

מבחן מגן 1 - מבנה נתונים ותכנות מונחה עצמים

מגמת תוכנה, שנה ג

משך המבחן: 4 שעות

חומר עזר מותר

- מחשבון שאינו ניתן לתכנות
- קלסר אחד

המרצה: אמיר דגן 054-4887554

מבנה המבחן

חלק א - 4 שאלות יש לפתור, 3 משקל כל שאלה 16 נקודות סה"כ 48 נקודות
חלק ב - 4 שאלות, יש לפתור, 2 משקל כל שאלה 18 נקודות סה"כ 36 נקודות
חלק ג - 2 שאלות, יש לפתור 1 שאלות משקל השאלה 16 נקודות

בסוף המבחן נמצאים ממשקי מבני הנתונים

חלק א

יש לפתור 3 שאלות מתוך השאלות הבאות. משקל כל שאלה הוא 16 נקודות

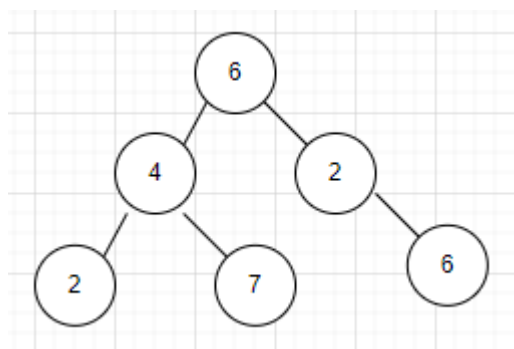
שאלה 1 (16 נקודות)

נא לכתוב פעולה המקבלת את השורש של עץ בינארי, ומחזירה את סכום כל העלים שערכם גדול מ-5

שימו לב: אין לכלול בסיכום ערכים של צמתים שאינם עלה

הערה: מותר להשתמש בפונקציות עזר, אך יש לכתוב את תוכן אם עושים זאת

למשל עבור העץ



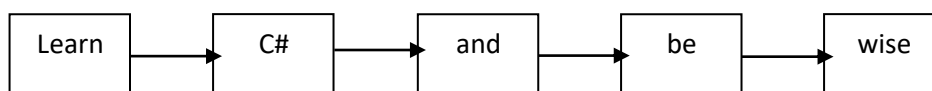
התוצאה תהיה $13 = 6+7$

שאלה 2 (16 נקודות)

נא לכתוב פעולה המקבלת שרשרת של מחרוזות, ומחזירה מחסנית המכילה את כל המחרוזות שיש להם מספר זוגי של תווים.

סדר המחרוזות במחסנית יהיה כזה שכאשר נעשה POP למחסנית, נקבל את המחרוזת הקרובה ביותר אל תחילת השרשרת

כלומר עבור השרשרת



נקבל את המחסנית

C#
be
wise

שאלה 3 (16 נקודות)

נתון הגדרת העצמים הבאים

```
1. interface Ix {
2.     void sod();
3. }
4. public class A:Ix {
5.     protected int q;
6.     public A() { q = 1; }
7.     public A(int q) { this.q = q; }
8.     public virtual void foo(int n) {
9.         Console.WriteLine("A.foo "+n);
10.    }
11.    public void sod() {
12.        Console.WriteLine("A.sod");
13.        foo(5);
14.    }
15. }
16.
17. public class B : A {
18.     public B() : base() { }
19.     public B(int q) : base(q * 2) { }
20.     public override void foo(int n) {
21.         Console.WriteLine("B.foo " + n);
22.     }
23. }
```

- א. האם יש שורות שיגרמו לשגיאות קומפילציה? מה מספרי שורות אלה? איך ניתן לתקן את השגיאה?
ב. האם יש שורות שמהוות העמסת פונקציות? מה מספרי שורות אלה?
ג. האם יש שורות שמהוות דריסת פונקציות? מה מספרי שורות אלה?

הוגדרו המופעים

```
A aa = new A();
A ab = new B();
Ix xb = new B();
```

נא לציין עבור כל אחת מהשורות הבאות מה יודפס. אם השורה גורמת לשגיאה, יש לציין זאת תוך ציון סוג השגיאה (ריצה או קומפילציה)

```
ד. aa.sod();
ה. ab.sod();
ו. xb.sod();
ז. xb.foo();
```

שאלה 4 (16 נקודות)

נתונות הגדרות האובייקטים הבאים:

```
class D {
    private A[] x;
}
class A {
    protected int a;
    public A(int b=5) {
        a = b;
        Console.WriteLine("A::"+a);
    }
    public virtual void Sod() {
        Console.WriteLine("sh sh sh .....");
    }
}

class B : A {
    public B() : base(7) {
        Console.WriteLine("B::"+a);
    }
    public override void Sod() {
        base.Sod();
        Console.WriteLine("B...B");
    }
}

class C : A {
    public C(int b=15):base(b) {
        Console.WriteLine("C::"+a);
    }
    public void Sod() {
        base.Sod();
        Console.WriteLine("C...C");
    }
}
```

1. A ad = new D();
2. A aa = new A(36);
3. A ab = new B();
4. B bb1 = new B();
5. B bb2 = new B(66);
6. A ac = new C();
7. C cc = new C();
8. ((C)bb1).Sod();
9. ab.Sod();
10. ac.Sod();
11. ((C)ac).Sod();

5) נקודות) א. נא לשרטט דיאגרמת UML עבור מערך אובייקטים זה

11) נקודות) ב. עבור כל אחת משורות 1-11 נא לציין

אם השורה גורמת שגיאת קומפילציה (יש לציין את מקור השגיאה)

אם השורה גורמת שגיאת זמן ריצה (יש לציין את מקור השגיאה)

אם השורה תקינה – מה תהיה ההדפסה למסך הנובעת ממנה (אם אין הדפסה יש לכתוב "אין")

חלק ב

יש לפתור 2 שאלות מתוך השאלות הבאות. משקל כל שאלה הוא 18 נקודות

שאלה 5 (18 נקודות)

נתון האובייקט

```
class A {
    protected int a;
    public A(int b=5) {
        a = b;
        Console.WriteLine("A::"+a);
    }
    public virtual void Sod() {
        Console.WriteLine("sh sh sh .....");
    }
    public virtual int GetA() {
        return a;
    }
}
```

ובנוסף נתון האובייקט

```
class D {
    private Node<A> x;
}
```

א. נא לכתוב מתודה בתוך אובייקט D אשר מחזירה את הערך a הגדול ביותר שיש בתוך השרשרת x

יוצרים את האובייקט

```
class F {
    private Stack<D> dd;
    public void SortMe() { }
    public void AddItem(D d) { }

    public F(Stack<D> d) {
        dd = d;
        SortMe();
    }
}
```

ב. נא לממש את הפעולה SortMe אשר לאחר הפעלתה מחסנית dd ממוינת לפי הערך החוזר מהפעולה של סעיף א

הערך הגדול ביותר יהיה הראשון שיתקבל מפעולת POP

ג. נא לממש את פעולה AddItem אשר מוסיפה את האיבר d אל המחסנית כך שהמחסנית נשארת ממוינת

ד. נא לציין מה הסיבוכיות של פתרוןך בשאלה ג

שאלה 6 (18 נקודות)

נתון האובייקט

```
class Student {
    private string id;
    private int grade;
    private int ClassNum;
    public string GetId() { return id; }
    public int GetGrade() { return grade; }
    public int GetClassNum() { return ClassNum; }
}
```

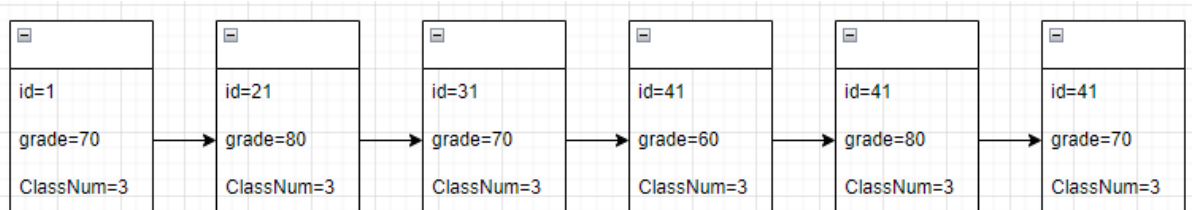
וכן נתון האובייקט

```
class GradeStat {
    private int Grade;
    private int Dist;
    public int GetGrade() { return Grade; }
    public void SetGrade(int Grade) { this.Grade = Grade; }
    public int GetDist() { return Dist; }
    public void SetDist(int dist) { this.Dist = dist; }
}
```

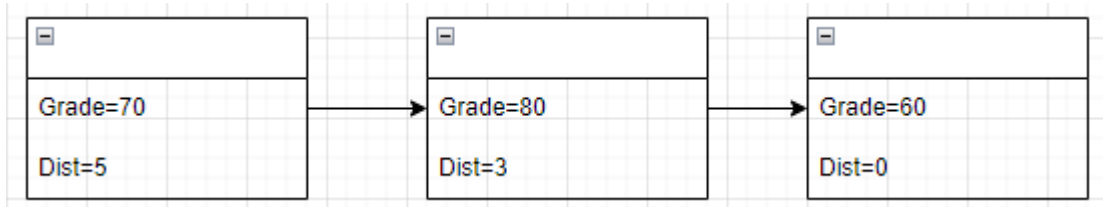
נתונה שרשרת חוליות של ציוני סטודנטים

- א. נא לכתוב פעולה, המקבלת חוליה בשרשרת, ומחזירה את המרחק של ציון החוליה כלומר, כמה חוליות יש עד המופע הבא של אותו ציון אם הציון לא מופיע בהמשך השרשרת – הפעולה תחזיר 0 אם יש מספר מופעים של הציון בהמשך השרשרת – הפעולה תחזיר את המרחק הגדול ביותר כלומר עד המופע האחרון של הציון
- ב. נא לכתוב פעולה המקבלת חוליה ראשונה בשרשרת ציונים, ומחזירה חוליה ראשונה לשרשרת של GradeStat
- בשרשרת החדשה עבור כל אחד מהציונים שיש בשרשרת המקורית, יהיה אובייקט שיכיל את המרחק המקסימלי שיש לו בשרשרת

לדוגמה עבור השרשרת



נקבל את השרשרת



```
class A {
    protected int a;
    public A(int b=5) { a = b; }
    public virtual int GetA() { return a; }
}

class B : A {
    public B() : base(7) {}
}

class C : A {
    public C(int b=15):base(b) {}
    public int GetA() { return 77; }
}

class D {
    private int a;
    public D(int b=5) { a = b;}
    public int GetA() { return a;}
}
```

וכן הוגדר גם האובייקט

```
class E {
    private Stack<A> st;
    public E(Stack<A> st) {
        this.st = st;
    }
    public int Sod() {
        int r = 0;
        A t = st.Pop();
        r = t.GetA();
        if (!st.IsEmpty()) {
            int x = Sod();
            if (x > r)
                r = x;
        }
        st.Push(t);
        return r;
    }
}
```


בתוכנית הראשית נכתבו השורות הבאות

```
1. Stack<A> ee = new Stack<A>();
2. ee.Push(new A());
3. ee.Push(new A(32));
4. ee.Push(new B());
5. ee.Push(new B(12));
6. ee.Push(new C());
7. ee.Push(new C(9));
8. ee.Push(new D());
9.
10. E e1 = new E(ee);
11. Console.WriteLine(e1.Sod());
```

(5 נקודות) א. אילו מבין השורות 1-10 יוצרת שגיאת קומפילציה או ריצה? נא לציין איזו שגיאה

(10 נקודות) ב. לאחר מחיקת השורות הנזכרות בסעיף א, מריצים את שורה 11
מה יוצג על המסך? חובה לעשות טבלת מעקב

(3 נקודות) ג. לאחר הרצת שורה 11, מריצים אותה פעם נוספת. מה יוצג הפעם על המסך?

שאלה 8 (18 נקודות)

נתונות הגדרות העצמים הבאים:

```
1. class F {
2.     protected int x;
3.     public F(int x) {
4.         Console.WriteLine("in F with " + x);
5.         this.x = x;
6.     }
7.     public void Sod() {
8.         Console.WriteLine("F.sod");
9.     }
10.    public virtual void Why() {
11.        Console.WriteLine("F.why");
12.    }
13.    public int GetX() { return x; }
14. }
15. class G : F {
16.     public G(int x) : base(x) {
17.         Console.WriteLine("in G with " + x);
18.     }
19.     public override void Why() {
20.         Console.WriteLine("G.why");
21.     }
22.     public bool Equals(F ff) {
23.         if (ff.GetX() == this.GetX())
24.             return true;
25.         return false;
26.     }
27. }
28. class H : F {
29.     private F fg;
30.     public H() : base(5) {
31.         Console.WriteLine("in H ");
32.         fg = new G(7);
33.     }
34.     public void Sod(int x) {
35.         Console.WriteLine("H.sod " + x);
36.     }
37.     public void Why(string s = "") {
38.         Console.WriteLine("H.why");
39.     }
40.     public string ToString(string pre) {
41.         return $"{pre} H";
42.     }
43.     public int GetX() { return fg.GetX(); }
44. }
```

(כל סעיף – 2 נקודות)

- א. האם יש שורות שיגרמו לשגיאות קומפילציה? מה מספרי שורות אלה? איך ניתן לתקן את השגיאה?
- ב. האם יש שורות שמהוות העמסת פונקציות? מה מספרי שורות אלה?
- ג. האם יש שורות שמהוות דריסת פונקציות? מה מספרי שורות אלה?

(המשך השאלה בעמוד הבא)

בכל אחד מהסעיפים הבאים יש לרשום מה הפקודה תדפיס.

אם הפקודה יוצרת שגיאת זמן ריצה, יש לציין זאת

לתשומת לבך – באם ציינת שגיאות קומפילציה בסעיף א, יש להניח שבוצע התיקון שלהן לשם הרצת הפקודות הבאות

(כל סעיף – 2 נקודות)

```
א. F fg = new G(9);  
ב. F fh = new H();  
ג. fg.Why();  
ד. fh.Why();  
ה. fh.Sod(3);  
ו. fg.GetX();
```

חלק ג

יש לענות על שאלה אחת

שאלה 9 (16 נקודות)

בתוכנה עבור בנק, נדרשים לטפל במספר סוגי בקשות.

לכל הבקשות יהיה מספר מזהה של הבקשה, מספר החשבון הפועל, תאריך הפעולה, והסכום הכספי של הפעולה

פעולת קבלת כסף מסתפקת בנתונים אלה (כלומר אין נתונים נוספים עבור סוג פעולה זה)

פעולת העברת כסף תכלול גם את מספר חשבון היעד

פעולת רכישת מניות תכלול גם את כמות המניות הנרכשת, ושער הרכישה

(5 נקודות) א. נא ליצור הגדרה של כל האובייקטים הנדרשים לשם פעולת התוכנה
יש לשרטט תרשים UML עבור הגדרה זו

לשם פעולה יעילה יותר, את כל הבקשות המתקבלות שומרים בהתחלה במבנה מסוג תור, כך שניתן יהיה לטפל בבקשות לפי סדר הגעתן

(3 נקודות) ב. נא לכתוב את הגדרת מבנה הנתונים אשר ישמש לקליטת הבקשות, כמו גם פונקציה שבעזרתה קולטים בקשה ומוסיפים אותה אל מבנה הנתונים הזה

עם סיום הטיפול בבקשה, מוציאים אותן מהתור, ומעבירים אותן לארכיון:

הארכיון מסודר לפי ימים, כאשר היום הוא הראשון בארכיון ולאחריו יהיה אתמול

הארכיון מרכז את כל הבקשות של אותו יום ביחד. סדר הבקשות של אותו יום אינו משנה.

(2 נקודות) ג. נא לכתוב את הגדרת מבני הנתונים אשר ישמשו לארכיון (האחד לכל יום והשני לכלל הארכיון)

(6 נקודות) ד. נא לכתוב קוד אשר מוסיף בקשה שהגיעה אתמול אל הארכיון (לא תהיה פגיעה בנתוני שאר הימים אשר בארכיון)

שאלה 10 (16 נקודות)

נתונים האובייקטים הבאים

```
class M {
    protected char c;
    public virtual string StringMe() { return c.ToString(); }
}
class N : M {
    public string StringMe(int x) { return x.ToString(); }
}
class L : M {
    public override string StringMe() { return $"{c}::{c}"; }
}
class K : L {
    public string Sod() { return "shshshsh"; }
}
```

- א. נא לכתוב הגדרה של מחסנית היכולה להכיל שילוב של כל העצמים הנ"ל ביחד
- ב. נא לכתוב פעולה המקבלת מחסנית כמו בסעיף א, ומדפיסה את ערך StringMe לכל אחד מהאיברים שניתן להפעיל בו פונקציה זו
- ג. נא לכתוב מה יודפס בסעיף ב עבור כל אחד מסוגי האובייקטים K,L,N,M
- ד. נא לכתוב פעולה המקבלת מחסנית כמו בסעיף א, ומשנה אותה כך שתכיל רק אלמנטים שניתן להריץ בהם פעולת Sod
יש לוודא שסדר האלמנטים לא משתנה

נספח לשאלון – 97105 מבני נתונים ותכנות מונחה עצמים –C#

ממשק המחלקה חוליה הגרית – $\text{Node}<T>$

המחלקה מגדירה חוליה גרית שבה ערך מטיפוס T והפניה לחוליה העוקבת.

Node (T x)	הפעולה בונה חוליה. הערך של החוליה הוא x , ואין לה חוליה עוקבת.
Node (T x, Node<T> next)	הפעולה בונה חוליה. הערך של החוליה הוא x , והחוליה העוקבת לה היא $next$. ערכו של $next$ יכול להיות null .
T GetValue()	הפעולה מחזירה את הערך של החוליה.
Node<T> GetNext()	הפעולה מחזירה את החוליה העוקבת. אם אין חוליה עוקבת, הפעולה מחזירה null .
void SetValue (T x)	הפעולה משנה את הערך השמור בחוליה ל- x .
bool HasNext()	הפעולה מחזירה true אם יש חוליה נוספת
void SetNext (Node<T> next)	הפעולה משנה את החוליה העוקבת ל- $next$. ערכו של $next$ יכול להיות null .
override string ToString()	הפעולה מחזירה מחרוזת המתארת את החוליה.

יעילות הפעולות : כל הפעולות מתבצעות בסדר גודל קבוע, $O(1)$.

ממשק המחלקה הגרית – מחסנית $\text{Stack}<T>$

המחלקה מגדירה טיפוס אוסף בעל פרוטוקול **LIFO** להכנסה והוצאה של ערכים.

Stack()	הפעולה בונה מחסנית ריקה.
bool IsEmpty()	הפעולה מחזירה "אמת" אם המחסנית הנוכחית ריקה, "שקר" אם היא אינה ריקה.
void Push (T x)	הפעולה מכניסה את הערך x לראש המחסנית הנוכחית (דחיפה).
T Pop()	הפעולה מוציאה את הערך שבראש המחסנית הנוכחית ומחזירה אותו (שליפה). הנחה : המחסנית הנוכחית אינה ריקה.
T Top()	הפעולה מחזירה את הערך שבראש המחסנית הנוכחית בלי להוציאו. הנחה : המחסנית הנוכחית אינה ריקה.
override string ToString()	הפעולה מחזירה תיאור של המחסנית, כסדרה של ערכים, במבנה הזה (x_1 הוא האיבר שבראש המחסנית) : $[x_1, x_2, \dots, x_n]$

יעילות הפעולות – מחלקה מיוצגת בעזרת שרשרת חוליות.

כל הפעולות מתבצעות בסדר גודל קבוע, $O(1)$, למעט הפעולה `ToString ()` המתבצעת בסדר גודל לינארי.

ממשק המחלקה הגנרית - תור $Queue<T>$

המחלקה מגדירה טיפוס אוסף עם פרוטוקול FIFO להכנסה והוצאה של ערכים.

Queue ()	הפעולה בונה תור ריק.
bool IsEmpty ()	הפעולה מחזירה "אמת" אם התור הנוכחי ריק, ו"שקר" אם הוא אינו ריק.
void Insert (Tx)	הפעולה מכניסה את הערך x לסוף התור הנוכחי.
T Remove ()	הפעולה מוציאה את הערך שבראש התור הנוכחי ומחזירה אותו. הנחה: התור הנוכחי אינו ריק.
T Head ()	הפעולה מחזירה את ערכו של האיבר שבראש התור מבלי להוציאו. הנחה: התור הנוכחי אינו ריק.
override string ToString ()	הפעולה מחזירה מחרוזת המתארת את התור כסדרה של ערכים, במבנה הזה (x הוא האיבר שבראש התור): $[x_1, x_2, \dots, x_n]$

יעילות הפעולות- המחלקה מיוצגת בעזרת שרשרת חוליות והפניה לזנב התור.

כל הפעולות מתבצעות בסדר גודל קבוע, $O(1)$, למעט הפעולה $ToString()$ המתבצעת בסדר גודל לינארי.

ממשק המחלקה חוליה בינרית $BinNode <T>$

המחלקה מגדירה חוליה בינרית שבה ערך מטיפוס T ושתי הפניות לחוליות בינריות.

BinNode (T x)	הפעולה בונה חוליה בינרית. ערך החוליה הוא x וערך שתי ההפניות שלה הוא null
BinNode (BinNode<T> left, T x, BinNode<T> right)	הפעולה בונה חוליה בינרית שערכה יהיה x, left ו-right הן הפניות אל הילד השמאלי והימני שלה. ערכי ההפניות יכולים להיות null
T GetValue()	הפעולה מחזירה את הערך של החוליה
void SetValue (T x)	הפעולה משנה את הערך השמור בחוליה ל-x
bool HasLeft()	הפעולה מחזירה true אם יש ילד שמאלי
bool HasRight()	הפעולה מחזירה true אם יש ימני
BinNode<T> GetLeft()	הפעולה מחזירה את הילד השמאלי של החוליה. אם אין ילד שמאלי הפעולה מחזירה null
BinNode<T> GetRight()	הפעולה מחזירה את הילד הימני של החוליה. אם אין ילד ימני הפעולה מחזירה null
void SetLeft (BinNode<T> left)	הפעולה מחליפה את הילד השמאלי בחוליה left
void SetRight (BinNode<T> right)	הפעולה מחליפה את הילד הימני בחוליה right
string ToString()	הפעולה מחזירה מחרוזת המתארת את הערך השמור בחוליה

יעילות הפעולות: כל הפעולות מתבצעות בסדר גודל קבוע, $O(1)$

עמוד 5 מתוך 5