**הסבר על הפרויקט**

בפרויקט התייחסתי למושג file stream כאל קובץ אשר נקרא שורה ושורה ועבור כל שורה מבצעת את הפעולה הדרושה On the fly, מכיוון שבעולם ה online לא ניתן להעריך מה הקלט הצפוי הבא לכן לא טענתי מראש את כל השורות אלא עבור כל שורה ביצעתי את הפעולה הדרושה ואז עברתי לשורה הבאה.

בפרויקט ישנם מספר interfaces.

1. Parser מכיל את הלוגיקה של הפרסור של הקלט
2. Model מכיל את הלוגיקה של מה לעשות עם הקלט המפורסר
3. Service הקלאס שממש אותו אמור ל"נהל" את כל הפעולה הדרושות על מנת להחזיר את התשובה.

הקלאס DAU Controller מכיל 3 מתודות חשובות

1. Process, שבעצם מניעה את כל התהליך
2. getAll, מחזירה כל ה Active Users שהיו עבור כל תאריך
3. getByDate , מחזיר את ה Active User עבור תאריך מסוים

ה Parser ו Model לא מכירים אחד את השני, והם יחידות עצמאיות, בנוסף לכך ה Service מכיר את שניהם, אבל ה DAU Controller מכיר רק את ה Service, כך נוצרת הפרדה מלאה בין כל החלקים המשתתפים בעיבוד המידע.

השתמשתי ב Generics עבור כל אחד כל ה interfaces וגם ל DAU controller , ככה אפשר לשלוט בדיוק באיזה מידע נרצה לשלוח לכל אחד מהחלקים.

לדוגמה ב Parser אני יכול לקבל את זה כסטרינג המייצג שם של קובץ ובפעולה אחרת לקבל אובייקט מסוים, כך גם לגבי ה Model שניתן לייצר קלאס המממש אותו ולהחליט איך נרצה לשמור את המידע ולפי מה להחזיר אותו ל service

כנ"ל לגבי ה service וגם ה DAU נוכל להגדיר לו איך נרצה להחזיר בעצם את המידע שהוא מקבל בסוף מה service, כל האובייקט מוזרקים בזמן ריצה כך שהפרויקט מאוד גמיש ויכול לקבל כל לוגיקה שנרצה מכיוון שהוא מופרד.

במקרה של הטקסט (של המשימה הנוכחית ) אני מחזיק בהאש מאפ שה key הוא התאריך וה value הוא HashSet של UserId

הוספתי Junit בתיקייה של Test שמראות את בדיקה עבור כל התאריכים