

תכנות מונחה עצמים**עבודת הגשה 5**

מועד הגשה: 14/6/2020 בשעה 23:50

הוראות הגשה:

1. **אנא קראו בעיון את כל תיאור העבודה בטרם תתחילו לכתוב קוד.**
2. הגשה באופן עצמאי בלבד. הגשה בקבוצות תוביל לציון 0 בעבודה.
3. אין לשתף או להעתיק את העבודה או חלקים ממנה. עבירה על הוראה זו תוביל לציון 0 בעבודה.
4. הגשה דרך מערכת מודול בלבד. **שום עבודה לא מתקבלת במייל!**
5. יש למקם כל מחלקה שיהיה עליכם ליצור, בשני קבצים נפרדים H ו-CPP. יש להכניס את החלק התיאורטי בקובץ וורד נפרד. יש להכניס את כל הקבצים של החלק המעשי + קובץ הוורד לתיקיה אחת, ואז לכווץ יחד. נדרש להגיש קובץ אחד בפורמט RAR או ZIP המכיל את כל הקבצים של כל השאלות. לקובץ המכווץ יהיה שם המהווה את מספר ת.ז. של המגיש.
6. **שאלות בקשר לעבודה להפנות אך ורק לאחראית לתרגיל, יעל וקסלר, במייל:**
yaelva@ac.sce.ac.il

חלק א' – תאורטי (מענה בקובץ טקסט – וורד): 12 נקודות

1. 2 נק' מחלקת הבסיס מכילה מתודה pure virtual שלא מימשנו במחלקה הנגזרת. האם ניתן ליצר אובייקטים מסוג המחלקה הנגזרת? יש לנמק.
2. 2 נק' הסבירו מה ההבדל בין קישור סטטי לקישור דינאמי ותארו מתי קורה כל אחד מהם.
3. 2 נק' נניח שנתונה מחלקה base שממנה יורשים באופן וירטואלי מחלקות A ו-B. נניח כי נתונה מחלקה AB שיורשת מ-A ו-M-B. מי מאתחל את הבנאי של base?
4. 6 נק' נתון הקוד הבא:

```
#include <iostream>
using namespace std;
```

```
class A{
public:
    A() {cout<<"A::C'tor"<<endl;}
    virtual ~A() {cout<<"A::D'tor"<<endl;}
    virtual void func1() {cout<<"A::func1()"<<endl;}
    virtual void func2(){cout<<"A::func2()"<<endl; func1();}
    virtual void func3() = 0;
};

class B1: virtual public A{
public:
    B1() {cout<<"B1::C'tor"<<endl;}
    virtual ~B1() {cout<<"B1::D'tor"<<endl;}
    void func1() {cout<<"B1::func1()"<<endl;}
    virtual void func3() {cout<<"B1::func3()"<<endl; func2();}
};

class B2: virtual public A {
public:
    B2() {cout<<"B2::C'tor"<<endl;}
```

```

~B2() {cout<<"B2::D'tor"<<endl;}
void func2() const {cout<<"B2::func2()"<<endl;}
};

class C :public B1, public B2{
public:
C () {cout<<"C::C'tor"<<endl;}
~C() {cout<<"C::D'tor"<<endl;}
virtual void func1() {cout<<"C::func1()"<<endl;}
void func3() {cout<<"C::func3()"<<endl;}
};
    
```

א. מלאו את הטבלאות הוירטואליות של המחלקות הנ"ל.

A class VTABLE	B1 class VTABLE	B2 class VTABLE	C class VTABLE

שימו לב: חובה לרשום לפני כל מתודה לאיזו מחלקה היא שייכת ע"י הוספת הקידומת
 ClassName:: לפני כל מתודה, כמו כן, יש לשמור על הסדר הנכון בטבלאות
 הוירטואליות.

א. מה הפלט המלא של קטע הקוד הבא:

```

int main()
{
    A* a = new B1();
    a->func2();
    B2* b2 = new C();
    b2->func3();
    delete a;
    delete b2;
    return 0;
}
    
```

חלק ב' – מעשי (ההגשה היא של קבצי ה- CPP ו-H בלבד) – 100 נקודות:**רקע:**

בעבודה זו, עליכם לממש מערכת ניהול חולים - המשבצת רופאים וצוותים מקצועיים למחלקות השונות ומנהלת את מערך החולים. לצורך בניית הפתרון עליכם להשתמש במספר מחלקות שיפורטו בהמשך, אשר **היחסים** בניהן יכולים להיות: **הכלה**, **הורשה**. עליכם לבחור את היחסים הרלוונטיים בהתאם לנתונים שקיבלתם. במימוש המחלקות, יש להשתמש בירושה בצורה יעילה (כלומר, מחלקת הבן נעזרת במתודות של מחלקת האב למימוש המתודה שלה – אם זה אפשרי) ובמקומות הנחוצים יש להשתמש במחלקות אבסטרקטיות במקומות הנכונים. הקפידו על תכנות נכון encapsulation, const, reference (וכו') אין להשתמש באובייקטים גלובליים.

הפתרון יש לממש בעזרת פולימורפיזם ו- RTTI.

- ✓ בבית החולים יש 4 מחלקות בהם – 3 רופאים ו- 3 אחיות.
 - ✓ בכל מחלקה בבית החולים יכולים להיות מקסימום 10 מאושפדים.
 - ✓ השנה הוקמה מחלקה חדשה בבית החולים, מחלקת הקורונה.
 - ✓ המשכורת בסיס לכל עובד היא $6500 = \text{basis}$.
 - ✓ כל איש צוות יכול להיות מצוות למחלקה אחת בלבד.
 - ✓ חולה יכול להיות שייך למחלקה אחת בלבד.
- (1) מחלקה המייצג **בן אדם** (כל האנשים אשר חולים או עובדים בבית החולים):
יש לשמור לכל בן אדם:

- שם פרטי ושם משפחה מסוג string כל אחד.
- מספר ת.ז מסוג string.
- יש להגדיר למחלקה את המתודות הבאות:
- בנאים
- הורס
- מתודה המדפיסה את פרטי הבן אדם.
- למחלקה הזו ניתן להוסיף מתודות לפי הצורך.

(2) מחלקה המייצג עובד:

העובד הוא גם בן אדם ויש לשמור לכל עובד בנוסף:

- שנות הותק של העובד מסוג int.
- יש להגדיר למחלקה את המתודות:
- בנאים
- הורס
- מתודה המחזירה את המשכורת.
- מתודה בוליאנית הבודקת האם העובד הנבחר הוא מצטיין. (בשלב זה המתודה לא תכיל חוקית ספציפית).
- מתודה המדפיסה את פרטיו של העובד, כולל הדפסת משכורתו ושנות הותק שלו.
- למחלקה זו ניתן להוסיף שדות ומתודות לפי הצורך.

(3) מחלקה המייצגת אחות:

האחות היא גם עובדת ויש לשמור לכל אחות בנוסף:

- שם המחלקה שהיא משוייכת אליה (יהיה שם בית החולים)
- כמות ההכשרות והקורסים שהיא עשתה.

יש להגדיר למחלקה את המתודות :

- בנאים
- הורס
- מתודה בוליאנית הבודקת האם האחות מצטיינת . אחות מצטיינת אם על כל שנת ותק היא עשתה לפחות הכשרה אחת.
- מתודה המחזירה את המשכורת . נסמן ב-x את מספר ההכשרות. משכורתה תהיה $basis+(x*50)$
- מתודה המדפיסה את פרטיה של האחות, כולל את מספר ההכשרות שעברה ואת המשכורת שלה. למחלקה הזו ניתן להוסיף מתודות לפי הצורך.

(4) מחלקה המייצגת רופא:

הרופא הוא גם עובד ויש לשמור לכל רופא בנוסף:

- שם המחלקה שהוא שייך אליה (יהיה שם בית החולים)
- האם הרופא הוא פרופסור – משתנה מסוג bool.
- מספר הערכות חיוביות שקיבל מעמיתיו ומנהליו.

יש להגדיר למחלקה את המתודות :

- בנאים
- הורס
- מתודה המחזירה את המשכורת . נסמן ב-x את שנות הוותק וב-y את מספר הערכות החיוביות שקיבל בהתאמה משכורתו תהיה שווה ל- $basis+200*x+50*y$. במידה והרופא בדרגת פרופסור תהיה תוספת נוספת למשכורתו על סך 2000 ש"ח.
- מתודה בוליאנית הבודקת האם הרופא מצטיין. רופא מוגדר כמצטיין אם הוא בעל יותר מ-10 הערכות חיוביות.
- מתודה המדפיסה את פרטי הרופא, כולל הדפסת המשכורת ואת שם המחלקה שהוא שייך אליה, אם הוא בעל דרגת פרופסור.
- למחלקה זו ניתן להוסיף מתודות לפי הצורך.

(5) מחלקה המייצגת חולה:

החולה הוא גם בן אדם ויש לשמור לכל חולה בנוסף :

- שם המחלקה שהוא שייך אליה (יהיה שם בית החולים)
- מספר בדיקות הקורונה שעשו לו.
- מספר בדיקות הקורונה שיצאו שליליות

יש להגדיר למחלקה את המתודות :

- בנאים
- הורס
- מתודה המחזירה את מספר הבדיקות שיצאו שליליות
- מתודה המדפיסה את פרטיו של החולה, כולל מספר הבדיקות השליליות שיצאו לו.
- למחלקה זו ניתן להוסיף שדות ומתודות לפי הצורך.

(6) מחלקה המייצגת מחלקת קורונה.

יש לשמור לכל קבוצה:

- שם בית החולים
- מערך המצביעים לרופאים השייכים למחלקה
- מערך המצביעים לאחיות השייכות למחלקה
- מערך המצביעים לחולים המאושפדים במחלקה

יש להגדיר למחלקה את המתודות :

- בנאים

- הורס

- מתודה שמוסיפה רופא - המתודה מקבלת מצביע לרופא ומוסיפה אותו למערך הרופאים.

- מתודה הוספת אחות - המתודה מקבלת מצביע לאחות ומוסיפה אותה למערך האחיות.

- מתודה להוספת חולה - המתודה מקבלת מצביע לחולה ומוסיפה אותו למערך החולים. **שימו לב**

שלא ניתן להוסיף חולה למחלקה במידה ואין לפחות רופא אחד ושתי אחיות.

- מתודה שמדפיסה את פרטי המחלקה: שם בית החולים, מערך העובדים ואת מספר החולים

המאושפזים במחלקה.

למחלקה זו ניתן להוסיף מתודות לפי הצורך.

(7) מחלקה - ניהול משבר הקורונה:

יש לשמור במחלקה הזו:

- מערך דינאמי של מצביעים לכל החולים

- מערך דינאמי של כל המחלקות

- מערך דינאמי של כל העובדים, כלומר Worker**

* הערה: אין צורך ביצירת אותו אובייקט פעמים (ניתן להוסיף ולשמור את הכתובת המקורית ולא ליצור העתק).

יש להגדיר למחלקה את המתודות :

-בנאי ברירת המחדל :

1. מאתחלים את מערך העובדים - בשלב ההתחלתי יהיו 2 אחיות ורופא אחד.

2. מאתחלים את מערך המחלקות - בשלב ההתחלתי יש מחלקה אחת של קורונה - המחלקה

מורכבת מרופא אחד ושתי אחיות (העובדים שאותחלו במערך העובדים).

3. מערך החולים יהיה ריק בשלב ההתחלתי

-הורס

- מתודה **Menu** אשר תדפיס למשתמש את כל האופציות הנתונות בפניו בתפועל המערכת ותיתן לו

את היכולת לבחור איזו אפשרות ברצונו להפעיל. המתודה תיתן למשתמש את האפשרויות הבאות:

1 הוספת אחות - על המשתמש להקליד את שם בית החולים שאליו האחות מצטרפת ופרטי

האחות (שם, שם משפחה, ת.ז, ותק, ומספר הכשרות). על המערכת להוסיף את האחות

למחלקת קורונה המתאימה ולמערך הכללי של העובדים. במידה ובן אדם עם אותו ת.ז. כבר קיים

במערך העובדים יש להדפיס הודעה מתאימה ולא להוסיף אותו.

2 הוספת רופא - על המשתמש להקליד את שם בית החולים שאליו הרופא מצטרף ופרטי הרופא

(שם, שם משפחה, ת.ז, ותק, האם הוא פרופסור, ומספר הפידבקים החיוביים). על המערכת

להוסיף את הרופא למחלקת קורונה המתאימה ולמערך הכללי של העובדים. במידה ובן אדם עם

אותו ת.ז. כבר קיים במערך העובדים יש להדפיס הודעה מתאימה ולא להוסיף אותו.

3 הוספת מחלקה - על המשתמש להקליד את שם בית החולים שעבורו הוא רוצה להוסיף

מחלקת קורונה מתוך רשימה ידוע של בתי חולים לדוגמה :

```
string HospitalNames[] = { "Sheba", "Wolfson", "Rambam", "Assaf  
Harofeh", "Soroka"};
```

לכל מחלקה חדשה יש רופא אחד ושתי אחיות. ניתן להקים מחלקה רק בבית חולים שאין בו עדין מחלקת קורונה.

4 הדפסת פרטי המחלקה לפי שם - על המשתמש לבחור מחלקה מתוך רשימת המחלקות

הקיימות על מנת שיודפסו פרטי המחלקה. הפרטים שיודפסו: שם בית החולים, מערך העובדים -

רופאים ואחיות ומספר החולים שמאושפזים במחלקה הנבחרת.

5 הדפסת פרטי הרופאים - המערכת תדפיס את כל הפרטים של הרופאים המטפלים במחלקות הקורונה השונות.

6 הדפסת משכורת של עובד לפי ת.ז. - על המשתמש להקליד את ת.ז. של העובד שעבורו הוא רוצה לקבל את השכר. במידה ולא קיים בן אדם עם אותו ת.ז. במערך העובדים יש להדפיס הודעה מתאימה.

7 הוספת חולה - על המשתמש להקליד את שם בית החולים שאליו הוא רוצה להוסיף מטופל ואת פרטי החולה (שם, שם משפחה, ת.ז., מספר בדיקות קורונה שעשה בסך הכל וכמה מהם היו שליליים). על המערכת להוסיף את החולה למחלקת קורונה המתאימה ולמערך הכללי של החולים. במידה ובן אדם עם אותו ת.ז. כבר קיים במערך החולים יש להדפיס הודעה מתאימה ולא להוסיף אותו.

8 הוספת בדיקת קורונה לחולה קיים - על המשתמש להקליד את ת.ז. של החולה ולאחר מכן האם התוצאה היתה חיובים או שלילית. המערכת תעדכן עבור אותו חולה את מספר הבדיקות שלו ואת כמות הבדיקות השליליות בהתאם לתוצאה שהתקבלה. במידה ולא קיים בן אדם עם אותו ת.ז. במערך החולים יש להדפיס הודעה מתאימה.

9 שחרור חולים - המערכת תעבור על כל החולים הקיימים במערך החולים ובמידה שלחולה יש 2 בדיקות שליליות הוא ישוחרר מבית החולים. שימו לב שעל המערכת למחוק את החולה ממערך החולים וגם מהמחלקה בה היה מאושפז.

10 הדפסת מצטיינים - המערכת תדפיס את כל הפרטים של כל האנשים המצטיינים אשר עובדים במחלקות הקורונה השונות (רופאים ואחיות). לפני הדפסת פרטי העובד יודפס מה התפקיד שלו.

11 הדפסת פרטי האחות עם המשכורת המינימלית - יש להדפיס את כל הפרטים הרלוונטיים עבור האחות עם המשכורת המינימלית ביותר שקיים במערך.

12 Exit - יציאה מהמערכת.

להלן מתודת התפריט -

```
void Menu() {
    int option;
    bool run = true;
    while (run) {
        printOptions();
        cin >> option;
        switch (option) {
            case ADD_NURSE:
                AddNurse();
                break;
            case ADD_DOCTOR:
                AddDoctor();
                break;
            case ADD_DEPARTMENT:
                AddDepartment();
                break;
            case PRINT_DEPARTMENT_DETAILS:
                PrintDepartmentDetails();
                break;
            case PRINT_DOCTORS_DETAILS:
                PrintDoctorsDetails();
                break;
        }
    }
}
```

```

case PRINT_SALARY_BY_ID:
    PrintSalary();
    break;
case ADD_PATIENT:
    AddPatient();
    break;
case ADD_CORONA_TEST_TO_PATIENT_ID:
    AddCoronaTestToPatientById();
    break;
case RELEASING_PATIENTS:
    Releasing();
    break;
case PRINT_EXCELLENCE_WORKER:
    PrintExcellenceWorker();
    break;
case PRINT_NURSE_DETAILS_WITH_SMALLEST_SALARY:
    PrintNurseWithSmallestSalary();
    break;
case EXIT:
    run = false;
    break;
    }
    }
    }

```

להלן מימוש המתודה (printOptions() :

```

void printOptions() {
    cout << "Menu: " << endl;
    cout << "1. ADD NURSE" << endl;
    cout << "2. ADD DOCTOR" << endl;
    cout << "3. ADD DEPARTMENT" << endl;
    cout << "4. PRINT DEPARTMENT DETAILS" << endl;
    cout << "5. PRINT DOCTORS DETAILS" << endl;
    cout << "6. PRINT SALARY BY ID" << endl;
    cout << "7. ADD PATIENT" << endl;
    cout << "8. ADD CORONA TEST TO PATIENT ID" << endl;
    cout << "9. RELEASING PATIENTS" << endl;
    cout << "10. PRINT EXCELLENCE WORKER" << endl;
    cout << "11. PRINT NURSE DETAILS WITH SMALLEST SALARY" << endl;
    cout << "12. EXIT" << endl;
}

```

למחלקה הזו ניתן להוסיף מתודות אך הן חייבות להיות **פרטיות** למחלקה (מתודות **get** הכרחיות יכולות להיות ציבוריות).

להלן הפונקציה הראשית (main) :

```
#include "CoronaVirusCrisisManagement.h"
```

```
int main() {  
    CoronaVirusCrisisManagement l;  
    l.Menu();  
    return 0;  
}
```

עבודה פוריה !!!