GuyAI
מבית GOOL בגרויות

סוג הבחינה: בחינת גמר ממלכתית
מועד הבחינה: קיץ 2021 מועד ב'
סמל השאלון: 97105
שם התלמיד/ה: _____

מבנה נתונים ותכנות מונחה עצמים

הנדסת תוכנה • הנדסאים וטכנאים

הוראות

א. משך הבחינה: ארבע שעות וחצי.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שני מבחנים. **יש לענות על מבחן אחד בלבד בהתאם למוסד הלימודים:** מבחן ב- Java (עמוד 2) מבחן ב- C# (עמוד 17) בכל מבחן 11 שאלות. **חלק א' - 45 נקודות** שאלות 1-4: יש לענות על **שלוש** שאלות בלבד. ערך כל שאלה 15 נקודות. **חלק ב' - 30 נקודות** שאלות 5-8: יש לענות על **שתי** שאלות בלבד. ערך כל שאלה 15 נקודות. **חלק ג' - 25 נקודות** שאלות 9-11: יש לענות על **שתי** שאלות בלבד. ערך כל שאלה 12 נקודות. נקודה אחת תינתן על הערכה. **בסך הכול: 100 נקודות.**

ג. חומר עזר מותר בשימוש: 1. מחשבון (אין להשתמש במחשב כף יד או במחשבון עם תקשורת חיצונית). 2. קלסר **אחד בלבד** עם **חומר ההרצאות**. אין להוציא דפים מהקלסר. **אין לצרף ספרים או חוברות עם פתרונות.** לשאלון מצורפים: נספח ממשקים לבחינה JAVA, נספח ממשקים לבחינה C# (ממשקי המחלקות Node, Stack, Queue).

ד. הוראות מיוחדות: 1. יש לקרוא בעיון את ההנחיות בדף השער ואת כל שאלות הבחינה, ולוודא שהן מובנות. 2. את התשובות יש לכתוב בצורה מסודרת, בכתב יד ברור ונקי (גם בכך תלויה הערכת הבחינה). 3. יש להשאיר את העמוד הראשון במחברת הבחינה ריק. בסיום המבחן יש לרשום בעמוד זה את מספרי התשובות לבדיקה. התשובות ייבדקו לפי סדר כתיבתן בעמוד זה. לא ייבדקו תשובות עודפות. 4. יש לכתוב את התשובות במחברת הבחינה **בעט בלבד**, בכתב יד ברור. 5. יש להתחיל כל תשובה בעמוד חדש ולציין את מספר השאלה ואת הסעיף. אין צורך להעתיק את השאלה עצמה. 6. טיוטה יש לכתוב במחברת הבחינה בלבד. יש לרשום את המילה "טיוטה" בראש העמוד ולהעביר עליו קו כדי שלא ייבדק. 7. יש להציג פתרון מלא ומנומק, כולל חישובים לפי הצורך. הצגת תשובה סופית ללא שלבי הפתרון לא תזכה בניקוד. 8. יש להסביר בפירוט כל תוכנית שנכתבה, **תוכנית ללא הסבר מפורט לא תזכה בניקוד.** **חל איסור מוחלט להוציא שאלון או מחברת בחינה מחדר הבחינה!** בשאלון זה 31 עמודי בחינה ו- 4 עמודי נספחים.

השאלות בשאלון זה מנוסחות בלשון רבים, אף על פי כן על כל תלמידה וכל תלמיד להשיב עליהן באופן אישי.

בהצלחה!

השאלות

1. תור "מדרגה N" הוא תור של מספרים שלמים חיוביים שהמספר הראשון בו הוא 1 וכל מספר מופיע בו ברצף, מספר פעמים השווה לערכו, עד N. כלומר, 1 יופיע פעם אחת, לאחריו 2 יופיע 2 פעמים וכו'.

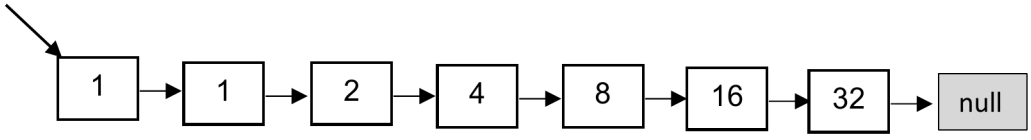
דוגמה לתור "מדרגה 4": 1,2,2,3,3,3,4,4,4,4.

א. כתבו פעולה שמקבלת מספר שלם חיובי N ומחזירה תור "מדרגה N".

ב. כתבו פעולה המקבלת תור q של מספרים שלמים ומספר שלם חיובי N ומחזירה true אם התור הוא תור "מדרגה N", ואם לא, הפעולה מחזירה false.

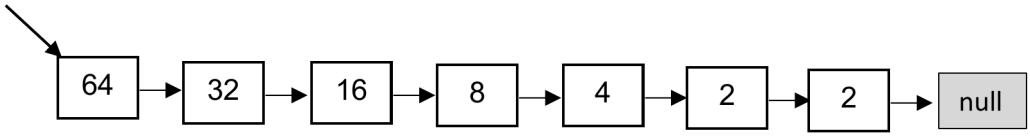
2. "**שרשרת סכומים**" היא שרשרת של מספרים שלמים שכל מספר בה (פרט לראשון) הוא סכום של המספרים אשר מופיעים לפניו.

דוגמה ל"שרשרת סכומים":



א. כתבו פעולה אשר מקבלת הפניה לחוליה ראשונה של שרשרת חוליות ומחזירה true אם השרשרת היא "**שרשרת סכומים**", ואם לא, הפעולה מחזירה false.

"**שרשרת סכומים הפוכה**" היא שרשרת של מספרים שלמים שכל מספר בה (פרט לאחרון) הוא סכום של המספרים אשר מופיעים אחריו.



ב. כתבו פעולה אשר מקבלת הפניה לחוליה ראשונה של שרשרת חוליות ומחזירה true אם השרשרת היא "**שרשרת סכומים הפוכה**", ואם לא, הפעולה מחזירה false.

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

3. נתונה התוכנית הראשית הבאה (הניחו כי כל השורות בקוד הן תקינות):

Java:

```
public static void main(String[] args)
{
    B b1 = new A();
    B b2 = new C();
    A a1 = (C) b2;
    D d1 = new C();
    B b3 = new B();
    D d2 = new A();
}
```

C#:

```
public static void Main(string[] args)
{
    B b1 = new A();
    B b2 = new C();
    A a1 = (C) b2;
    D d1 = new C();
    B b3 = new B();
    D d2 = new A();
}
```

א. שרטטו תרשים UML המתאר קשר אפשרי ונכון בין כל המחלקות המוזכרות בקוד.

ב. הניחו כי השורה האחרונה בקוד מבצעת המרה **לא תקינה** (המחשב מודיע על טעות). שרטטו תרשים UML **חדש** המתאר את הקשרים הנכונים במקרה זה, בין כל המחלקות המוזכרות בקוד.

4. נתונה המחלקה A הבאה:

Java:

```
public class A
{
 public int foo (int i)
 {
    return i;
 }
}
```

C#:

```
public class A
{
 public int Foo (int i)
 {
    return i;
 }
}
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

א. עבור אחת מהפעולות 1-5 הבאות קבעו מה יתרחש אם מוסיפים את הפעולה למחלקה A.

בררו את האפשרות המתאימה מאפשרויות הבאות:

- שגיאת קומפילציה.
- שגיאת זמן ריצה.
- קוד תקין, דריסה (overriding).
- קוד תקין, העמסה (overloading).

Java:

```
1. public int foo () { return 1; }
2. public int foo(double i){ return (int)i; }
3. public double foo(int i){ return i; }
4. public double foo(double i){ return i; }
5. public void foo(double i){ System.out.print(i); }
```

C#:

```
1. public int Foo () { return 1; }
2. public int Foo(double i){ return (int)i; }
3. public double Foo(int i){ return i; }
4. public double Foo(double i){ return i; }
5. public void Foo(double i){ Console.Write (i); }
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

ב-C# בלבד: חתימה של הפעולה Foo השתנתה:

```
public virtual int Foo (int i)
```

ב. עבור אחת מהפעולות 1-5 הבאות קבעו מה יתרחש אם מוסיפים את הפעולה למחלקה B שיורשת מ- A.

בררו את האפשרות המתאימה מאפשרויות הבאות:

- שגיאת קומפילציה.
- שגיאת זמן ריצה.
- קוד תקין, דריסה (overriding).
- קוד תקין, העמסה (overloading).

Java:

```
1. public int foo () { return 10; }
2. public int foo(int i){ return i*10; }
3. public double foo(int i){ return i*10; }
4. public double foo(double i){ return i*10; }
5. public void foo(double i){ System.out.print(i); }
```

C#:

```
1. public int Foo () { return 10; }
2. public override int Foo(int i){ return i*10; }
3. public override double Foo(int i){ return i*10; }
4. public double Foo(double i){ return i*10; }
5. public override void Foo(double i){ Console.Write(i); }
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

5. נתונה מחלקת Child

Java:

```
public class Child extends Parent
{
    private int y;

public Child( String name, double x)
  {
    super(x, name);
    this.y = 10;
    System.out.println( this.x );
  }
public Child(int y)
  {
    super(6.5,"David","Cohen");
    this.y = y;
  }
public int sum()
  {
    System.out.println(this.name+" "+this.getLastName());
    return this.y + this.method();
  }
public double calculate(double d)
  {
    return d * super.calculate(5.2);
  }
public void print()
  {
    super.print();
    System.out.println("Child");
  }
 public int getYear()
  {
    return Parent.year;
  }
}
```

C#:

```
class Child: Parent
{
   private int y;
```

```
  public Child(string name, double x) : base(x, name)
  {
    this.y = 10;
    Console.WriteLine(this.x);
  }
  public Child(int y) : base(6.5, "David", "Cohen")
  {
    this.y = y;
  }
  public int Sum()
  {
    Console.WriteLine(this.name+" " + this.GetLastName());
    return this.y + this.Method();
  }
  public override double Calculate(double d)
  {
    return d * base.Calculate(5.2);
  }
  public override void Print()
  {
    base.Print();
    Console.WriteLine("Child");
  }

  public int GetYear()
  {
    return Parent.year;
  }
}
```

א. כתבו את מחלקת העל - Parent (את המחלקה המורישה) של המחלקת Child.
מחלקת העל צריכה להכיל את כל התכונות והפעולות המתחייבות ממחלקת Child.

אין צורך לממש את הפעולות של מחלקת העל ואין לשנות את מחלקת Child.

ב. היכן בשתי המחלקות ישנה העמסה? הסבירו את תשובתכם.

ג. היכן בשתי המחלקות ישנה דריסה? הסבירו את תשובתכם.

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

ד. ממשו במחלקה חיצונית את הפעולה:

Java:

```
public static void output(Object[] arr)
```

C#:

```
public static void Output(Object[] arr)
```

הפעולה מקבלת מערך של עצמים. עבור כל עצם מסוג Parent או Child במערך, הפעולה תזמן את פעולת print() המתאימה לסוג העצם.

6. מטבע חוץ ForeignCurrency מאופיין באמצעות שם המטבע (למשל דולר, יורו, פאונד וכו') ורשימה של שווי המטבע בשקלים (זהו שער המטבע) במשך תקופה מסוימת. לכל יום בתקופה יש שער משלו. האיבר בראש הרשימה מציין את שער היום הנוכחי.

להלן ייצוג המחלקה:

Java:

```
class ForeignCurrency
{
  private String name;
  private Node<Double> rates;
```

C#:

```
class ForeignCurrency
{
  private string name;
  private Node<double> rates;
```

במחלקה הוגדרו פעולה בונה ופעולות set/get.

א. כתבו פעולה פנימית addRate(double rate) המוסיפה שער מטבע ביום הנוכחי.

מטבע נחשב בשיא שלו אם השער הנוכחי שלו גבוה מכל השערים הקודמים שלו.

ב. כתבו פעולה פנימית המחזירה true אם המטבע נמצא בשיא שלו.

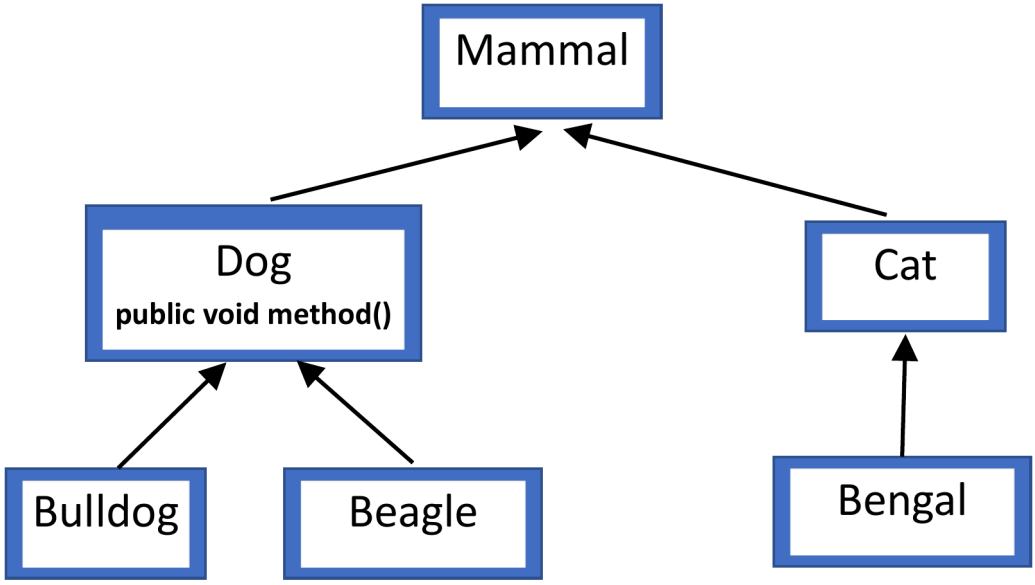
נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

מטבע נחשב "כדאי להשקעה" אם השער שלו עלה לפחות בחמישה הימים האחרונים אבל לא נמצא בשיא.

ג. כתבו פעולה פנימית המחזירה true אם מטבע הוא "כדאי להשקעה", ואם לא, הפעולה תחזיר false.

אפשר להניח שברשימת השערים יש מידע על לפחות חמשת הימים האחרונים.

7. לפניכם היררכיית מחלקות והפעולה method() הנמצאת במחלקה Dog.



נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

נתון קטע בפעולה הראשית (main)

```
Mammal mam = new Bulldog();
Dog dog = new Beagle();
Bulldog bul = new Bulldog();
……
```

בהתייחס לכל אחת מהשורות הבאות, כתבו מה יקרה בעקבות הוספת לקטע קוד שלעיל, לאחר ההצהרות על האובייקטים (במקום שמסומן בכוכביות*).

אם ההוראה אינה תקינה, ציינו נוסף להסבר, האם הוראה תגרום לשגיאת הידור (קומפילציה) או לשגיאת זמן ריצה.

Java:

```
1. Cat d1 = (Cat)mam;
2. Cat d2 = (Cat)((Mammal)mam);
3. Mammal d3 = (Cat)new Bengal();
4. Dog d4 = (Mammal)dog;
5. Beagle d5 = (Mammal)new Dog();
6. Object d6 = (Bengal)new Cat();
7. ((Bulldog)((Mammal)bul)).method();
```

C#:

```
1. Cat d1 = (Cat)mam;
2. Cat d2 = (Cat)((Mammal)mam);
3. Mammal d3 = (Cat)new Bengal();
4. Dog d4 = (Mammal)dog;
5. Beagle d5 = (Mammal)new Dog();
6. Object d6 = (Bengal)new Cat();
7. ((Bulldog)((Mammal)bul)).Method();
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

8. לפניכם שלוש פעולות secret, what, something הבאות:

Java:

```
public static int secret(Queue<Integer> q) {
  Queue<Integer> temp=new Queue<Integer>();
  while(!q.isEmpty()){
    temp.insert(q.remove());
  }
  int c=0;
  while(!temp.isEmpty()){
    c++;
    q.insert(temp.remove());
  }
  return c;
}
public static int what(Queue<Integer>q) {
  int len = secret(q);
  int temp = q.remove();
  q.insert(temp);
  for(int i = 0; i < len; i++){
    int curr = q.remove();
    q.insert(curr);
    if(curr > temp)
      temp = curr;
  }
  return temp;
}

public static boolean something(Queue<Integer> q1,
                    Queue<Integer> q2)
{
  if(secret(q1)==0)
    return false;
  int temp = what(q1);
  int len = secret(q2);
  for(int i = 0; i < len; i++){
    int curr = q2.remove();
    q2.insert(curr);
    if(temp <= curr)
      return false;
  }
  return true;
}
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

C#:

```
public static int Secret(Queue<int> q) {
  Queue<int> temp=new Queue<int>();
  while(!q.IsEmpty()){
    temp.Insert(q.Remove());
  }
  int c=0;
  while(!temp.IsEmpty()){
    c++;
    q.Insert(temp.Remove());
  }
  return c;
}
public static int What(Queue<int>q) {
  int len = Secret(q);
  int temp = q.Remove();
  q.Insert(temp);
  for(int i = 0; i < len; i++){
    int curr = q.Remove();
    q.Insert(curr);
    if(curr > temp)
      temp = curr;
  }
  return temp;
}
public static bool Something(Queue<int> q1, Queue<int> q2)
{
  if(Secret(q1)==0)
    return false;
  int temp = What(q1);
  int len = Secret(q2);
  for(int i = 0; i < len; i++){
    int curr = q2.Remove();
    q2.Insert(curr);
    if(temp <= curr)
      return false;
  }
  return true;
}
```

נתונים שלושת התורים הבאים:

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

a: ← 3, 1, 5, 1 ←

b: ← 2, 1, 3, 1 ←

c: ← 4, 4, 4 ←

א. עקבו אחרי זימון הפעולה secret(a) ורשמו מה תהיה תוצאת הזימון.

ב. עקבו אחרי זימון הפעולה what(b) ורשמו מה תהיה תוצאת הזימון.

ג. עקבו אחרי זימון הפעולה something (a ,c) ורשמו מה תהיה תוצאת הזימון.

ד. האם קיימים שני תורים q1, q2 כך something(q1, q2) = something(q2, q1)? הסבירו.

ה. מה מבצעות הפעולות secret, what, something?

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

9. לפניכם חלק ממערכת המטפלת בלקוחות של חברת הכבלים.
החברה מאפשרת לברור באחת משלוש חבילות: חבילה בסיסית, חבילה מורחבת וחבילה משולבת.
בקוד הבא נתונות שש מחלקות הקשורות ביניהן בקשרי הורשה (ללא הפעולות שיש בהן).

Java:

```
public class Cable
  {
    private double subscriptionPayment;  // מחיר קבוע למנוי
    public double getPrice()
    {
     return subscriptionPayment;
    }
  }

// חבילה בסיסית
  public class BasicPackage extends Cable
  {
    private double price; // מחיר חבילה בסיסית
  }

// חבילה מורחבת המכילה ערוצים נוספים
  public class ExtendedPackage extends BasicPackage
  {
    private double price;   // מחיר קבוע עבור כל ערוץ נוסף
    private int numOfChannels;
    private String[] namesOfChannels; // מערך שמות ערוצים
   }

// מחלקה עבור טלפון
  public class Phone extends Cable
  {
    private double price;  // מחיר שיחה
    private int numberOfCalls;  // מספר שיחות
  }

// חבילה משולבת
  public class Triple extends Cable
  {
    private Cable package;  // חבילה בסיסית או מורחבת
    private Phone phone;    // טלפון
    private double priceOfInternet;  // מחיר השימוש באינטרנט
  }

// מחלקה עבור לקוח
  public class Client
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

```
  {
    private String nameClient;
    private Cable cable;
  }
```

C#:

```
public class Cable
  {
    private double subscriptionPayment;  // מחיר קבוע למנוי
    public double GetPrice()
    {
     return subscriptionPayment;
    }
  }

// חבילה בסיסית
  public class BasicPackage : Cable
  {
    private double price; // מחיר חבילה בסיסית
  }

// חבילה מורחבת המכילה ערוצים נוספים
  public class ExtendedPackage : BasicPackage
  {
    private double price;   // מחיר קבוע עבור כל ערוץ נוסף
    private int numOfChannels;
    private string[] namesOfChannels; // מערך שמות ערוצים
   }

// מחלקה עבור טלפון
  public class Phone : Cable
  {
    private double price;  // מחיר שיחה
    private int numberOfCalls;  // מספר שיחות
  }

// חבילה משולבת
  public class Triple : Cable
  {
    private Cable package;  // חבילה בסיסית או מורחבת
    private Phone phone;    // טלפון
    private double priceOfInternet;  // מחיר השימוש באינטרנט
  }
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

```
// מחלקה עבור לקוח
  public class Client
  {
    private string nameClient;
    private Cable cable;
  }
```

הניחו שעבור כל התכונות בכל המחלקות קיימות פעולות get ו- set.

לקוח החברה יכול לרכוש מנוי לפי בחירתו. החברה קבעה שיטת מחיר לכל חבילה שהיא מספקת לפי הכללים הבאים:

- למחיר כל חבילה מתווסף גם המחיר הקבוע למנוי.
- מחיר חבילה מורחבת הוא מחיר החבילה הבסיסית + [מחיר ערוץ] * [מספר הערוצים].
- התשלום עבור שיחות הטלפון הוא [מחיר שיחה] * [מספר שיחות]
- לוקח הרוכש חבילה משולבת מקבל 10% הנחה (זאת אומרת שמחיר חבילה משולבת הוא 90% מסכום כל מרכיביו).

א. ממשו את פעולת getPrice בכל אחת מ**ארבע** המחלקות האחרות:
BasicPackage, ExtendedPackage,Phone, Triple.
ממשו את הפעולה getPrice במחלקה: Client

נתונה מחלקה "חברת הכבלים" Company המכילה מערך clients מטיפוס Client בגודל 1000 ומונה לקוחות numClients.

כתבו במחלקה Company את הפעולות הבאות:

ב. הפעולה המדפיסה את שמות הלקוחות שיש להם ערוץ "ספורט".

ג. הפעולה המחזירה את כמות הלקוחות שיש להם חבילה בסיסית בלבד.

10. חברת הובלות TrickTruck מעוניינת לעקוב אחרי תצרוכת הדלק של כל אחד כלי הרכב שהיא מפעילה, במשך התקופה של חודש יוני (30 יום).

כל נהג מתדלק מביא לתחנה חשבונית המכילה את הפרטים הבאים:

- יום התדלוק (מספר שלם בין 1-30).
- מספר רכב (מספר שלם).
- שם הנהג המתדלק (מחרוזת).
- כמות הדלק שתודלקה בתדלוק זה (מספר ממשי).

רכב יכול להיות מתודלק כמה פעמים ביום או לא מתודלק כלל ביום מסוים.

במחלקת **חשבונית** Payment הוגדרו פעולות הבאות:

- פעולה בונה היוצרת חשבונית הכוללת את שם הנהג name את מספר הרכב num ואת כמות הדלק שתידלק fuel.

  Payment (String name, int num, double fuel)
- פעולות get ו- set - לכל תכונה והפעולה toString המחזירה מחרוזת המתארת את מצב החשבונית.

המידע עבור כל התדלוקים בחודש אפריל נשמר במחלקה **TrickTruck** במערך חד-ממדי payments של רשימות מסוג חשבונית, Payment. כל תא במערך מייצג יום תדלוק. שתי תכונות נוספות הן רשימת הנהגים drivers, ורשימת הרכבים cars:

Java:

```
class TrickTrack
{
 private Node<String> drivers;
 private Node<Integer> cars;
 private Node<Payment>[] payments;
... .
```

C#:

```
class TrickTrack
{
  private Node<string> drivers;
  private Node<int> cars;
  private Node<Payment>[] payments;
```

במחלקה הוגדרו פעולה בונה, פעולות get/set.

במחלקה הוגדרו שלוש פעולות בוליאניות נוספות:

- existDriver(String name) - הפעולה מחזירה true אם הנהג name עבד בחודש יוני, ואם לא הפעולה מחזירה false.

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

- existCar(int num) - הפעולה מחזירה true אם הרכב num עבד בחודש יוני, ואם לא הפעולה מחזירה false.
- worked(String name, int day) - הפעולה מחזירה true אם נהג name הגיש קבלת תשלום ביום day, ואם לא, הפעולה תחזיר false.

א. כתבו פעולה במחלקה TrickTruck בשם addPayment שתקבל כפרמטר את היום בחודש day, את מספר הרכב num, את שם הנהג name ואת כמות הדלק fuel שתדלק.

אם כל הפרמטרים שהפעולה מקבלת הם תקינים, הפעולה מוסיפה חשבונית חדשה לרשימה המתאימה במערך payments ומחזירה true, ואם לא הפעולה מחזירה false, ולא מבצעת דבר.

ב. כתבו במחלקה TrickTruck את הפעולה totalFuel(int num) המקבלת מספר רכב ומחזירה את כמות הדלק הכללית שתדלקו את הרכב.

ג. כתבו במחלקה TrickTruck את הפעולה printWorkDays() המדפיסה עבור כל נהג את מספר הימים שעבורם הוא הציג קבלות.

11. לפניכם ארבע מחלקות:

Java:

```
class A
{
  private static int countA = 0;
  protected int n;

  public A(int n1)
  {
    this.n = n1;
    A.countA++;
    System.out.println(A.countA + " " + this.n);
  }

  public int getN()
  {
    return this.n;
  }

  public void doIt()
  {
    for (int i = 0; i < this.n; i++)
      System.out.print (i + ",");
    System.out.println();
  }

  public void print()
  {
    System.out.println (this.n);
  }
}

class B extends A
{
  private static int countB = 0;

  public B(int n2)
  {
   super(n2);
   B.countB++;
   System.out.println (B.countB +" " + n2);
  }

  public void doIt()
  {
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

```
    System.out.println ("B Does It!");
  }
}

class C extends B
{
  private static int countC = 0;
  private A a;

  public C(int n1, int n2)
  {
    super(n1);
    this.a = new A(n2);
    countC++;
    System.out.println ("C is " + (a.getN() + this.n));
  }

  public void print()
  {
    super.print();
    this.a.print();
  }
}

class Test
{
  private A[] arr;

  public Test()
  {
    arr = new A[3];
    arr[0] = new A(4);
    arr[1] = new B(2);
    arr[2] = new C(1, 6);
  }

  public void testIt()
  {
    for (int i = 0; i < arr.length; i++)
    {
      arr[i].doIt();
      arr[i].print();
    }
  }
}
```

C#:

```
class A
{
  private static int countA = 0;
  protected int n;

  public A(int n1)
  {
    this.n = n1;
    A.countA++;
    Console.WriteLine(A.countA + " " + this.n);
  }

  public int GetN()
  {
    return this.n;
  }

  public virtual void DoIt()
  {
    for (int i = 0; i < this.n; i++)
      Console.Write (i + ",");
    Console.WriteLine();
  }

  public virtual void Print()
  {
    Console.WriteLine (this.n);
  }
}

class B : A
{
  private static int countB = 0;

  public B(int n2): base(n2)
  {
    B.countB++;
    Console.WriteLine (B.countB +" " + n2);
  }

  public override void DoIt()
  {
    Console.WriteLine ("B Does It!");
  }
```


נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

```
}

class C : B
{
  private static int countC = 0;
  private A a;

  public C(int n1, int n2):base(n1)
  {
    this.a = new A(n2);
    countC++;
    Console.WriteLine("C is " + (a.GetN() + this.n));
  }

  public override void Print()
  {
    base.Print();
    this.a.Print();
  }
}

class Test
{
  private A[] arr;

  public Test()
  {
    arr = new A[3];
    arr[0] = new A(4);
    arr[1] = new B(2);
    arr[2] = new C(1, 6);
  }

  public void TestIt()
  {
    for (int i = 0; i < arr.Length; i++)
    {
      arr[i].DoIt();
      arr[i].Print();
    }
  }
}
```

א. **בנו תרשים מחלקות UML (הכולל קשרים) עבור ארבע המחלקות המוצגות בשאלה.**
על התרשים לכלול את שמות המחלקות ואת הקשרים ביניהם.

נכתב ונבדק על ידי צוותי GOOL בגרויות ו-GuyAI · www.guyai.co.il

ב. לפניכם קטע קוד:

Java:

```
public  static void main(String[] args)
{
   Test test = new Test();
   test.testIt();

}
```

C#:

```
public static void Main(string[] args)
{
   Test test = new Test();
   test.TestIt();
}
```

עבור קטע קוד זה בצעו והציגו את פלט הקטע. **חובה להציג מעקב.**

כל מבחני מה"ט עם פתרונות מלאים: www.guyai.co.il/mahat