מעבדה 9. נושא: טבלאות גיבוב

תאריך הגשה: 18/07/2024 בשעה 23:00 (בזוגות)

יש לקרוא היטב לפני תחילת העבודה !

מבוא:

במעבדה הנוכחית נממש טבלת גיבוב.

תיאור:

טבלת גיבוב היא מבנה נתונים שנותן גישה לרשומה באמצעות ה**מפתח** המתאים לה. המבנה עובד באמצעות הפיכת המפתח על ידי ***פונקציית גיבוב***, למספר המייצג אינדקס במערך שמפנה אל הרשומה המבוקשת.

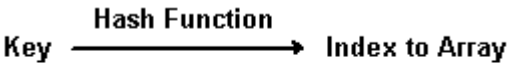
למשל: במפעל של 100 איש הוחלט לתת לכל עובד מספר ייחודי בן 3 ספרות. באופן כזה ניתן לשמור את כל העובדים במערך של 1000 איברים.

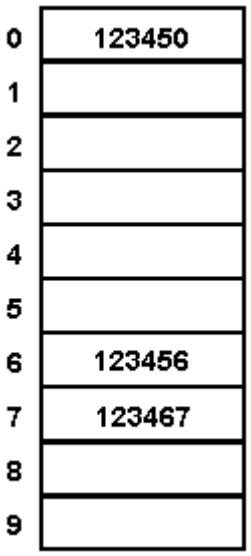
אולם, אם משתמשים בת.ז. של העובדים בני 9 ספרות, נצטרך מערך בגודל מיליארד (10^9)
כדי להחזיק 100 עובדים בסה"כ.
ולכן רצוי להשתמש במבנה נתונים שונה ממערך בו עדיין אפשר לבצע חיפוש מהיר.

יתרונות:

חיפוש מהיר: בהינתן מפתח נתון (למשל שם של אדם), מצא את הרשומה המתאימה (למשל מספר הטלפון של אותו אדם).

פונקצית גיבוב:





יש הרבה סוגים של פונקציות גיבוב.

דוגמא 1 לפונקציית גיבוב כאשר המפתח הוא מספר שלם:

נתונים מספרים שלמים. יש למפות אותם למערך בגודל N

F(key) = key%N

למשל, עבור מספרים: 123456, 123467, 123450

ומערך בגודל N=10

נבצע:

123456%10=6

123467%10=7

123450%10=0

דוגמא 2 לפונקציית גיבוב כאשר המפתח הוא אינו מספר שלם:

למשל, אם נתונים השמות:

Sarah Jones
Tony Balognie
Tom Katz

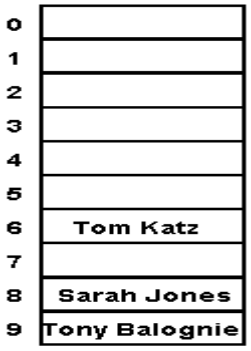
בשלב ראשון, נעביר את השמות למספרים שלמים ע"י פונקציית מעבר ממחרוזת למספר שלם. דוגמא לפונקציה כזאת יכולה להיות, אם נצמיד לכל תו את קוד הASCII שלו **ונחבר** את כל הערכים יחד. התוצאה תהיה:

Sarah Jones --> 1038
Tony Balognie --> 1259
Tom Katz --> 746

בשלב שני, נשתמש בפונקציית הגיבוב מהדוגמא הקודמת :

Sarah Jones --> 1038 % 10 --> 8
Tony Balognie --> 1259 % 10 --> 9
Tom Katz --> 746 % 10 --> 6

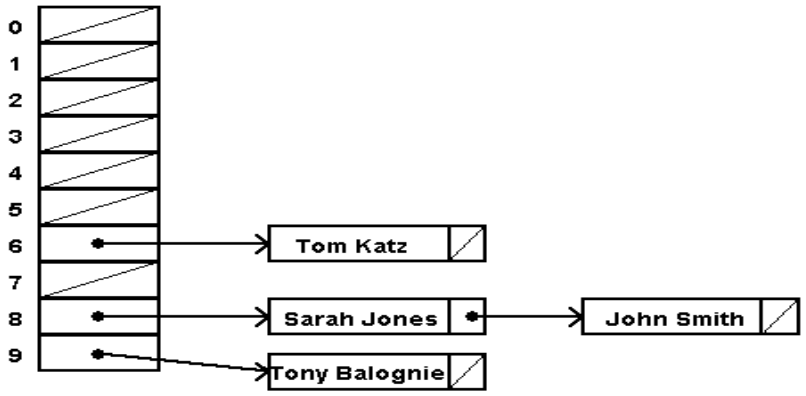
התוצאה תהיה:



טיפול בהתנגשויות:

לעתים, יתכן כי שני מפתחות שונים יופנו לאותו התא במערך.
למשל, אם נוסיף לדוגמא 2 : John Smith --> 948 % 10 --> 8 . נקבל כי תא מספר 8 כבר תפוס.

במעבדה הנוכחית, נטפל בגישת השרשור(Chaining) לטיפול בהתנגשות לפיה, במקרה של התנגשות, המפתחות ישורשרו ברשימה מקושרת. למשל, עבור דוגמא 2 שלעיל,



משימה:

צרו מחלקה המגדירה **טבלת גיבוב** (HashTable) המכילה את השיטות contains, add, remove, clear, isEmpty.

תיעוד המחלקה נתון לכם.

מחלקה זו דומה במעט למחלקה HashSet הממשת את Set:

https://docs.oracle.com/javase/8/docs/api/java/util/Set.html

הדרך התקנית להפוך מפתח כלשהו למספר שלם הוא להשתמש בשיטה hashCode. ניתן ליצור פונקציה כזאת באופן אוטומטי ב ECLIPSE.

הדרך התקנית לבדוק שוויון של עצמים היא להשתמש בשיטה equals. ניתן ליצור פונקציה כזאת באופן אוטומטי ב ECLIPSE.

ניתן להשתמש בפתרון של מעבדות קודמות, אבל לא בספריה java.util.

המשימה (איך לא?) היא לגרום למחלקת הבדיקה HashTableTest לעבוד בצורה תקינה.

1. הוסיפו למחלקה Person מתודה equals המשווה בין שלושת השדות של Person.
   הוסיפו מתודת hashCode מתאימה. שימו לב שהשוואה זו שונה ממה שהיה במעבדה של עצי חיפוש.
   המחלקה Person נמצאת בחבילה: il.ac.telhai.ds.misc package;
2. ממשו את HashTable הנתונה, תוך שימוש במחלקה DLinkedList שמימשתם במעבדה 3.
   על המחלקה DLinkedList והממשק List להיות בחבילה il.ac.telhai.ds.linkedlist.
   המחלקה HashTable נמצאת בחבילה il.ac.telhai.ds.hash. יש להשלימה.
   יתכן שבבנאי תקבלו "הערת אזהרה". כדי שהקומפיילר לא יתייחס לאזהרה זו, ניתן לכתוב מעליה:
   @SuppressWarnings("unchecked")
   אם עדיין יש אזהרה בבודק האוטומטי - ניתן להחליף שורה זו בשורה הבאה:
   @SuppressWarnings({"unchecked","rawtypes"})

סדר העבודה ופרטים טכניים

שליפת הפרויקט DS-Lab09-HashTable מתוך GITHUB בקישור:

https://github.com/michalHorovitz/DSLab2024Public

- אם אין לכם גישה לפרויקט שהורדתם מ GITHUB במעבדה הראשונה יש לבצע שליפה מחדש לפי ההוראות במעבדה הראשונה.
- אם יש לכם גישה לפרויקט שהורדתם מ GITHUB במעבדה הראשונה אז בצעו:
  - קליק על שם הפרויקט.
  - עכבר ימני
  - Team-->Pull
  - File-->Import->Git->Projects From Git->Existing Local Repository

אם אתם עובדים ב VDI, מומלץ לשנות את המיקום המוצע לפרויקט בתיקייה כלשהי בכונן H.

<u>פורמט קובץ ההגשה ובדיקתו:</u>

פורמט: יש להגיש קובץ ZIP בשם

51_lab09_123456789_987654321.zip

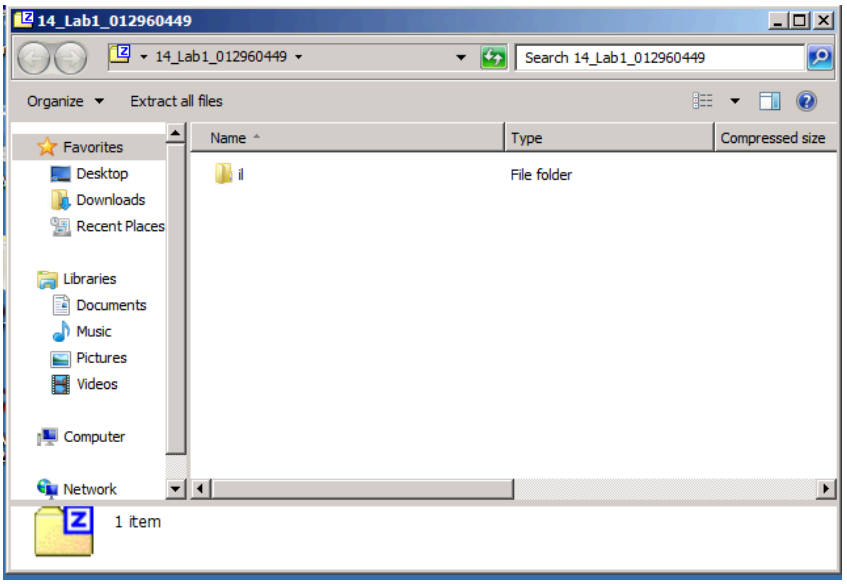
(כמובן, יש להחליף את המספרים עם מספרי ת.ז. של המגישים).

על הקובץ להכיל את כל קבצי ה JAVA שכתבתם כאשר הם נמצאים בתתי תיקיות בתוך התיקייה il/ac/telhai/ds/

על פי המבנה של הפרוייקט הנתון.

כלומר, השורש של קובץ ההגשה יכיל רק תיקייה בשם il, והוא יכיל את כל קבצי - java על פי התבנית הנתונה בפרוייקט.

להמחשה תמונה של קובץ כזה שנפתח ב - WindowsExplorer



<u>בדיקת קובץ ההגשה:</u> בדקו את הקובץ שיצרתם בתוכנת הבדיקה בקישור:

https://csweb.telhai.ac.il/

ראו סרטון הדגמה של השימוש בתוכנת הבדיקה.

חשוב !!!

בדיקת ההגשות תבוצע ברובה ע"י תוכנית הבדיקה האוטומטית הנ"ל. תוצאת הבדיקה תהייה בעיקרון זהה לתוצאת הבדיקה הנ"ל שאתם אמורים לערוך בעצמכם . כלומר, אם ביצעתם את הבדיקה באתר החוג, לא תקבלו הפתעות בדיעבד. אחרת, ייתכן שתרגיל שעבדתם עליו קשה ייפסל בגלל פורמט הגשה שגוי וכו'. דבר שהיה ניתן לתקנו בקלות אם הייתם מבצעים את הבדיקה. היות ואין הפתעות בדיעבד, לא תינתן אפשרות של תיקונים, הגשות חוזרות וכד'.

הגשה שלא מגיעה לשלב הקומפילציה תקבל ציון 0.

הגשה שלא מתקמפלת תקבל ציון נמוך מ- 40 לפי סוג הבעיה.

הגשה שמתקמפלת תקבל ציון 40 ומעלה בהתאם לתוצאות הריצה, ותוצאת הבדיקה הידנית של הקוד (חוץ ממקרה של העתקה).

תכנית הבדיקה האוטומטית מכילה תוכנה חכמה המגלה העתקות. מקרים של העתקות יטופלו בחומרה