

המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

ת.ז. הסטודנט:	ברקוד נבחן
מספר חדר:	
מספר נבחן:	
מספר אסמכתא:	

מבחן בקורס: תכנות מונחה עצמים

תאריך הבחינה: 10.07.18

שנת הלימודים: תשע"ח, סמסטר: ב', מועד: א'.

משך הבחינה: 5 שעות

שם המרצה/ים:	שם המתרגל/ים:
ד"ר אליהו חלסצ'י	מר חיים שפיר
	מר אביב שוקרון
	גב' מיכל חבאני

מבנה הבחינה: הבחינה מורכבת מחלק אחד.

מספר השאלות בבחינה: 3

משקל כל שאלה: בצמוד לכל שאלה

הוראות לנבחן:

- מותר השימוש בכל חומר עזר לרבות חומר מודפס, עמדת המחשב במעבדה, דיסק און קי.
 - אסור השימוש בכל אמצעי תקשורת לרבות מחשב אישי, מחשב לוח או טלפון.
- יש לענות במחשב
- לא נדרש להחזיר את השאלון
- לא מצורף נספח לבחינה
- מחברת טיוטה: אין
- מחברת נפרדת לכל שאלה: לא
- למבחן בדיקה אוטומטית ולכן יש להגיש ע"פ הוראות ההגשה במדויק!!!
- שאלה שלא תתקמפל או שתהיה לה שגיאת ריצה ציונה אוטומטית 0. אין בדיקה ידנית למבחן. יש מספיק זמן למבחן כדי לנפות את כל השגיאות.
- אזהרה: למבחן מתקיימת בדיקת העתקות אוטומטית ובעבר סטודנטים שהעתיקו הועלו לוועדת משמעת והורחקו מהמכללה לשנה. אל תעתיקו במבחן.

בהצלחה!!

המסלול האקדמי המכללה למנהל

ביה"ס למדעי המחשב

שאלה 1 תכנות גנרי, STL, object functions (30 נק'):

בקובץ Q1.h נתונה לכם המחלקה PriorityQueue המהווה **תור עדיפויות** גנרי – תור בו כל איבר נכנס לפי עדיפותו לתוך התור. האיברים כמובן מטיפוס גנרי.

עליכם להשלים את האלמנטים הבאים כך שנוכל להשתמש במחלקה זו כבדוגמת הקוד הבאה:

- הגדרת המחלקה - האם יש צורך בהגדרות תבנית וכיצד אם כן?
- Data members - בפרט, את התור עצמו עליכם לממש באמצעות רשימה (list) של STL, ואין להשתמש ב priority_queue המוכן של STL, או בפונקציית מיון מוכנה.
- המתודות:
 - push שמכניסה אלמנט לתור העדיפויות ע"פ העדיפות שלו כמובן. אין בעיה שפעולה זו תיקח $O(n)$ זמן.
 - poll שמוציאה את האלמנט הראשון בתור (הכי עדיף) ומחזירה אותו.
 - forEach שבהינתן פונקציה (object function / lambda expression) היא תריץ את הפונקציה הזו על כל איברי התור, לאחר מכן היא לא תחזיר כלום.

דוגמת הקוד הבאה ממחישה כיצד ניתן להשתמש ב PriorityQueue.

בהינתן ה Object Function הבא בשם MyComparator שזו הגדרתו:

```
struct MyComparator{
    int operator()(int a, int b){
        return a - b;
    }
};
```

ניתן להשתמש ב PriorityQueue באופן הבא:

```
PriorityQueue<int, MyComparator> queue;
queue.push(5);
queue.push(1);
queue.push(3);
queue.push(2);
queue.push(4);
queue.push(6);
queue.forEach([](int i){cout << i << endl; }); // prints 1..6
cout << endl;
cout << queue.poll() << endl; // prints 1
cout << queue.poll() << endl; // prints 2
```

הסבר: MyComparator משווה שני int-ים ע"י החזרת הפרש שני הערכים שקיבל. יצרנו תור עדיפויות של int-ים עם משווה מסוג MyComparator, ולכן תור זה יעדיף ערכים נמוכים יותר על פני גבוהים. דחפנו את הערכים 6..1 שלא לפי סדר מוגדר כלשהו, אך ע"פ MyComparator הערכים הסתדרו לפי הסדר מהקטן לגדול. לכן, כאשר הפעלנו את forEach עם ביטוי למדה שבהינתן int מדפיס אותו, יודפסו כל הערכים 6..1 לפי הסדר. כמו כן, הדפסת הערך הראשון שנשלח מהתור תציג 1, ולאחר מכן 2 מפני שהשליפה מוציאה את האיבר הראשון מהתור.

המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

שאלה 2 פולימורפיזם, ירושה מרובה, קבצים (35 נק')

בקובץ Q2.h נתונה לכם המחלקה MusicalInstrument (כלי מוסיקאלי). למחלקה זו משתנה פרטי nomOfStrings מציין את מספר המיתרים שיש לכלי המוסיקלי, ומערך של שני תווים בשם id עם הרשאת .protected.

אין לשנות חתימות או להוסיף מתודות למחלקה זו.

(א) במחלקה זו עליכם לממש את המתודות הבאות בלבד:

- saveType – עליה לשמור את id בצורה בינארית בקובץ. שימו לב שהמתודה אינה וירטואלית ואין לשנות את חתימתה.
- save – עליה לשמור את numOfStrings, בצורה בינארית, ולהפעיל את המתודה האבסטרקטית saveMe.
- load – עליה לטעון את numOfStrings, בצורה בינארית, ולהפעיל את המתודה האבסטרקטית loadMe.

שימו לב שגם המתודות load ו save אינן וירטואליות. אלא הן מפעילות מתודות וירטואליות ואבסטרקטיות. אין לשנות את חתימתן או את אופן פעולתן זה.

(ב) בהמשך ממשו את המחלקה Guitar (גיטרה). גיטרה היא סוג שלי כלי מוסיקאלי בעל 6 מיתרים. המשתנה highestPitch צריך להיות מאותחל ל 329.63. ממשו בהתאמה את הבנאי ואת המתודות loadMe ו saveMe האחריות בהתאמה לשמירה ולטעינה של גיטרה מקובץ בינארי. זכרו שמתודות אלו מופעלות מהמתודות load ו save של MusicalInstrument.



(ג) באופן דומה ממשו את המחלקה הבאה Harp (נבל). נבל הוא כלי מוסיקאלי בעל 256 מיתרים. המשתנה deep הבוליאני צריך להיות מאותחל להיות "אמת".

(ד) המחלקה HarpGuitar מייצגת כלי מוסיקאלי שהוא גם גיטרה וגם נבל (ראו תמונה). לכלי זה יש 12 מיתרים. גם כאן עליכם לממש בהתאמה את המתודות loadMe & saveME – שימו לב שלא תהיה חזרה מיותרת של מידע.

(ה) כעת ממשו את הפונקציה הגלובלית countType כך

שבהינתן מערך של כלים מוסיקאליים וגודלו, היא תחזיר את מס' הכלים שהם בדיוק מהטיפוס המבוקש. למשל הפעלה הבאה תספור כמה איברים במערך הם בדיוק מהטיפוס גיטרה:

`countType<Guitar>(instruments,3)`

(ו) כעת ממשו את הפונקציה הגלובלית countTypesOf כך שבהינתן מערך של כלים מוסיקאליים וגודלו, היא תחזיר את מס' הכלים שהם מהסוג של הטיפוס המבוקש. למשל הפעלה הבאה תספור כמה איברים במערך הם מהסוג של הטיפוס גיטרה:

`countTypesOf<Guitar>(instruments,3)`

(ז) ממשו את הפונקציה הגלובלית saveInstruments כך שבהינתן אובייקט שמירה, מערך של כלים מוסיקאליים וגודלו, הפונקציה תשמור את איברי המערך בקובץ בינארי. בפרט, עליכם לחשוב קדימה כך שיהיה קל לטעון את הנתונים מהקובץ מאוחר יותר.

(ח) ממשו את הפונקציה הגלובלית loadInstruments כך שבהינתן אובייקט טעינה, פוינטר למערך לא מאותחל של כלים מוסיקאליים ומשתנה size שציין את גודלו, הפונקציה תשנה את size ותקצה גודל מתאים למערך, ואז תטען את נתוני הכלים מהקובץ הבינארי אל תוך המערך.

שימו לב שהמערך שנטען צריך להיות זהה לזה שנשמר – כלומר כל האיברים במערך שנטען ע"פ סוגם ומיקומם מתאימים למערך המקורי שנשמר. טיפ: היעזרו ב id וב saveType כדי לגלות את הטיפוסים.

המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

דוגמת קוד להפעלה:

```
MusicalInstrument** instruments = new MusicalInstrument*[3];
instruments[0] = new Guitar();
instruments[1] = new Harp();
instruments[2] = new HarpGuitar();

cout << instruments[0]->getNumOfStrings() << endl; // 6
cout << instruments[1]->getNumOfStrings() << endl; // 256
cout << instruments[2]->getNumOfStrings() << endl; // 12

cout << countType<Guitar>(instruments, 3) << endl; // 1
cout << countTypesOf<Guitar>(instruments, 3) << endl; // 2

ofstream out("database.db");
saveInstruments(out, instruments, 3);
out.close();

ifstream in("database.db");
int size2;
MusicalInstrument** instruments2;
loadInstruments(in, instruments2, size2);
in.close();

for (int i = 0; i<size2; i++)
if (instruments2[i]->getNumOfStrings() != instruments[i]->getNumOfStrings())
cout << "something went wrong..." << endl;
// should not print anything...

for (int i = 0; i<3; i++){
    delete instruments[i];
    delete instruments2[i];
}

delete[] instruments;
delete[] instruments2;
// no segmentation faults...
```

הסבר: יצרנו מערך של 3 כלים מוסיקאליים, והכנסנו אליו גיטרה, נבל וגיטרה-נבל. ההדפסה של מס' המיתרים תניב 6, 256, ו 12 בהתאמה. במערך זה יש בדיוק אובייקט אחד מהטיפוס גיטרה, ושני אובייקטים מהסוג של גיטרה. שמרנו את המערך בקובץ database.db ולאחר מכן טענו מתוכו נתונים למערך חדש וגודל חדש בשמות instruments2, ו size2 בהתאמה. שימו לב שהפונקציה loadInstruments אתחלה את instruments2, ו size2. לכל משתנה במקום ה i בשני המערכים צריך להיות מן הסתם מספר זהה של מיתרים (לכן לא צריך להיות מודפס דבר). לבסוף נשחרר את שני המערכים ונוודא שאין segmentation fault מסיבה כלשהי.

המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

שאלה 3 אופרטורים (35 נק'):

בקובץ Q3.h נתונה לכם המחלקה Delegate. עליכם לממש את המחלקה Event כך ש:

- היא תחזיק רשימה של מצביעים לאובייקטים מסוג Delegate
- האופרטור += יוסיף לרשימה מצביע לאובייקט מסוג Delegate
- האופרטור () יפעיל את האופרטור () לכל אובייקט ה delegate שברשימתו.

כך יהיה ניתן לצרף כמה פונקציות שנרצה למשתנה מסוג Event, וכשנפעיל אותו למעשה נפעיל את כל הפונקציות האלו אחת אחרי השניה.

דוגמא להפעלה:

```
class Action1 :public Delegate{
public:
    virtual void operator()(){
        cout << "Action1" << endl;
    };
};

class Action2 :public Delegate{
public:
    virtual void operator()(){
        cout << "Action2" << endl;
    };
};

int main()
{
    Event e;
    Action1* a1 = new Action1();
    Action2* a2 = new Action2();
    e += a1;
    e += a2;
    e += a1;
    e();
    /* output:
    Action1
    Action2
    Action1
    */

    delete a1;
    delete a2;
    return 0;
}
```

המחלקות Action1, Action2 ירשו את Delegate ומימשו את האופרטור () בדרכים שונות. ב main, למשתנה e Event צרפנו באמצעות += מצביעים לאובייקטים מסוג Action1, Action2. הפעלת האופרטור () של e גרמה להפעלת האופרטור () של האובייקטים שצורפו ולכן קבלנו את הפלט שמופיע בהערות.

יש להגיש ע"פ הוראות ההגשה במדויק ככתוב במסמך הוראות ההגשה (שפרסמתי לכם במודול, ומצורף לקובץ ה ZIP שהורדתם)

בהצלחה!