

תכנות מונחה עצמים

תרגיל 1

מועד הגשה 4/4/2021: בשעה 23:50

הוראות הגשה:

1. הגשה באופן עצמאי בלבד. הגשה בקבוצות תוביל לציון 0 בעבודה.
2. אין לשתף או להעתיק את העבודה או חלקים ממנה. עבירה על הוראה זו תוביל לציון 0 בעבודה.
3. בתחילת הקובץ יש להוסיף את התיעוד הבא:
/*Assignment: 1
Author: Israel Israeli, ID: 01234567*/
כמובן שיש לעדכן את השמות ומספרי תעודות הזהות שלכם.
4. חשוב לבדוק כי הפרוייקט מתקמפל בסביבת visual studio 2019 enterprise.
5. הגשה דרך מערכת מודול בלבד. שום עבודה לא מתקבלת במייל!
6. את החלק התאורתי יש להגיש כקובץ word/pdf.
7. יש להגיש קובץ אחד בפורמט RAR או ZIP המכיל את החלק התאורתי, תיקיה עבור שאלה 1 ותיקיה עבור שאלה 2.
8. אסור להוסיף משתנים למחלקה, אך ניתן להוסיף מתודות עזר (פרטיות).
9. ניתן להשתמש בתרגיל בספריות: cstring, iostream (ספריית מחרוזות שלמדתם ב-C) וב- random (הסבר בנספח).
10. יש להשתמש ב-const בכל מקום שניתן.
11. שאלות ובקשות בקשר לעבודה יש להפנות אך ורק לאחראי התרגיל, ליאור במייל: liorre@ac.sce.ac.il

שאלה 1 (מחלקות ואובייקטים)-30 נק':

במשימה זו עליכם לבנות מחלקה המייצגת מספר דו-סיפרתי

1. שדות (data members):
 - char שמחזיק בתוכו את ספרת האחדות של המספר.
 - char שמחזיק בתוכו את ספרת העשרות של המספר.
2. מתודות (שיטות):
 - בנאי ברירת מחדל אשר יצור את המספר 0 (0 בעשרות ואפס באחדות).
 - בנאים לא דיפולטיים יש לפחות שניים:
 - מקבל int כפרמטר ויוצר את האובייקט שלנו כך שהשדות שלו (ספרת האחדות והעשרות שלו) מאותחלות לפי כספרת האחדות והעשרות בפרמטר ה-int.
 - מקבל שני תווי char כפרמטר ומאתחל את השדות לפיהם.
 - המרה למספר (int). המתודה תחזיר את הערך ה-int של המספר השמור בתוכה. (לדוגמא אם האובייקט מכיל את התו '9' בספרת העשרות ו-'1' באחדות הפונקציה תחזיר את המספר 91).
 - שימו לב האובייקט **לא משתנה**.
 - שינוי ערך. המתודה מקבלת int כפרמטר ומשנה את האובייקט לפי הערך שקיבלה (**רמז ליעילות reuse**: פעולה דומה לבנאי...).
 - הדפסת הערך. המתודה תדפיס את ערך האובייקט למסך ואז תרד שורה. שימו לב האובייקט **לא משתנה**.

כתוב תוכנית ראשית (דרייבר) אשר:

1. מאתחלת שלושה אובייקטים שונים (כל אחד בעזרת שיטה/בנאי אחר).
2. בודקת את כלל הפונקציות של המחלקה, כלומר: מפעילה בעזרת האובייקטים הנוצרים את מתודות ההמרה, שינוי הערך והדפסת הערך.

עליכם לחלק את התרגיל ל-3 קבצים (דרייבר, קובץ cpp המהווה מימוש המחלקה וקובץ h).

שאלה 2 (הכלה)-62 נק':

בשאלה זו אתם תיצרו **משחק איקס עיגול – ויקיפדיה(wikipedia.org)** לשתי שחקנים.

אתם תעשו זאת בעזרת **יחס הכלה**.

הקבצים שיש להגיש:

Cell.h
Cell.cpp
Board.h
Board.cpp
Player.h
Player.cpp
Game.h
Game.cpp
driver.cpp

תיאור העבודה:

תיאור המשחק

משחק איקס עיגול הינו משחק המשוחק על ידי שתי שחקנים על לוח של 9 משבצות (3 על 3). בכל תור, השחקן מסמן משבצת לפי הסימון שניתן לו (X או O), השחקן המנצח הוא זה אשר הצליח למלא שורה, עמודה או אלכסון בסימון שלו.

אם אף שחקן לא הצליח לעשות זאת לפני שכלל המשבצות התמלאו, אז המשחק נגמר בתיקו.

מחלקות המשחק

מחלקת תא (Cell): מחלקה אשר מייצגת תא בלוח.

שדות המחלקה(פרטי):

- **char cell** המייצג את הסימון של התא(ברירת מחדל שלו הוא ' ' -תו ריק).
- **bool free** אינדיקטור לכך שהתא פנוי או תפוס (כלומר, אם הוא מכיל תו ריק או שאחד השחקנים כבר סימן אותו).

מתודות המחלקה(ציבוריות):

- **בנאי ברירת מחדל** בנאי המאתחל את שדות התא ל-תא ריק ופנוי.
- **הדפסה** מתודה אשר מדפיסה את התא.
- **setter** משנה את המצב הפנימי של התא. כלומר, מקבל תו ומעדכן את שדות התא (כולל עדכון על כך שהתא אינו ריק).
- **getter** מחזיר את הסימון של התא.
- **Is_free** מחזירה אינדיקציה על מצב התא (פנוי או לא).

מחלקת לוח (Board):

שדות המחלקה(פרטי):

- **Cell* board** מערך דינאמי המייצג את לוח המשחק(שימו לב, מדובר בהכלה של Cell).
 - **int size** גודל הלוח.
- מתודות המחלקה(ציבוריות):
- **בנאי ברירת מחדל** הבנאי מאתחל את הגודל של הלוח ל-0 ואת הלוח עצמו ל-nullptr.
 - **בנאי המקבל גודל**, מעדכן את שדה הגודל ומקצה זיכרון בהתאם (שימו לב, במקרה זה מופעל בנאי דיפולטיבי על כל תאי המערך).
 - **בנאי מעתיק** הבנאי מקבל אובייקט מסוג לוח ומעתיק אותו.
 - **הורס** הדואג לשחרר את הזיכרון.
 - **הדפסה** פונקציה אשר מדפיסה את הלוח(ניתן לראות איך הלוח נראה בהמשך).

- **set_symbol** אשר מקבלת מספר תא ותו שיש להכניס לתא, הפונקציה בודקת האם ניתן להכניס את התו לתא (גבולות מערך, חוקיות תו, האם התא תפוס או לא), במידה ואפשר היא מעדכנת את התא. בכל מקרה, המתודה מחזירה אם הצליחה או לא.
- **count_board** מתודה אשר מחזירה את כמות התאים הריקים בלוח המשחק.
- **get_symbol** מקבלת את מספר התא הרצוי ומחזירה את התוכן שלו.

מחלקת שחקן (Player):

שדות המחלקה (פרטי):

- **char* name** מערך דינאמי המייצג את שם השחקן.
- **char symbol** הסימון הניתן לשחקן (איקס או עיגול).
- **בנאי מעתיק** הכרזה בלבד בתוך private (בכדי שלא יוכלו לשכפל את השחקן).
- מתודות המחלקה (ציבוריות):
- **בנאי ברירת מחדל** הבנאי מאתחל את שם השחקן ל-nullptr ואת הסימון ל-תו ריק.
- **בנאי** המקבל שם וסימון ומעדכן את השחקן בהתאם.
- **הורס** הדואג לשחרר את הזיכרון.
- **הדפסה** פונקציה אשר מדפיסה את הלוח (ניתן לראות איך הלוח נראה בהמשך).
- **make_move** קולטת מהמשתמש את המהלך שהוא מעוניין לבצע ומחזירה אותו.
- **הדפסה** אשר מדפיסה את פרטי השחקן.
- **פונקציות set ו-get לכלל השדות.**

מחלקת משחק (Game):

מחלקה זו אחראית לנהל את כל המשחק, המחלקה משתמשת בפונקציונליות של הלוח ובפונקציונליות של השחקן בשביל לעשות זאת.

שדות המחלקה (פרטי):

- **Board board** שדה המייצג את לוח המשחק.
- **Player players[2]** מערך סטטי של שתי שחקנים.
- הגדרה של מצבי המשחק האפשריים בעזרת enum, enum statuses {win, draw, start}.
- **statuses status** המייצג את מצב המשחק.
- **char turn** אשר שומרת בתוכה את הסימון של השחקן שתורו כעת.
- מתודות המחלקה (ציבוריות):
- **בנאי** אשר מקבל את שמות השחקנים.
- מאתחל את השחקנים והלוח, לצורך הפשטות השחקן במקום 0 מקבל סימון 'X' והשחקן במקום 1 מקבל סימון 'O'.
- בנוסף, הבנאי בוחר בצורה רנדומלית את השחקן אשר מתחיל.
- **void play** אשר מנהלת את המשחק.
- המתודה רצה בלולאה עד אשר המשחק נגמר (תיקו או ניצחון של אחד השחקנים). השלבים הכללים של המשחק:
 - ניקוי המסך (ניתן לראות בנספחים איך לעשות זאת).
 - הדפסת המסך.
 - עשיית צעד של השחקן שתורו הגיע (כמובן, יש לבדוק תקינות הצעד ובמידת הצורך לקלוט צעד חדש).
 - עדכון סטטוס המשחק לאחר הצעד והחלטה לאן להמשיך בהתאם (אם יש תיקו או ניצחון אז להכריז על כך ולסיים את המשחק, אחרת לשנות את התור של השחקן ולהמשיך את המשחק).

דוגמא לריצת המשחק:

1. תחילה השחקנים מתבקשים להכניס את פרטיהם:

```

C:\Users\lior\source\repos\hw1sol\Debug\lab1.exe
enter the first name:
lior
enter the second name:
alex
  
```

2. לאחר שהפרטים הוכנסו, יופיע לוח ריק והתוכנית תחכה לקלט מהמשתמש

```

Select C:\Users\lior\source\repos\hw1sol\Debug\lab1.exe
|_|_|_|
|_|_|_|
|_|_|_|
choose your move
Player name: alex with symbol: O
  
```

שימו לב למספר דברים חשובים:

- השחקן בשלב זה נבחר רנדומלית על ידי המחשב.
- שם השחקן והסימול מופיע על המסך כחלק מהבקשה לביצוע הצעד.
- השחקן צריך להכניס ספרות מ-1 עד 9 כאשר: 1 היא המשבצת השמאלית העליונה ו-9 היא המשבצת הימנית התחתונה.
- שימו לב שעליכם לבצע המרה מהמספר הנכנס מהמשתמש למספר המתאים למערך (0-8).

3. לאחר בחירת הצעד(למשל 2) המסך יראה כך:

```

C:\Users\lior\source\repos\hw1sol\Debug\lab1.exe
|_|O|_|
|_|_|_|
|_|_|_|
choose your move
Player name: lior with symbol: X
  
```

כלומר, הלוח המעודכן יוצג על המסך והמשחק יצפה לקלט מהשחקן השני.

4. המשחק ימשיך ככה עד שיקרה אחד משלושת המצבים הבאים:

ניצחון של O:

```

Microsoft Visual Studio Debug Console

  |  |  |
--|--|--
  | O | X
--|--|--
X | O | 
--|--|--
  | O | 

we have a winner: Player name: alex with symbol: O

C:\Users\liorr\source\repos\hw1sol\Debug\lab1.exe (process 17228) exited with code 0.
Press any key to close this window . . .

```

ניצחון של X:

```

Microsoft Visual Studio Debug Console

O | O | X
--|--|--
  | O | 
--|--|--
X | X | X

we have a winner: Player name: lior with symbol: X

C:\Users\liorr\source\repos\hw1sol\Debug\lab1.exe (process 13704) exited with code 0.
Press any key to close this window . . .

```

תיקון

```
Microsoft Visual Studio Debug Console

X | O | X
---|---|---
O | O | X
---|---|---
X | X | O

It is a Draw

C:\Users\liorr\source\repos\hw1sol\Debug\lab1.exe (process 9468) exited with code 0.
Press any key to close this window . . .
```

בנוסף, במידה והשחקן ביצע מהלך לא חוקי: צעד מחוץ לגבולות המערך (עבורו חורג מ-1 עד 9) או שניסע לבצע מהלך על משבצת לא פנויה -> המשחק יבקש קלט מחדש: המערך מ-0 עד 8)

```
C:\Users\liorr\source\repos\hw1sol\Debug\lab1.exe

  |  | 
--|--|--
  | 0 | 
--|--|--
  |  | 
--|--|--
  |  | 

choose your move
Player name: lior with symbol: X
-3
Please try again
2
0
Please try again
```

הדרייבר של התוכנית:

```
#include "game.h"
#include <iostream>
using namespace std;
int main() {
    char name1[60], name2[60];
    cout << "enter the first name: " << endl;
    cin >> name1;
    cout << "enter the second name: " << endl;
    cin >> name2;
    Game game(name1,name2);
    game.play();
    return 0;
}
```

נספחים:

- ניקוי מסך: מתבצע על ידי הפקודה `system("cls")` שימו לב, פקודה זו עובדת רק בוינדוס ולכן, אלו ממכם אשר בונים בסביבות אחרות, תוסיפו אותה כאשר תבדקו את העבודה בויזואל של וינדוס.
- בחירת מספר רנדומלי: ניתן לבחור מספר רנדומלי בצורה הבאה:

```
#include <random>//
int main() {
    srand(time(nullptr)); //אתחול מחולל רנדומלי
    int pr = rand() % 10; // מחזיר מספר מ-0 עד 9
    pr = rand() % 2; // מחזיר 0 או 1
    return 0;}

```


שאלה 3: (חלק תאורתי)-8 נק'

1. מה יקרה אם נגדיר פונקציה עם לולאה כ- `inline`?
2. מה יקרה אם לא נוסף `#ifndef`, `define`, `endif` ב- `h`?
3. מדוע עלינו להשתמש באופרטורים `new` ו-`delete` על פני הפונקציות אשר נלמדו בקורס המבוא?
4. במידה ולא נשתמש במצייני גישה כלומר ב- `public` / `private` במחלקה, האם נוכל לגשת מה- `driver` לפונק' המחלקה?
5. מהי שורת האתחול, למה היא משמשת ובאיזה תרחישים אנו מחויבים להשתמש בה?
6. מתי אנו חייבים לממש בנאי העתקה משלנו?
7. האם ניתן להחזיר אובייקט לוקלי `by reference`, יש לנמק.
8. סטודנט רשם את הבנאי ההעתקה הבא:

```
Myclass:Myclass(const Myclass obj){...}
```

 - א. האם קיימת בעיה בהגדרת הבנאי?
 - ב. אם כן, האם השגיאה שתתרחש היא שגיאת זמן ריצה או קומפילציה?
 - ג. אם כן, איך ניתן לתקן את הבעיה?