

המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

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

מבחן בקורס: השלמות תכנות מונחה עצמים

25.07.18 תאריך הבחינה:

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

משך הבחינה: 5 שעות

שם המרצה/ים:	
ד"ר אליהו חלסצ'י	
שם המתרגל/ים:	

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.

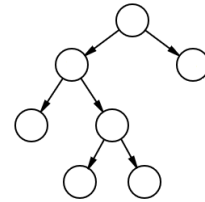
בהצלחה!!

המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

שאלה 1 תכנות גנרי, object functions (35 נק'):

עץ בינארי הוא מבנה נתונים בו לכל קודקוד ייתכנו לכל היותר שני בנים (ראו את התרשים הבא לדוגמא).

בקובץ Q1.h נתונה לכם המחלקה BinTree המהווה עץ בינארי של מספרים כלשהם (למשל מסוג int או float וכו'). בתוכה ישנה מחלקת עזר פרטית בשם Item. שימו לבנאי ולאופן ההריסה שלה.



ל BinTree יש רק שני data members: מצביע ל Item המהווה את שורש העץ (הקודקוד העליון) ומשתנה size עבור כמות האיברים. שימו לב לבנאי ולאופן ההריסה של BinTree. עד כאן אין לגעת בקוד הנתון. כעת עליכם לערוך 3 מתודות ע"פ ההגדרות הבאות:

- המתודה insert – בהינתן מספר היא תכניס אותו לעץ.
 - אם העץ ריק (קיים רק קודקוד השורש ללא מספר בתוכו) אז המס' ייכנס לקודקוד השורש.
 - אחרת, נטייל מהשורש לכיוון מטה באופן הבא:
 - אם המספר קטן מערך הקודקוד הנוכחי אז נרד לבן השמאלי
 - אם המספר גדול מערך הקודקוד הנוכחי אז נרד לבן הימני
 - נמשיך לטייל כך מטה כל עוד יש לאן. לבסוף, נכניס קודקוד חדש עם המספר המבוקש במקום המתאים (כבן שמאלי או ימני בהתאמה ללוגיקה לעיל).
 - הערה: אין צורך לאזן את העץ בעת הכנסה, אלא פשוט לממש את הלוגיקה לעיל.
 - ראו דוגמת הפעלה בהמשך.
- המתודה deleteSubTree – בהינתן מספר, היא תמחק מהעץ את כל תת-העץ המתחיל מהקודקוד שמכיל את המספר.
 - אם לא קיים קודקוד שמכיל מספר זה לא יקרה דבר
 - אם כן קיים קודקוד מתאים אז עליו ועל כל הקודקודים שתחתיו להימחק מהעץ והשתחרר מהזיכרון.
 - ראו דוגמת הפעלה בהמשך
- המתודה forEach - בהינתן פונקציה כפרמטר (object function) עליה להפעיל אותה על כל הקודקודים בעץ. כלומר על כל המספרים שקודקודים אלו מכילים.
 - ראו דוגמת הפעלה בהמשך

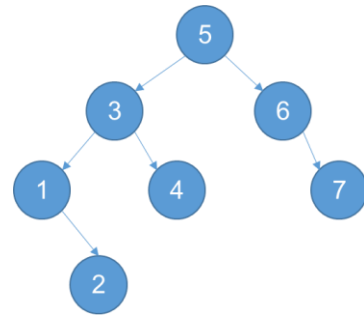
מותר להוסיף מתודות עזר במחלקה BinTree. שימו לב שבבדיקה אנו מפעילים את 3 המתודות לעיל ולא את מתודות העזר שלכם. אין לכתוב קוד באף מקום אחר פרט למקום המסומן בקוד. בהמשך המחלקה ישנן מתודות עזר להדפסה כדי שיהיה לכם קל לדבאג את הקוד. אין לערוך או למחוק קוד זה.

דוגמת הפעלה:

```
BinTree<int> tree;
tree.insert(5);
tree.insert(3);
tree.insert(6);
tree.insert(1);
tree.insert(2);
tree.insert(4);
tree.insert(7);
tree.print(cout);
tree.forEach(SQR());
tree.print(cout);
tree.deleteSubTree(35); // 35 does not exist. Nothing happens.
tree.deleteSubTree(36);
tree.print(cout);
```

המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

יצרנו עץ בינארי של int-ים. הכנסנו ערכים והדפסנו את העץ. מבחינה לוגית העץ צריך להיראות כך:



פלט ההדפסה (ע"פ פונקציית העזר הנתונה לכם) יראה כך:

```

5
3 6
1 4 7
2
  
```

טיפ: בעת דיבאג מומלץ להדפיס את העץ לאחר כל הכנסה.

לאחר מכן הפעלנו את פונקציית ה foreach ונתנו לה אובייקט קבוע, זמני וחסר שם מהסוג של SQR המגודר כך:

```

struct SQR{
    template<class Num>
    void operator() (Num& num) {
        num=num*num;
    }
};
  
```

פלט ההדפסה יראה כך (כל איבר עלה בריבוע):

```

25
9 36
1 16 49
4
  
```

לאחר בקשה למחוק את 35 לא קרא דבר, שכן ערך כזה לא קיים. אולם, לאחר בקשה למחוק את תת-העץ של 36, אז 36 ובנו הימני 49 נמחקו מהעץ ומהזיכרון. ולכן נישאר עם הפלט הבא:

```

25
9
1 16
4
  
```

התוכנית תסתיים ללא שגיאות ריצה כשכל האיברים השתחררו.

ה mainTrain (מוד האימון הנתון לכם) מבצע בדיקות דומות כאשר הדפסות קורות רק אם יש הודעות אודות שגיאות לוגיות.

המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

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

בקובץ Q2.h נתונה לכם המחלקה Dog (כלב). למחלקה זו משתנה פרטי id המציין את מספר הזהות של הכלב, ו weight המציין את משקלו. בנוסף ניתן לראות מתודת הדפסה print() שתלויה במתודה האבסטרקטית printAdditionalFeatures().

אין לשנות חתימות או להוסיף מתודות למחלקה זו או לשנות בה דבר מה בכל צורה.

עליכם לממש את המחלקות הבאות:

- Bulldog כלב בולדוג בעל התכונה הנוספת האם הוא חברותי או לא (friendly)
 - ערך דיפולטיבי לחברותי הוא אמת
- Poodle כלב פודל בעל התכונה הנוספת האם הוא חכם או לא (intelligent)
 - ערך דיפולטיבי לחכם הוא אמת
- BulldogPoodle כלב כלאיים (דמיוני) בין בולדוג לפודל, ובעל כל התכונות של שניהם.

בנוסף עליכם לממש את שתי הפונקציות הגלובליות הבאות:

- saveDogs - בהינתן מערך של כלבים, גודלו, ואובייקט ofstream כל פרטי הכלבים שבמערך ישמרו לקובץ. אתם מחליטים על הפורמט. זכרו לחשוב קדימה עבור הטעינה מהקובץ.
- loadDogs – בהינתן מצביע לא מאותחל למערך של (מצביעים ל) כלבים, וגודל לא מאותחל, ואובייקט ifstream, נתוני הכלבים ייקראו מהקובץ ויאותרלו את המערך ואת גודלו.
 - אתם קובעים את הפורמט. הוא צריך להיות עקבי עם צורת השמירה בקובץ.

תזכורת:

במידה ותרצו להשתמש ב typeid אז זכרו של typeid(x).name() ייתכן פלט שונה בין visual studio לבין לינוקס (מערכת הבדיקות). הסטנדרט הוא אורך השם ואז השם עצמו. לדוגמא אם היתה לנו מחלקה בשם Eli, x יהיה משתנה מסוג Eli, אז הפלט בלינוקס הוא "Eli3".

על הקוד שלכם להיות קונסיסטנטי עם הדוגמא הבאה והפלט שלה (תגרמו לה לעבוד כמו שצריך).

```
Dog** dogs = new Dog*[5];
dogs[0] = new Bulldog(123, 20);
dogs[1] = new Bulldog(1234, 20, false);
dogs[2] = new Poodle(321, 10, false);
dogs[3] = new Poodle(4321, 10);
BulldogPoodle bp(Bulldog(111, 30), Poodle(123, 40));
dogs[4] = &bp;

for (int i = 0; i<5; i++){
    dogs[i]->print(cout);
    cout << endl;
}

ofstream out("output.txt");
saveDogs(dogs, 5, out);
out.close();

cout << "-----" << endl;
```

המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

```
ifstream in("output.txt");
Dog** dogs2;
int size;
loadDogs(dogs2, size, in);
in.close();

for (int i = 0; i<size; i++){
    dogs2[i]->print(cout);
    cout << endl;
}

for (int i = 0; i<4; i++){
    delete dogs[i];
}
delete[] dogs;

for (int i = 0; i<size; i++){
    delete dogs2[i];
}
delete[] dogs2;
```

ניתן לראות שיצרנו מערך פולימורפי של כלבים ושמו בו כלבים מכל הצורות. שימו לב להפעלות השונות של הבנאים. הדפסנו את איברי המערך. שמרנו אותם בקובץ output.txt באמצעות saveDogs. טענו אותם מתוך הקובץ לתוך מערך חדש בשם dogs2 ובגודל size. הדפסנו את איברי מערך החדש וקבלנו את אותו הפלט. שימו לב שאין הדפסות כפולות עבור ה BulldogPoodle.

הפלט:

Dogs:	Dogs2:
ID: 123 Weight: 20 Friendly 1	ID: 123 Weight: 20 Friendly 1
ID: 1234 Weight: 20 Friendly 0	ID: 1234 Weight: 20 Friendly 0
ID: 321 Weight: 10 Intelligent 0	ID: 321 Weight: 10 Intelligent 0
ID: 4321 Weight: 10 Intelligent 1	ID: 4321 Weight: 10 Intelligent 1
ID: 111 Weight: 30 Friendly 1 Intelligent 1	ID: 111 Weight: 30 Friendly 1 Intelligent 1

אופי הבדיקה דומה לקוד לעיל. הפורמט בקובץ אינו משנה. העיקר שנוכל ליצור אובייקטים בהתאם לדוגמא לעיל, לשמור אותם ולקבל בטעינה את אותם האובייקטים בדיוק (לפי טיפוס וערכים).

המסלול האקדמי המכללה למנהל ביה"ס למדעי המחשב

שאלה 3 אופרטורים (30 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!