ביט.

onlyCode()-מבצע בדיקה של קובץ הקוד בלבד (כאשר חסר קובץ EXC או במידה ולא נרצה לבצע עליו בדיקה ). במצב כזה תתאפשר קומפילציה .

onlyExec()- מבצע בדיקה של הקובץ EXC (כאשר חסר קובץ קוד או במידה ולא נרצה לבצע עליו בדיקה).במצב כזה לא תתאפשר קומפילצירה ולכן כפתור ה COMPILE לא יהיה מאופשר (נוכל לבצע רק הרצה).

- ExecAndCode()מבצע בדיקה על קבצי הוד ועל קבצי ה EXC . לאחר מכן ניתן לבצע קומפילציה .

Close() – סוגר את חלון האפליקציה .