

הקדמה

בתרגיל זה נבנה מערכת המנהלת עובדים בחברה, לשם כך ניצור שתי מחלקות חדשות – מחלקת Employee ומחלקת DataBase. מחלקת Employee תכיל את המידע של העובד ומחלקת DataBase תנהל את עובדי החברה.

DataBase – מחלקת מאגר נתונים

מחלקה זו תנהל את העובדים בחברה.

שדות (data members):

- מצביע כפול לאובייקטים ממחלקת עובד (מצביע למצביע).
- מספר העובדים הנוכחי בחברה (int).

מתודות (שיטות):

- בנאי ברירת המחדל נבנה מערך בגודל 3 עובדים.
- בנאי העתקה
- מציאת עובד לפי ID :
 - תקבל ID ותחזיר מצביע לאותו העובד.
 - במידה ואין כזה עובד המתודה תחזיר null. (אין צורך בהודעת שגיאה במקרה זה).
- הוספת עובד :
 - מקבלת עובד חדש, ומוסיפה אותו למערך.
 - במידה ואין מספיק מקום – תיצור מערך חדש בגודל פי 2 מהנוכחי ותעתיק אליו את העובדים הישנים ואז תכניס את עובד החדש.
 - במידה וקיים עובד עם אותה תעודת הזהות במאגר העובד יימחק **ותודפס הודעת שגיאה**.
- מחיקת עובד :
 - תקבל ID של עובד ובמידה והוא חוקי (קיים) – תסיר אותו.
 - במידה ולא קיים עובד עם תעודת הזהות המבוקשת **ותודפס הודעת שגיאה**.
 - שימו לב שכעת צריך גם לצמצם את המערך, זאת אומרת במידה ומחקנו עובד מהאמצע – יש להזיז את כל העובדים שהיו אחריו מקום אחד אחורה במערך.
- הדפסת כל העובדים במאגר :
 - מאגר הנתונים **יודפס** באופן המתאור בדוגמא.

דוגמת הרצה :

עבור הקוד הבא :

```
int main() {  
  
    DataBase dataBase;  
    dataBase.addEmployee(new Employee((char*)"Sharon",11,10000));  
    dataBase.addEmployee(new Employee((char*)"Yael",11,11000));  
    dataBase.addEmployee(new Employee((char*)"Yael",12,11000));  
    dataBase.addEmployee(new Employee((char*)"Barak",10,20000));  
    dataBase.print();  
    dataBase.removeEmployee(1);  
    dataBase.removeEmployee(11);  
    dataBase.print();  
}
```

הפלט יהיה :

This employee is already in the database

Employees database:

Name:Sharon
ID:11
Salary:10000

Name:Yael
ID:12
Salary:11000

Name:Barak
ID:10
Salary:20000

This employee isn't in the database

Employees database:

Name:Yael
ID:12
Salary:11000

Name:Barak
ID:10
Salary:20000

Employee - מחלקת עובד

שדות (data members):

- השם הפרטי של העובד (char*).
- מספר מזהה של העובד (int).
- משכורת (double).

מתודות (שיטות):

בנאי עם פרמטרים:

- מקבל את הערכים: שם, מספר מזהה, ומשכורת (בסדר הזה).

בנאי העתקה:

- מקבל אובייקט ויוצר אובייקט חדש עם אותם הערכים (כפי שנלמד בכיתה).

בנאי ברירת מחדל:

- יוצר עובד עם הפרמטרים:
 - שם - "Name"
 - מספר מזהה - 0
 - משכורת - 0

ארגון מסיבה:

- הפונקציה מחזירה את עלות ארגון מסיבה לעובדים, כשהעלות מחושבת: (מספר העובדים הקיימים * 10).
- מספר העובדים יכול להשתנות בכל רגע בו יוצרים עובד חדש או מוחקים עובד ישן.
- שימו לב כי הפונקציה **לא** מקבלת כפרמטר את מספר העובדים.

אופרטורים:

עליכם לממש את האופרטורים הבאים על מנת שיהיה אפשר לבצע את הפעולות הבאות בין עובדים שונים.

אופרטור שוויון (==):

- שני עובדים יחשבו זהים במידה ויש להם את אותה **תעודת הזהות**, אחרת תוצאת ההשוואה תהיה FALSE.
- האופרטור מחזיר תשובה בוליאנית, TRUE אם התזו זהים ו FALSE אחרת.

אופרטור גדול מ (>):

- עובד נחשב "גדול" יותר מעובד אחר רק במידה **והמשכורת שלו בחברה גדול** (גדול ממש, לא שווה) מהמשכורת של העובד השני.
- האופרטור מחזיר תשובה בוליאנית, TRUE אם העובד הראשון גדול מהשני ו FALSE אחרת.

אופרטור השמה (=):

- אופרטור השמה בין שני עובדים משנה את **שם** העובד בלבד, כלומר אחרי פעולת ההשמה בין 2 עובדים רק השמות שלהם יהיו זהים (תעודות זהות ומשכורות יישארו כמו שהן).

אופרטור חיבור (+):

- פעולת חיבור בין שני עובדים מוסיפה את **המשכורת** של העובד מימין לאופרטור לזה שמשמאל.

אופרטור העלאה (++) :

- פעולת העלאה (++) עבור עובד מעלה לו את **המשכורת** ב100.
- יש לתמוך באופרטור משני הצדדים.

אופרטור הוספה והשמה (+=) :

- פעולת הוספה והשמה (+=) משנה את **שם** העובד שמשאל לאופרטור לשם שהוא שרשור השמות של שני העובדים (קיימת דוגמא בסוף).

אופרטור הדפסה (<<) :

- האופרטור **מדפיס** את פרטי העובד.
- הפורמט צריך להיות זהה לדוגמאות ההרצה בסוף.

****שימו לב שהאופרטורים עובדים על שדות שונים במחלקה**

דוגמת הרצה :

עבור הקוד הבא :

```
dataBase.print();
Employee *yael = DataBase.getEmployee(12);
Employee *barak = DataBase.getEmployee(10);

cout<<"Cost party for Barak: "<< barak->planOfficeParty() << "\n" <<endl;

if(*barak>*yael)
{
    cout<<"Barak's salary is higher than Yael's \n"<<endl;
}

cout <<+>(*barak)<<endl;

*barak+*yael;
*barak += *yael;

dataBase.print();

if(*barak>*yael)
{
    cout<<"Barak's salary is higher than Yael's \n"<<endl;
}
```

Employees database:

Name:Yael
ID:12
Salary:20000

Name:Barak
ID:10
Salary:11000

Cost party for Barak: 20

Name:Barak
ID:10
Salary:11100

Employees database:

Name:Yael
ID:12
Salary:20000

Name:BarakYael
ID:10
Salary:31100

Barak's salary is higher than Yael's

בדוגמא זו יש שימוש באופרטורים : <, >, <=, >=, וכן כן בפונקציה המדפיסה את עלות המסיבה.

הערות:

1. עליכם להתאים את חתימות הפונקציות שלכם (שם ופרמטרים) לדוגמאות ההרצה (מצורף גם קובץ main לדוגמא).
2. קובץ main המצורף לא בודק את כל הפונקציונליות של התוכנית, הוא עזר כדי לוודא שהחתימות נכונות.
3. יש להוסיף שדות, מתודות והורסים לפי הצורך.
4. יש להקפיד על פלטים דומים לאלו שיש בדוגמאות.
5. יש למחוק כל מה שהוקצה דינאמית לאחר השימוש בו, תהיה בדיקה לדליפת זיכרון.
6. הנכם נדרשים לתכנת בשפת C++ ולכן כל שימוש בפונקציות וספריות של שפת C אסורה, למעט שימוש בפונקציות strlen strcpy cout.
7. אין להשתמש במחלקת string (להשתמש ב char*).

הנחיות הגשה:

ההגשה בזוגות בלבד. עבודה שתוגש בפורמט אחר, לא תיבדק.

יש להגשי את כל הקבצים: .h, .cpp, .makefile.

כל הקבצים יוגשו יחדיו תחת קובץ בעל השם (ex4_ID1_ID2.zip) כמובן שיש לשנות את ID1, ID2 לתעודות הזהות שלכם.

מועד הגשה – עד לתאריך 19/12/2018 23:59

בהצלחה !!!