

קורס מבוא לתכנות מערכות 10010 – דו"ח פרויקט חברת קוקה קולה

הנדון : דו"ח פרויקט – חברת קוקה קולה
מחלקה : הנדסת תוכנה

פרטי המגישים :

- עידן נוישול 207813635
- ספיר גילני 322358284

1. תוכן עניינים

| | | |
|------|--|----|
| 1. | תוכן עניינים | 2 |
| 2. | הקדמה | 3 |
| 3. | תיאור מבנה המערכת | 3 |
| 3.1 | מבנה Bottle | 3 |
| 3.2 | מבנה BottlePacking | 3 |
| 3.3 | מבנה Truck | 4 |
| 3.4 | מבנה Employee | 4 |
| 3.5 | מבנה Supplier | 4 |
| 3.6 | מבנה HistoricalEvent | 5 |
| 3.7 | מבנה CocaColaTour | 5 |
| 3.8 | מבנה CocaColaFactory | 6 |
| 3.9 | מבנה Address | 7 |
| 3.10 | מבנה DateTime | 8 |
| 4. | פעולות יצירתיות | 8 |
| 4.1 | הצפנה בעזרת ביטים | 8 |
| 4.2 | עבודה עם SEEK ו-SEED | 9 |
| 5. | GeneralArray | 10 |
| 6. | Macros | 10 |
| 7. | פולימורפיזם | 10 |
| 7.1 | מבנה Employee | 10 |
| 7.2 | מבנה EmployeeGuide | 11 |
| 7.3 | מבנה EmployeeDriver | 12 |
| 8. | אופן דחיסת המבנים לקובץ בינארי | 12 |
| 8.1 | פורמט כתיבת חברת קוקה קולה לקובץ Coca_Cola_Factory | 12 |
| 8.2 | פורמט כתיבת חברת קוקה קולה לקובץ Historical_Events | 14 |
| 8.3 | אופן דחיסת בקבוק | 14 |
| 9. | פעולות הניתנות לביצוע ע"י המשתמש | 15 |
| 9.1 | התפריט הראשוני | 15 |
| 9.2 | תפריט מרכזי | 15 |
| 10. | שרטוט המערכת | 16 |
| 10.1 | שינויים מהשרטוט הראשוני | 16 |
| 10.2 | שרטוט – פירוט הפונקציות הראשיות של המערכת | 16 |
| 10.3 | שרטוט – Class Diagram | 16 |
| 11. | חלוקת אחריות | 16 |

2. הקדמה

- אנו בחרנו לפתח מערכת המדמה את מפעל קוקה קולה.
המערכת מאפשרת פונקציונליות המדמה ארגון ושליטה של מפעל קוקה קולה.
1. משאבי אנוש – ניהול עובדי המפעל (מדריכים ונהגים).
 2. שינוע – ניהול צי המשאיות והספקים של המפעל.
 3. תיירות – ניהול הסיורים המתבצעים במפעל.
 4. סחורה – סוגי הסחורה המיוצרים במפעל.

3. תיאור מבנה המערכת

המבנים הקיימים במערכת הינם:

3.1 מבנה Bottle

מבנה זה מדמה את המוצר אותו המפעל מייצר.

תכונות המבנה

- תאריך תוקף
- טעם – ENUM (קלאסי, זירו, לימון, ללא קפאין).
- סוג הבקבוק – ENUM (פלסטיק, זכוכית, פחית)
- סוכר – מקסימום 127 גרם.
- קלוריות – מקסימום 1,023.
- מ"ל – מקסימום 2,000.

פונקציות מרכזיות

1. אתחול – יצירת המבנה Bottle.
2. הדפסה – הדפסת פרטי Bottle.
3. כתיבה לקובץ בינארי/טקסט.
4. קריאה מקובץ בינארי/טקסט.

הערות

לא נכתבה פונקציית free משום שתכונותיו לא דרשו הקצאת זיכרון דינאמי.

3.2 מבנה BottlePacking

מבנה זה מדמה את האריזות הקיימות בכל משאית, מפעל קוקה קולה מייצר אריזות ולא בקבוקים בודדים. בכל אריזה קיים סוג בקבוק אחד.

תכונות המבנה

- בקבוק – יכול מבנה Bottle.
- כמות - ייצג את כמות הבקבוקים באריזה.

פונקציות מרכזיות

1. אתחול – יצירת המבנה BottlePacking.
2. הדפסה – הדפסת פרטי BottlePacking.
3. כתיבה לקובץ בינארי/טקסט.
4. קריאה מקובץ בינארי/טקסט.

3.3 מבנה Truck

מבנה זה מייצג משאית של מפעל הקוקה קולה.

תכונות המבנה

- הצבעה לעובד – מדובר בפולימורפיזם (יפורט בהמשך) ועל כן הכוונה בפועל תהיה לעובד מסוג נהג.
- הצבעה לספק.
- מערך בגודל קבוע של אריזות בקבוקים - כל משאית יכולה להעמיס עד 10 אריזות.
- כמות אריזות בפועל.
- לוחית רישוי למשאית - 7 תווים.

פונקציות מרכזיות

1. אתחול – יצירת המבנה Truck
2. הדפסה – הדפסת פרטי Truck (ישנן שתי פונקציות הדפסה, האחת פרטיו היבשים של המשאית והשנייה תכלול את פרטי הבקבוקים אשר היא משנעת).
3. כתיבה לקובץ בינארי/טקסט.
4. קריאה מקובץ בינארי/טקסט.
5. הוספת אריזה – יקבל כפרמטר משאית ויצור אריזה חדשה על סמך הזנת המשתמש ויוסיף אותה למשאית.

הערות

1. המבנה משאית אינו יוצר עובד/ ספק אלא מבצע הצבעה לעובד/ספק הרלוונטיים מתוך מאגר העובדים והספקים הנמצא במפעל.
2. לא ניתן לאתחל משאית במידה ולא קיימים נהגים וספקים.

3.4 מבנה Employee

מבנה זה בנינו באופן המקיים פולימורפיזם והורשה.
יוסבר עליו באופן נרחב בסעיף פולימורפיזם.

3.5 מבנה Supplier

מבנה זה מייצג ספק למשאית.

תכונות המבנה

1. מספר סריאלי – מספר ייחודי, ללא כפל.
2. שם
3. מספר טלפון – גודל קבוע מראש.
4. כתובת – מבנה Address.

פונקציות מרכזיות

1. אתחול – יצירת המבנה Supplier.
2. הדפסה – שתי פונקציות הדפסה, האחת מקבלת מצביע של Supplier, השנייה מקבלת void*.
3. כתיבה לקובץ בינארי/טקסט.
4. קריאה מקובץ בינארי/טקסט.
5. שחרור – שתי פונקציות שחרור, האחת מקבלת מצביע של Supplier, השנייה מקבלת void*.

6. פונקציית `getSupplierId` – פונקציה זו מקבלת `Supplier` כמצביע ובנוסף מערך של מצביעים של ספקים, הפונקציה מאתחלת מהמשתמש מספר סריאלי ייחודי לספק.

הערות

1. נדגיש כי מספר סידורי של ספק ושל עובד עשויים להיות זהים שכן מבחינת המפעל מדובר בישויות שונות.

3.6 מבנה `HistoricalEvent`

חברת קוקה קולה הוקמה בשנת 1886, החברה עברה אירועים היסטוריים ושינויים רבים במהלך שנותיה.

כיום חברת קוקה קולה מקיימת סיורים במפעליה בכלל רחבי העולם.

לאור זאת החלטנו לתכנת במערכת את היכולת לבצע סיורים, כל סיור מציג מספר אירועים היסטוריים של חברת קוקה קולה.

תכונות המבנה

- תאריך האירוע – מבנה `DateTime`.
- תיאור האירוע.

פונקציות מרכזיות

1. אתחול – יצירת המבנה `HistoricalEvent`.
2. הדפסה – הדפסת פרטי `HistoricalEvent`.
3. כתיבה לקובץ בינארי/טקסט.
4. קריאה מקובץ בינארי/טקסט.
5. שחרור – שחרור המבנה מקבלת `void*`.
6. פונקציית השוואה `compare` לפי תאריך.
7. פונקציית השוואה `compare` לפי תיאור האירוע.

הערות

1. האירועים ההיסטוריים ישמרו תחת קובץ נפרד בשם `Historical_Events.bin/txt`.
2. שנת המינימום לאירוע היסטורי הינה 1886.

3.7 מבנה `CocaColaTour`

מבנה זה מייצג סיור במפעל קוקה קולה.

תכונות המבנה

- תאריך – מבנה `DateTime`
- אירועים היסטוריים – שמור כ `LIST` (ההזנה תתבצע באופן ממוין על פי התאריך).
- משך הסיור
- מדריך – מצביע ל `Employee`, מדובר בחורשה (יפורט בהמשך).
- מספר משתתפים בסיור.

פונקציות מרכזיות

1. אתחול – יצירת המבנה `CocaColaTour`.

2. הדפסה – הדפסת פרטי CocaColaTour באופן יבש (ללא האירועים ההיסטוריים המשוויכים לסיור).
3. כתיבה לקובץ בינארי/טקסט.
4. קריאה מקובץ בינארי/טקסט.
5. שחרור – שתי פונקציות שחרור, האחת מקבלת מצביע של CocaColaTour, השנייה מקבלת void*.
6. פונקציה מיוחדת – `GetEventFromFileBySeek` יפורט תחת הסעיף פונקציות מיוחדות.
7. הוספת אירוע היסטורי לסיור באופן רנדומלי, פונקציה זו נעזרת בפונקציה המיוחדת לעיל, ומוסיפה את האירוע.
8. מילוי אירועים היסטוריים בשני אופנים שונים – דרך קובץ בינארי/דרך האירועים השמורים במפעל כ LIST.
9. תחילת סיור – הדפסת פרטי הסיור והדפסת האירועים ההיסטוריים אשר המדריך מסביר עליהם.
10. פונקציית השוואה compare לפי משך הסיור.
11. פונקציית השוואה compare לפי מדריך.
12. פונקציית השוואה compare לפי תאריך הסיור.

הערות

1. הזנת האירועים ההיסטוריים לסיור מוזנת באופן אוטומטי ולא על ידי המשתמש.
 2. כמות האירועים המוצגת בסיור תלויה במשך זמן הסיור שהמשתמש בחר.
- כלומר, משך הסיור המקסימלי שהמשתמש יכול לקבוע יהיה מכפלת מספר האירועים הקיימים במפעל כפול משך ההסברה לאירוע בודד (כל האירועים בעלי אותו משך הסברה)

$$MaxDuration = numOfEvents \cdot EVENT_TIME$$

$$MinDuration = EVENT_TIME$$

3.8 מבנה CocaColaFactory

זהו המבנה המרכזי בפרויקט, מבנה זה מתאר את הישות הראשית של המערכת.

מבנה זה כולל את ניהול המפעל.

תכונות המבנה

- מערך מצביעים לעובדים.
- כמות בפועל של עובדי המפעל.
- מערך של משאיות.
- כמות בפועל של משאיות.
- מערך של מצביעים לספקים.
- כמות בפועל של ספקים.
- מערך מצביעים של סיורים.
- כמות בפועל של סיורים.
- רשימה מקושרת של כל האירועים ההיסטוריים של חברת קוקה קולה.

- כמות האירועים.
- משתנה מסוג ENUM המייצג את אופי מיון הסיורים, אי מיון מוגדר כ 1-.
- ערך seed השומר את אופי העלאה הרנדומלית של אירועים לסיורים (נשמר בקבצים הבינאריים לשחזור מקורי של נתוני המערכת – יפורט בהמשך).

פונקציות מרכזיות

1. אתחול – יצירת המבנה CocaColaFactory.
2. הדפסה – הדפסת פרטי CocaColaFactory.
3. כתיבה לקובץ בינארי/טקסט.
4. קריאה מקובץ בינארי/טקסט.
5. הוספת עובד – מוסיף עובד למפעל, יפורט בסעיף פולימורפיזם בהמשך.
6. הוספת אירוע היסטורי.
7. הוספת ספק.
8. הדפסת פרטי משאית – הדפסת פרטי המשאית הנבחרת על ידי המשתמש.
9. בחירת ספק – החזרת מצביע של ספק על פי בחירת המשתמש.
10. מציאת ספק לפי id - החזרת מצביע של ספק.
11. מציאת עובד לפי id – החזרת מצביע של עובד.
12. הוספת סיור למפעל.
13. בחירת סיור להפעלתו.
14. מיון הסיורים של המפעל – בעזרת qsort.
15. חיפוש סיור – בעזרת bsearch.
16. שחרור המפעל.

הערות

נתוני המפעל נשמרים בשני קבצים Coca_Cola_Factory.bin/txt.

קובץ אחד לאירועים היסטוריים בלבד, קובץ שני לכל שאר הנתונים.

הנתונים נשמרים הן לקבצים בינאריים והן לקבצי טקסט.

3.9 מבנה Address

מבנה זה מייצג כתובת.

תכונות המבנה

- עיר
- רחוב
- מספר בית.

פונקציות מרכזיות

1. אתחול – יצירת המבנה Address.
2. הדפסה – הדפסת פרטי Address.
3. כתיבה לקובץ בינארי/טקסט.
4. קריאה מקובץ בינארי/טקסט.
5. שחרור

3.10 מבנה DateTime

- מבנה זה מייצג תאריך ושעה, מורכב משני מבנים.
- מבנה Time מכיל שעות ודקות.
- מבנה Date מכיל יום, חודש, שנה.

פונקציות מרכזיות

1. אתחול – יצירת המבנה DateTime.
2. הדפסה – הדפסת פרטי DateTime
3. פונקציה קבלת תאריך – מקבלת תאריכים על פי פורמט ספציפי.
4. כתיבה לקובץ בינארי/טקסט.
5. קריאה מקובץ בינארי/טקסט.
6. שחרור
7. פונקציית השוואה לפי DateTime

4. פעולות יצירתיות

לאורך הפרויקט אנו פיתחנו 2 פעולות יצירתיות.

4.1 הצפנה בעזרת ביטים

אחד הנושאים המדוברים ביותר בקוקה קולה הינו ה"המרכיב הסודי" של הקולה, תגלית הקולה התרחשה בשנת 1886. משנה זו ועד היום מרכיבה הסודי של הקולה נשמר באופן כמס.

לאור זאת החלטנו שאחת מהפעולות היצירתיות אשר נפתח תהא בהקשר לנושא זה. תחת הקובץ CocaColaFunctions.h ניתן למצוא את הפונקציה :
findSecretRecipe()

מטרת הפונקציה

לפונקציה יש מערך של תווים, כל תו מיוצג כהקסא דצימלי, כאשר מנסים להדפיס את התווים מקבלים אוסף של תווי ASCII לא מוכרים.
מערך זה כביכול "מוצפן" ופיענוחו ידפיס את המרכיב הסודי לקוקה קולה.

אופן הפעולה

הפונקציה מציגה למשתמש את ההדפסה של המערך המוצפן, על המשתמש לבחור באחד מבין חמשת המפתחות המוצגים לו, המפתחות הינם :

- מפתח צהוב
- מפתח אדום
- מפתח כחול
- מפתח ורוד
- מפתח ירוק

כל מפתח מבצע פונקציה שונה בעבודה על הביטים של התווים.

האחד מבצע XOR למול ערך מסוים, האחד OR לערך אחר וכו'.

לאחר כל בחירת מפתח יודפס למשתמש הערך החדש, רק מפתח אחד הינו המפתח הנכון אשר יפענח את המרכיב הסודי של הקולה.

המפתח הנכון הינו המפתח הוורוד.

4.2 עבודה עם SEEK ו-SEED

הפונקציה המיוחדת השנייה אשר בחרנו פועלת עם נושאים נוספים, SEEK ו-SEED.

בסעיף המבנים הסברנו כי כל האירועים ההיסטוריים (Historical Event) נשמרים בקובץ בינארי נפרד, הסיבה לכך הינה הפעולה המיוחדת.

שם הפונקציה המיוחדת: `GetEventFromFileBySeek` תחת הקובץ `CocaColaTour.h`.

אופן הפעולה

כאמור בהוספת סיור חדש, המשתמש מזין את משך זמן הסיור ולפיו נקבע מספר האירועים שיוספרו בסיור. האירועים מוגרלים באופן רנדומלי בלבד, ללא צורך בבחירת המשתמש, על פי פונקציית הזמן של המחשב.

במהלך השמירה של הנתונים לקובץ, כלל האירועים ההיסטוריים של קוקה קולה נשמרים בקובץ נפרד ללא קשר לנתוני המפעל. בקובץ המפעל, נשמר ערך ה-`SEED` שאיתו הוגרלו האירועים עבור הסיורים, ולעומת זאת האירועים שנבחרו לסיור לא יישמרו בשנית, אלא יוגרלו מחדש במהלך הריצה הבאה שתבצע בהעלאה הנתונים מהקובץ.

פעולה זאת מאפשרת מספר דברים:

- שלא יישמרו כפילויות של נתונים בקבצים.
- הגרלה מחודשת לסיורים נוספים.
- בשחזור הנתונים מהקבצים, ייקבע ה-`SEED` לשחזור ויוגרלו אותם האירועים שנבחרו מקודם בזמן הוספת הסיור.

סדר הפעולות

הפעולות המתבצעות לצורך שליפה של אירוע היסטורי רנדומלי מתוך הקובץ/מהמפעל.

1. `FillEventFromBFile`

פונקציה זו הינה המעטפת הראשית, היא מקבלת הן קובץ בינארי והן רשימה מקושרת של כל האירועים השמורים ב-`FACTORY`.

2. `addRandomEvent`

- במידה ופרמטר שם הקובץ אינו ריק, תתבצע קריאה **לפעולה המיוחדת**.
- במידה ושם הקובץ ריק, יתבצע חיפוש רנדומלי על הרשימה המקושרת של האירועים.

3. `GetEventFromFileBySeek` – **הפעולה המיוחדת**

פעולה זו נעזרת ב-`SEEK` בכדי לנוע לכל אורך הקובץ הבינארי ולשלוף את הערך הספציפי (אירוע היסטורי) אותו היא רוצה, הפעולה מחזירה הצלחה או כישלון ומעדכנת את הפרמטר – מצביע לאירוע היסטורי.

דוגמה:

הערך הרנדומלי תמיד יהיה בטווח בין 0 לבין כמות האירועים (נשמר כערך ראשון בקובץ הבינארי).

בכדי להגיע לערך ה-`i` אשר יצא,

על ה-`seek` לקבל את הערך הבא:

$$offset = sizeof(int) + \sum_{n=1}^i (sizeof(DateTime) + sizeof(Description))$$

כאשר offset יהיה ערך ה SEEK החדש למציאת האירוע ההיסטורי הרנדומלי.

5. GeneralArray

לאור דרישות הפרויקט יישמנו את הפונקציה GeneralArrayFunction במספר מקומות:

1. שחרור ספקים בעזרת הפונקציה הייעודית freeSupplierPtr.
2. שחרור סיורים בעזרת freeCocaColaTourPtr.
3. הדפסת כל הספקים של המערכת בעזרת printSuppliersPtr.
4. הדפסת כל הסיורים של המערכת בעזרת printCocaColaTourPtr.
5. הדפסת כל חבילות הבקבוקים של המערכת בעזרת printBottlePackingPtr.

הערות

הפעולות הללו מיושמת עם GeneralArrayFunction במקומות שונים לאורך הקוד, היכן שנדרשו לבצעם.

6. Macros

בפרויקט יצרנו שלושה Macros המוגדרים בקובץ Macros.h:

- CHECK_RETURN – מקבל פונקציה ומצביע ובכישלון מחזיר 0.
- CLOSE_FILE_RETURN – עבור בעיה בערך חזרה כלשהו, המקרו מופעל עם קבלת הודעה להדפסה, קובץ לסגירה ומחזיר 0 לסיום.
- PRINT_TITLE – במהלך התוכנית, מודפסים כותרות שונות עבור התפריטים וחלק מהפעולות. מקרו זה מקבל צבע להדפסה למסך והודעה להדפסה.

7. פולימורפיזם

בפרויקט קבענו כי מבנה Employee יקיים פולימורפיזם.

מחלקת Employee הינה האב, אותה "יורשים" המבנים EmployeeGuide ו EmployeeDriver.

פירוט המבנים:

7.1 מבנה Employee

מבנה זה הינו מבנה האב, המאפיין את תכונותיו הבסיסיות של כל עובד והן את פעולותיו.

לקובץ המבנה מוגדר ENUM המכיל Driver, Guide

בנוסף לכך הגיל המינימלי לעובד הינו 16.

ראשית, בקובץ הזה ביצענו הגדרת typedef למספר פרמטרים:

- `typedef void (*fptrPrintEmployee)(Employee*);`
- `typedef void (*fptrFree)(Employee*);`

- `typedef int (*fptrWriteBFile)(FILE*, Employee* const);`
- `typedef int (*fptrWriteTFile)(FILE*, Employee* const);`

הגדרות אלו מסייעות להגדרת המצביעים לפונקציות של מבנה האב.

תכונות המבנה

- מספר סריאלי.
- הצבעה למבנה האב מסוג `void*` - עבור מבנה זה ההצבעה תהא `NULL` שכן אין מבנה תחתיו.
- שם העובד.
- גיל העובד.
- סוג העובד – שימוש ב `ENUM` המתואר לעיל.
- משך תעסוקה.
- הצבעה לפונקציית הדפסת העובד – `fptrPrintEmployee`
- הצבעה לפונקציית מחיקת עובד – `fptrFree`
- הצבעה לפונקציית כתיבת עובד לקובץ בינארי – `fptrWriteBFile`
- הצבעה לפונקציית כתיבת עובד לקובץ טקסט – `fptrWriteTFile`

פונקציות מרכזיות

1. יצירת עובד חדש – מבצעת הקצאה דינאמית. (מדמה `Constructor`) פונקציה זו מגדירה את מצביעי הפונקציות לפונקציות ספציפיות.
2. שחרור עובד – `destructor`.
3. הדפסת עובד.
4. כתיבה/קריאה לקבצי טקסט/בינאריים.

7.2 מבנה `EmployeeGuide`

מבנה זה "יורש" את תכונותיו של המבנה `Employee`.
מבנה זה מייצג מדריך במפעל.
תחת קובץ זה הוגדר `ENUM` לסוג ההשכלה (תיכונית, אקדמית, קורס).
בקובץ זה ביצענו הגדרת `typedef` חדשה:

- `typedef struct _EmployeeGuide EmployeeGuide;`
- `typedef void (*fptrTellFact)(Employee*, HistoricalEvent*);`

תכונות המבנה

- הצבעה למבנה הבסיס `Employee`.
- רמת משכל – על בסיס ה `ENUM` המתואר לעיל.
- מצביע לפונקציה – `tellFact`.

פונקציות מרכזיות

1. יצירת מדריך חדש – מבצעת הקצאה דינאמית. (מדמה `Constructor`) פונקציה זו מגדירה את מצביעי הפונקציות לפונקציות ספציפיות, יתרה מכך הפונקציה נעזרת ב `contstructor` של `Employee`.

2. שחרור מדריך – destructor.
3. הדפסת מדריך.
4. לספר עובדה – מקבלת מצביע למדריך ומצביע לאירוע ההיסטורי ומדפיסה את האירוע ההיסטורי.
5. כתיבה/קריאה לקבצי טקסט/בינאריים.

7.3 מבנה EmployeeDriver

- מבנה זה "יורש" את תכונותיו של המבנה Employee.
- מבנה זה מייצג נהג במפעל.
- תחת קובץ זה הוגדר ENUM לסוג הרישיון (C1,C,E,B).

תכונות המבנה

- הצבעה למבנה הבסיס Employee.
- סוג רישיון – על בסיס ה ENUM המתואר לעיל.

פונקציות מרכזיות

1. יצירת נהג חדש – מבצעת הקצאה דינאמית. (מודמה Constructor) פונקציה זו מגדירה את מצביעי הפונקציות לפונקציות ספציפיות, יתרה מכך הפונקציה נעזרת ב constructor של Employee.
2. שחרור נהג – destructor.
3. הדפסת נהג.
4. כתיבה/קריאה לקבצי טקסט/בינאריים.

8. אופן דחיסת המבנים לקובץ בינארי

8.1 פורמט כתיבת חברת קוקה קולה לקובץ Coca_Cola_Factory

[seed]

[כמות העובדים]

[עובד 1]

[עובד 2]

...

[עובד n]

כאשר כל עובד ישמר בפורמט הבא :

[סוג העובד - נהג/מדריך]

[id]

[שם העובד]

[גיל]

[ותק]

[פרט נוסף - בהתאם לסוג העובד]

- עבור נהג יישמר [סוג הרישיון]
- עבור מדריך יישמר [רמת השכלה]

[כמות הספקים]

[ספק 1]

[ספק 2]

...

[ספק n]

כאשר כל ספק יישמר בפורמט הבא :

[id]

[שם הספק]

[מספר טלפון]

[כתובת]

כאשר הכתובת תשמר בפורמט הבא :

[עיר]

[רחוב]

[מספר בית]

[כמות משאיות]

[משאית 1]

[משאית 2]

...

[משאית n]

כאשר כל משאית תשמר בפורמט הבא :

[לוחית רישוי של המשאית]

[id של הספק]

[id של הנהג]

[מספר חבילות]

[חבילה 1]

[חבילה 2]

...

[חבילה n]

כאשר כל חבילה תשמר בפורמט הבא :

[בקבוק - פרטיו יישמרו בדחיסה - פורמט בהמשך]

[כמות בקבוקים]

[כמות סיורים]

[סיור 1]

[סיור 2]

...

[סיור n]

כאשר כל סיור יישמר בפורמט הבא :

[תאריך סיור]

[id של המדריך]

[משך הסיור]

[כמות מבקרים]

הערות

1. האירועים של הסיורים יוגרלו מחדש לפי הseedn השמור בקובץ.

2. האירועים של המפעל נשמרים בקובץ בנפרד.

8.2 פורמט כתיבת חברת קוקה קולה לקובץ Historical_Events

[כמות האירועים]

[אירוע 1]

[אירוע 2]

...

[אירוע n]

כאשר כל אירוע יישמר בפורמט הבא :

[תאריך וזמן האירוע]

[תיאור האירוע]

8.3 אופן דחיסת בקבוק

32 ביטים סך הכול (4 בתים).

| | | | | | | | | | | | | | | | | | | | | |
|------------|---|----------|---|-----------|---|---|---|---|----|----|---------------|----|----|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| flavor (2) | | type (2) | | sugar (7) | | | | | | | calories (10) | | | | | | | | | |

Data Direction ->

| | | | | | | | | | | |
|--------|----|----|----|----|----|----|----|----|----|----|
| 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| ml(11) | | | | | | | | | | |

הסבר

- Flavor – סוג הטעם 2 ביטים (ישנם 4 סוגים)
- Type – סוג הבקבוק 2 ביטים (ישנם 3 סוגים)
- Sugar – 7 ביטים (127 הינה הכמות המקסימלית).
- Calories – 10 ביטים (1023 כמות מקסימלית).
- מ"ל – 11 ביטים (2000 כמות מקסימלית).

הערות

1. בקבוק ישמר בקובץ טקסט בפורמט רגיל, ואילו בקובץ בינארי באופן דחוס.

9. פעולות הניתנות לביצוע ע"י המשתמש

9.1 התפריט הראשוני

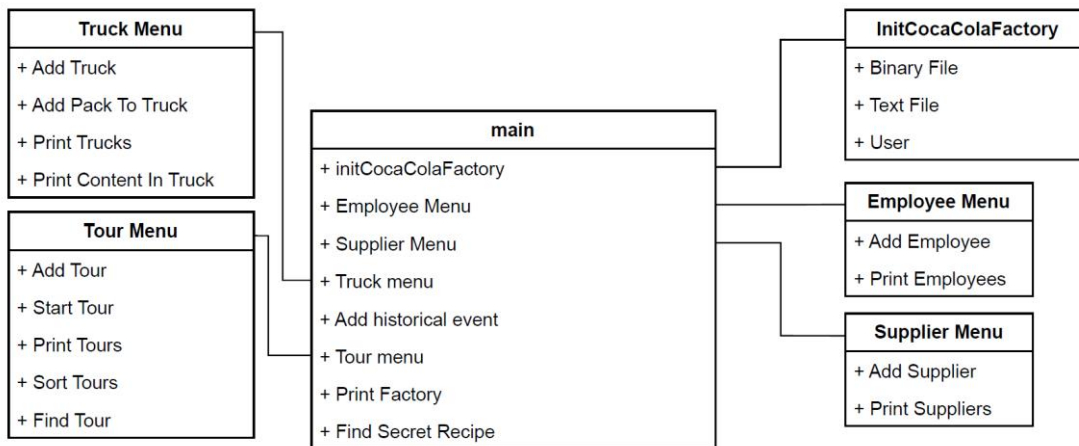
- 1 – העלאת נתוני המערכת מקובץ בינארי.
- 2 – העלאת נתוני המערכת מקובץ טקסט.
- 3 – אתחול המערכת על סמך נתוני המשתמש.

9.2 תפריט מרכזי

1. תפריט עובד
 - הוספת עובד
 - הדפסת כל העובדים
2. תפריט ספק
 - הוספת ספק
 - הדפסת כל הספקים
3. תפריט משאית
 - הוספת משאית
 - הוספת אריזה למשאית
 - הדפסת כל המשאיות
 - הדפסת מה המשאית העמיסה
4. הוספת אירוע היסטורי
5. תפריט סיור
 - הוספת סיור
 - התחלת סיור
 - הדפסת כל הסיורים
 - מיון סיורים
 - ❖ לפי משך הסיור
 - ❖ לפי תאריך
 - ❖ לפי מדריך
 - חיפוש סיור
 - ❖ לפי מדריך
 - ❖ לפי משך הסיור
 - ❖ לפי תאריך
6. הדפסת מפעל קוקה קולה
7. פענוח המתכון הסודי של הקוקה קולה.

10. שרטוט המערכת**10.1 שינויים מהשרטוט הראשוני**

1. מחלקת Employee – אנו החלטנו כי מימוש הפולימורפיזם יתבצע עבור מחלקת העובד. לכן, ביצענו שינוי גורף למחלקת העובד, כך שיקיים את עקרונות הפולימורפיזם (הוספת שני תתי מבנים נהג ומדריך).
2. העברת הseedn למפעל במקום לסיור, אנו רוצים להגדיר את הseedn פעם אחת בלבד במהלך העלאת המפעל מהקבצים.
3. הוספת id לספק.

10.2 שרטוט – פירוט הפונקציות הראשיות של המערכת**10.3 Class Diagram – שרטוט**

מצורף בנספח – עמוד אחרון.

11. חלוקת אחריות

הפרויקט מתבצע באופן משותף, ללא חלוקת אחריות מוגדרת. שנינו עובדים ומממשים יחד את כל המבנים והפונקציות הנדרשות.

