

а. на аттестат зрелости для средних школ
б. на аттестат зрелости для экстернов
Время проведения экзамена: 2016 год, срок "бет"
Номер вопросника: 899205, 603
Перевод на русский язык (5)

א. בגרות לבתי ספר על-יסודיים
ב. בגרות לנבחנים חיצוניים
מועד הבחינה: תשע"ו, מועד ב
מספר השאלון: 899205, 603
תרגום לרוסית (5)

Компьютеры "бет"

2 единицы обучения (дополнение до 5-ти единиц)

מדעי המחשב ב'

2 יחידות לימוד (השלמה ל-5 יח"ל)

Указания экзаменуемым

- а. Продолжительность экзамена: три часа.
б. Строение вопросника и ключ к оценке:
в этом вопроснике два раздела.
Раздел первый: в этом разделе четыре вопроса.
Вы должны ответить на два из них.
 $(2 \times 25) = 50$ баллов
Раздел второй: в этом разделе вопросы из четырех различных курсов. Вы должны ответить только на вопросы курса, который вы изучали, согласно указаниям этого курса. $(2 \times 25) = 50$ баллов
Всего – 100 баллов
в. Разрешенный вспомогательный материал:
любой вспомогательный материал, за исключением компьютера с возможностью программирования.
г. Особые указания:

- Все программы, которые вы должны написать на языке программирования в первом разделе, следует писать только на одном языке – Java или C#.
- На внешней обложке тетради укажите язык программирования, который вы используете: Java или C#.
- На внешней обложке тетради укажите название курса, который вы изучали, одного из четырех следующих курсов: "Компьютерные системы и ассемблер" [מערכות מחשב ואסמבלר], "Введение в исследование операций" [מבוא לחקר ביצועים], "Вычислительные модели" [מודלים חישוביים], "Объектно-ориентированное программирование" [תכנות מונחה עצמים].

Примечание: в программах, которые вы должны написать, за использование большой буквы вместо маленькой или наоборот баллы сниматься не будут.

כתוב במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונך לכתוב כטייפ (ראשי פרקים, חישובים וכדומה). רשום "טייפ" בראש כל עמוד טייפ. רישום טייפות לשהן על דפים מחוץ למחברת הבחינה עלול לגרום לפסילת הבחינה!

Все черновые записи (тезисы, вычисления и т.п.) делайте только на отдельных страницах экзаменационной тетради. Напишите слово «черновик» в начале каждой страницы, отведенной вами под черновик. Выполнение любых черновых записей на листах, не относящихся к экзаменационной тетради, может привести к тому, что экзамен будет аннулирован!

Желаем успеха!

בהצלחה!

הוראות לנבחן

- א. משך הבחינה: שלוש שעות.
ב. מבנה השאלון ומפתח ההערכה:
בשאלון זה שני פרקים.
פרק ראשון – בפרק זה ארבע שאלות, ומהן יש לענות על שתיים.
 $(25 \times 2) = 50$ נקודות
פרק שני – בפרק זה שאלות בארבעה מסלולים שונים, ענה על שאלות רק במסלול שלמדת, לפי ההוראות בקבות השאלות במסלול זה.
 $(25 \times 2) = 50$ נקודות
סה"כ – 100 נקודות
ג. חומר עזר מותר בשימוש:
כל חומר עזר, חוץ ממחשב הניתן לתכנות.
ד. הוראות מיוחדות:

- את כל התכניות שאתה נדרש לכתוב בשפת מחשב בפרק הראשון כתוב בשפה אחת בלבד – Java או C#.
 - רשום על הכריכה החיצונית של המחברת באיזו שפה אתה כותב – Java או C#.
 - רשום על הכריכה החיצונית של המחברת את שם המסלול שלמדת, המסלול הוא אחד מארבעת המסלולים האלה: מערכות מחשב ואסמבלר, מבוא לחקר ביצועים, מודלים חישוביים, תכנות מונחה עצמים.
- הערה: בתכניות שאתה כותב לא יורדו לך נקודות, אם תכתוב אות גדולה במקום אות קטנה או להפך.

Вопросы

В данном вопроснике два раздела: **раздел первый** и **раздел второй**.
Вы должны ответить на вопросы обоих разделов, согласно инструкциям каждого раздела.

РАЗДЕЛ ПЕРВЫЙ (50 баллов)

Внимание: во всех вопросах, где требуется реализация [מימוש], разрешается пользоваться методами классов Список, Очередь, Стек, Двоичное дерево и Звено [רשימה, תור, מחסנית, עץ בינרי וחוליה], не реализуя их. Если вы используете дополнительные методы, вы должны их реализовать [לממש אותם].

Ответьте на два из вопросов 1-4 (за каждый вопрос – 25 баллов).

1. Список [רשימה] L называется **тройным**, если он удовлетворяет следующим условиям:
- * список не пустой;
 - * число элементов списка делится на 3 без остатка;
 - * элементы в первой трети списка содержат те же значения, что и элементы во второй трети списка, и те же значения, которые содержат элементы в третьей трети списка. Причем, порядок элементов в каждой из третей списка одинаков.
- Например, список L1, представленный на схеме, является **тройным** списком из 12 элементов.

L1: 2 → 5 → 3 → 7 → 2 → 5 → 3 → 7 → 2 → 5 → 3 → 7 → null

Напишите на Java или C# внешний метод [פעולה חיצונית], который получит список L с элементами целого типа.

Если L является **тройным** списком, метод вернет true .

Иначе [אחרת] – метод вернет false .

2. **ВНИМАНИЕ:** данный вопрос представлен в двух версиях:
одна – на Java (стр. 3-4), а другая – на C# (стр. 5-6).
Отвечайте в соответствии с языком, который вы изучали.

Для решающих на Java:

Перед вами методы `sod` и `what`, получающие массив `[מערך]` `a` с элементами целого типа `[טיפוס שלם]`, отсортированный по возрастанию, а также целое число `k`.

У обоих методов одинаковое условие выхода `[טענת יציאה]`.

```
public static boolean sod(int[] a , int k)
{
    for (int i = 0; i < a.length-1; i++)
    {
        int j = i+1;
        while (j < a.length)
        {
            if (a[i] + a[j] == k)
                return true;
            j++;
        }
    }
    return false;
}
```

```
public static boolean what(int[] a , int k)
{
    int left = 0 , right = a.length-1;
    while (left < right)
    {
        if (a[left] + a[right] == k)
            return true;
        if (a[left] + a[right] < k)
            left++;
        else
            right--;
    }
    return false;
}
```

/продолжение на странице 4/

Дан массив a : a

2	4	7	12	18
---	---	---	----	----

- (א) Проследите с помощью таблицы трассировки [טבלת מעקב] за выполнением метода `sod` для данного массива a и числа $k = 11$. Запишите возвращаемое значение.

Таблица трассировки должна содержать столбцы для i , j , $a[i]$, $a[j]$ и дополнительный столбец, в котором будет указано, выполняется или не выполняется условие команды `if`.

- (ב) Проследите с помощью таблицы трассировки за выполнением метода `sod` для данного массива a и числа $k = 10$. Запишите возвращаемое значение. Таблица трассировки должна содержать столбцы, перечисленные в пункте (א).

- (ג) Каково условие выхода [טענת יציאה] метода `sod` ?

- (ד) Какова временная сложность [סיבוכיות זמן הריצה] выполнения метода `sod` ? Обоснуйте ваш ответ.

- (ה) Проследите с помощью таблицы трассировки за выполнением метода `what` для данного массива a и числа $k = 11$. Запишите возвращаемое значение.

Таблица трассировки должна содержать столбцы для `left`, `right`, `a[left]`, `a[right]` и два дополнительных столбца для каждой из команд `if`. В каждом столбце будет указано, выполняется или не выполняется условие команды `if`.

- (ו) Какова временная сложность [סיבוכיות זמן הריצה] выполнения метода `what` ? Обоснуйте ваш ответ.

- (ז) Какой из двух методов – `sod` или `what` – более эффективный? Обоснуйте ваш ответ.

- (ח) Условие входа [טענת הכניסה] методов `sod` и `what` было изменено так, что им можно передавать неотсортированный массив a .

(1) Изменится ли условие выхода метода `sod` ? Обоснуйте ваш ответ.

(2) Изменится ли условие выхода метода `what` ? Обоснуйте ваш ответ.

Для решающих на C#:

Перед вами методы Sod и What , получающие массив [מערך] a с элементами целого типа [טיפוס שלם], отсортированный по возрастанию, а также целое число k .

У обоих методов одинаковое условие выхода [טענת יציאה].

```
public static bool Sod(int[] a , int k)
{
    for (int i = 0; i < a.Length-1; i++)
    {
        int j = i+1;
        while (j < a.Length)
        {
            if (a[i] + a[j] == k)
                return true;
            j++;
        }
    }
    return false;
}
```

```
public static bool What(int[] a , int k)
{
    int left = 0 , right = a.Length-1;
    while (left < right)
    {
        if (a[left] + a[right] == k)
            return true;
        if (a[left] + a[right] < k)
            left++;
        else
            right--;
    }
    return false;
}
```

/продолжение на странице 6/

Дан массив a :

2	4	7	12	18
---	---	---	----	----

- (א) Проследите с помощью таблицы трассировки [טבלת מעקב] за выполнением метода Sod для данного массива a и числа $k = 11$. Запишите возвращаемое значение.
Таблица трассировки должна содержать столбцы для: i , j , $a[i]$, $a[j]$ и дополнительный столбец, в котором будет указано выполняется или не выполняется условие команды `if`.
- (ב) Проследите с помощью таблицы трассировки за выполнением метода Sod для данного массива a и числа $k = 10$. Запишите возвращаемое значение. Таблица трассировки должна содержать столбцы, перечисленные в пункте (א).
- (ג) Каково условие выхода [טענת יציאה] метода Sod ?
- (ד) Какова временная сложность [סיבוכיות זמן הריצה] выполнения метода Sod? Обоснуйте ваш ответ.
- (ה) Проследите с помощью таблицы трассировки за выполнением метода What для данного массива a и числа $k = 11$. Запишите возвращаемое значение.
Таблица трассировки должна содержать столбцы для: `left`, `right`, $a[\text{left}]$, $a[\text{right}]$ и два дополнительных столбца для каждой из команд `if`. В каждом столбце будет указано, выполняется или не выполняется условие команды `if`.
- (ו) Какова временная сложность [סיבוכיות זמן הריצה] выполнения метода What ? Обоснуйте ваш ответ.
- (ז) Какой из двух методов – Sod или What – более эффективный? Обоснуйте ваш ответ.
- (ח) Условие входа [טענת הכניסה] методов Sod и What было изменено так, что им можно передавать неотсортированный массив a .
(1) Изменится ли условие выхода метода Sod ? Обоснуйте ваш ответ.
(2) Изменится ли условие выхода метода What ? Обоснуйте ваш ответ

3. (א) Напишите внешний метод [פעולה חיצונית], который получит непустое двоичное дерево [עץ בינארי] t , в котором каждая вершина содержит стек [מחסנית] целых чисел, больших чем 0.

Метод вернет стек. Для каждой вершины на дереве t в возвращаемый стек будет введен элемент следующим способом: если в стеке данной вершины есть только один элемент, то его значение будет введено в возвращаемый стек; если в стеке данной вершины есть два элемента, то в возвращаемый стек будет введена их сумма; если стек в данной вершине пустой – будет введен 0. В любом другом случае будет введена сумма трех верхних элементов стека данной вершины.

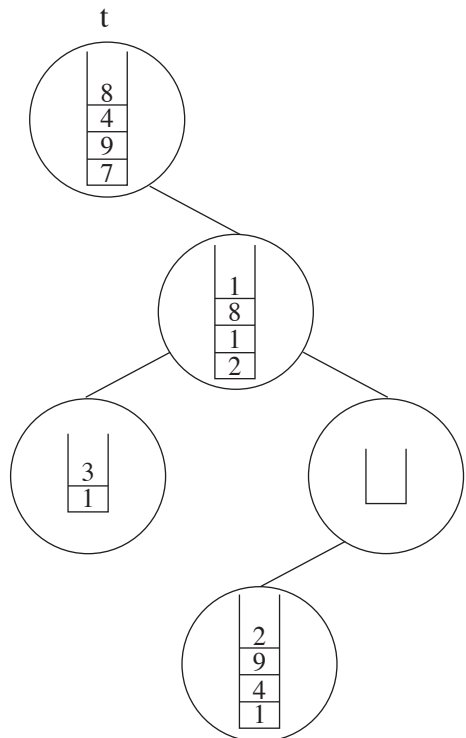
Порядок элементов в возвращаемом стеке будет соответствовать внутреннему порядку обхода (inorder) вершин дерева t .

Нет необходимости сохранять содержимое стеков дерева t .

Например, для дерева t :

возвращенный стек:

0
15
10
4
21



- (ב) Какова временная сложность [סיבוכיות זמן הריצה] метода, написанного вами для пункта (א)? Обоснуйте ваш ответ.

/продолжение на странице 8/

4. **ВНИМАНИЕ:** данный вопрос представлен в двух версиях:
одна – на Java (стр. 8-9), а другая – на C# (стр. 10-11).
Отвечайте в соответствии с языком, который вы изучали.

Для решающих на Java:

Игра “Сыграй-ка” – это игра для одного игрока, состоящая из двух этапов. Игрок получает 52 перемешанные карты, отличные друг от друга. На каждой из карт есть число от 1 до 13 и изображение одной из четырех фигур. Фигуры представлены числами от 1 до 4 .

На первом этапе игрок раскладывает карты по 4 стопкам, в соответствии с фигурой, изображенной на карте, чтобы в каждой стопке оказались карты с одинаковой фигурой. Карты складываются одна на другую.

На втором этапе игрок случайным образом выбирает число от 1 до 4, обозначающее фигуру. Игрок поднимает верхнюю карту со стопки, фигура которой соответствует выбранному числу, и переносит карту в пятую стопку.

Этот этап продолжается до завершения игры.

Игра завершается, когда выбрано число, соответствующее стопке, в которой больше нет карт.

Выигрыш – это ситуация, при которой сумма чисел на картах в пятой стопке делится на 100 без остатка.

После того, как игра окончена, проверяют, есть ли выигрыш.

Для реализации данной игры заданы классы: **Card**, представляющий одну карту, **Deck**, представляющий пять необходимых для игры стопок и **Test**, управляющий игрой.

Также написан метод `public static boolean game(Card[] cards)` в классе **Test**, который получает массив из 52 разных карт, упорядоченных случайным образом. Метод проводит игру до ее завершения. Метод возвращает “true”, если игрок выиграл в игре, в противном случае метод возвращает “false”.

Перед вами класс **Card**, представляющий карту с числом (value) от 1 до 13 и фигурой (shape) от 1 до 4.

```
public class Card
{
    private int value;
    private int shape;

    public Card (int value, int shape)
    {
        this.value = value;
        this.shape = shape;
    }
    public int getValue () { return this.value; }
    public int getShape () { return this.shape; }
}
```

- (א) i Напишите на Java заголовок [כותרת] класса **Deck** и его атрибуты [תכונות].
- ii Напишите на Java в классе **Deck** заголовки следующих методов:
- Конструктор, определяющий все 5 стопок пустыми.
 - Метод insert , получающий карту и добавляющий ее в соответствующую стопку на первом этапе игры.
 - Метод move, случайным образом выбирающий число, определяющее фигуру. Если соответствующая выбранной фигуре стопка карт пуста, метод вернет “false”. В противном случае метод перенесет в пятую стопку верхнюю карту из стопки, соответствующей выбранной фигуре, и вернет “true”.
 - Метод sum, возвращающий сумму чисел на картах, находящихся в пятой стопке в момент вызова данного метода.

Внимание: в этом подпункте нет необходимости реализовывать эти методы

- (ב) Реализуйте на Java конструктор класса **Deck**.
- (ג) Реализуйте на Java метод sum класса **Deck**.
- (ד) Реализуйте на Java метод game класса **Test**.

Вы можете использовать методы insert и move класса Deck не реализуя их. Если вы пользуетесь дополнительными методами, вы должны реализовать их.

/продолжение на странице 10/

Для решающих на C#:

Игра “Сыграй-ка” – это игра для одного игрока, состоящая из двух этапов. Игрок получает 52 перемешанные карты, отличные друг от друга. На каждой из карт есть число от 1 до 13 и изображение одной из четырех фигур. Фигуры представлены числами от 1 до 4 .

На первом этапе игрок раскладывает карты по 4 стопкам, в соответствии с фигурой, изображенной на карте, чтобы в каждой стопке оказались карты с одинаковой фигурой. Карты складываются одна на другую.

На втором этапе игрок случайным образом выбирает число от 1 до 4, обозначающее фигуру. Игрок поднимает верхнюю карту со стопки, фигура которой соответствует выбранному числу, и переносит карту в пятую стопку.

Этот этап продолжается до завершения игры.

Игра завершается, когда выбрано число, соответствующее стопке, в которой больше нет карт.

Выигрыш – это ситуация, при которой сумма чисел на картах в пятой стопке делится на 100 без остатка.

После того, как игра окончена, проверяют, есть ли выигрыш.

Для реализации данной игры заданы классы: **Card**, представляющий одну карту, **Deck**, представляющий пять необходимых для игры стопок и **Test**, управляющий игрой.

Также написан метод `public static boolean game(Card[] cards)` в классе **Test**, который получает массив из 52 разных карт, упорядоченных случайным образом. Метод проводит игру до ее завершения. Метод возвращает “true”, если игрок выиграл в игре, в противном случае метод возвращает “false”.

Перед вами класс **Card**, представляющий карту с числом (value) от 1 до 13 и фигурой (shape) от 1 до 4.

```
public class Card
{
    private int value;
    private int shape;

    public Card (int value, int shape)
    {
        this.value = value;
        this.shape = shape;
    }

    public int GetValue () { return this.value; }
    public int GetShape () { return this.shape; }
}
```

- (א) i Напишите на C# заголовок [כותרת] класса **Deck** и его атрибуты [תכונות].
- ii Напишите на C# в классе **Deck** заголовки следующих методов:
- Конструктор, определяющий все 5 стопок пустыми.
 - Метод Insert , получающий карту и добавляющий ее в соответствующую стопку на первом этапе игры.
 - Метод Move, случайным образом выбирающий число, определяющее фигуру. Если соответствующая выбранной фигуре стопка карт пуста, метод вернет “false”. В противном случае метод перенесет в пятую стопку верхнюю карту из стопки, соответствующей выбранной фигуре, и вернет “true”.
 - Метод Sum, возвращающий сумму чисел на картах, находящихся в пятой стопке в момент вызова данного метода.

Внимание: в этом подпункте нет необходимости реализовывать эти методы

- (ב) Реализуйте на C# конструктор класса **Deck**.
- (ג) Реализуйте на C# метод Sum класса **Deck**.
- (ד) Реализуйте на C# метод Game класса **Test**.

Вы можете использовать методы Insert и Move класса **Deck** не реализуя их. Если вы пользуетесь дополнительными методами, вы должны реализовать их.

/продолжение на странице 12/

РАЗДЕЛ ВТОРОЙ (50 баллов)

В данном разделе вопросы четырёх разных курсов:

- Компьютерные системы и ассемблер [מערכות מחשב ואסמבלי], стр. 12-16.
- Введение в исследование операций [מבוא לחקר ביצועים], стр. 17-27.
- Вычислительные модели [מודלים חישוביים], стр. 28-29.
- Объектно-ориентированное программирование [תכנות מונחה עצמים] на Java, стр. 30-41; на C#, стр. 42-53.

Ответьте только на вопросы того курса, который вы изучали.

Компьютерные системы и Ассемблер [מערכות מחשב ואסמבלי]

Если вы изучали этот курс, ответьте на два вопроса из вопросов 5-8 (за каждый вопрос – 25 баллов).

5. В данном вопросе два не связанных между собой пункта. Ответьте на оба пункта (א) и (ב).

(א) Перед вами способ вычисления квадрата целого числа, большего нуля:

$$1^2 = 1$$

$$2^2 = 1 + 3$$

$$3^2 = 1 + 3 + 5$$

$$4^2 = 1 + 3 + 5 + 7$$

$$5^2 = 1 + 3 + 5 + 7 + 9$$

и так далее.

В AL хранится целое число. Это число больше нуля и меньше 16 .

Перед вами фрагмент [טקט] программы на Ассемблере, который вычисляет квадрат числа, записанного в AL , в соответствии с описанным способом, и заносит результат в AX . Из этого фрагмента были удалены целые строки и части строк в местах, обозначенных цифрами i-iv. Перепишите в вашу тетрадь цифры, обозначающие удаленные места, и запишите напротив каждой из цифр строку команды, включая недостающие части, чтобы фрагмент программы выполнял требуемое действие.

		MOV	BX , 0
i.		-----	-----
		MOV	DX , 1
		MOV	CX , AX
		DEC	CX
ii.		-----	-----
	A1:	ADD	BX , DX
iii.		ADD	DX , -----
		LOOP	A1
	A2:	ADD	BX , DX
iv.		MOV	AX , -----
	A3:	NOP	

/продолжение на странице 13/

(ב) (Нет связи с пунктом (א)).

Фрагмент данных [מקטע הנתונים] содержит следующее определение данных:

T DB 55 , 90 , 110 , 1

Внимание: числа десятичные [עשרוניים].

Перед вами фрагмент [קטע] программы на Ассемблере, цель которого вычислить сумму элементов массива T .

```
MOV      CX , 4
XOR      AX , AX
LEA      BX , T
AGAIN:   ADD      AL , [BX]
INC      BX
LOOP     AGAIN
```

Данный фрагмент программы не выполняет требуемого действия.

- (1) Проследите с помощью таблицы трассировки [טבלת מעקב] за выполнением данного фрагмента и запишите, что будет содержаться в AL , а также во флаге переноса [דגל הנשא] и флаге переполнения [דגל הגלישה] по окончании выполнения данного фрагмента.
- (2) Измените данный фрагмент программы так, чтобы он вычислял сумму элементов данного массива T .
Перепишите в вашу тетрадь измененный фрагмент программы.

6. В данном вопросе два не связанных между собой пункта (א) и (ב).

Ответьте на оба пункта.

(א) Во фрагменте данных [מקטע הנתונים] определен массив:

ARR DB 5 DUP (?)

Перед вами 4 фрагмента [קטעים] (i)-(iv) на Ассемблере, цель которых записать 0 во все ячейки массива ARR .

Проследите с помощью таблицы трассировки [טבלת מעקב] за выполнением каждого из приведенных ниже фрагментов (i)-(iv) и определите, выполняет он требуемое действие или не выполняет.

i		MOV	SI , 0
		MOV	CX , 4
	A1:	MOV	ARR[SI] , 0
		INC	SI
		LOOP	A1
ii		MOV	CX , 5
		LEA	BX , ARR
		MOV	AL , 0
	A1:	MOV	[BX] , AL
		INC	BX
iii		LOOP	A1
		MOV	BX , 5
		DEC	BX
	A1:	MOV	ARR[BX] , 0
iv		DEC	BX
		JNZ	A1
		MOV	DI , 0
	A1:	MOV	ARR[DI] , 0
		INC	DI
		CMP	DI , 5
		JC	A1

(ב) (Нет связи с пунктом (א)).

Перед вами фрагмент программы на Ассемблере:

Обратите внимание: данные шестнадцатиричные [הקסדצימליים].

```
START:    MOV     AX , C83BH
          MOV     BX , A89CH
          SHL     AX , 1
          OR      AL , 33H
          NOT     BL
          ADD     AX , BX
```

Проследите с помощью таблицы трассировки [טבלת מעקב] за выполнением данного фрагмента программы. В таблице трассировки детализируйте содержимое CF , SF , ZF , BX , AX на каждом этапе.

7. Фрагмент данных [מקטע הנתונים] содержит определение двух массивов: массив NUMBERS размером 100 байт, содержащий шестнадцатичные числа от 0 до 0FF, и массив APP размером 256 байт, начинающийся с адреса 2500H и содержащий нули.
Напишите на Ассемблере программу, которая совершит только один проход по всем элементам массива NUMBERS и запишет в элементе i массива APP, сколько раз число i встретилось в массиве NUMBERS.
8. В регистре [אוגר] AX хранится двоичное [בינרי] число.
Напишите на Ассемблере фрагмент программы [קטע תכנית], который запишет в регистр BL число появлений последовательности 1011 в двоичном числе из регистра AX.
Например: если в регистре AX хранится двоичное число
0110110010110110110 ,
то в регистр BL будет записано число 3 .

Введение в исследование операций [מבוא לחקר ביצועים]

Если вы изучали этот курс, ответьте на два из вопросов 9-12 (за каждый вопрос – 25 баллов).

9. Дана задача линейного программирования [תכנון לינארי]:

$$\text{Max } \{z = 5x_1 - x_2\}$$

при условиях: $2x_1 + 3x_2 \leq 18$

$$x_1 - x_2 \leq 2$$

$$x_1 \leq 3$$

$$x_2 \geq 0$$

x_1 без ограничения знака.

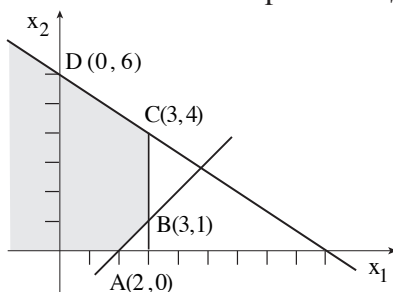
Известно также, что решение $(3, 1)$ является оптимальным решением данной задачи.

Все последующие пункты (א)-(ה) относятся к данной задаче линейного программирования.

Ответы на вопросы этих пунктов не зависят друг от друга.

Ответьте на вопросы всех пунктов.

(א) Перед вами чертеж области возможных решений данной задачи.



Скопируйте чертеж в вашу тетрадь и выполните следующие шаги:

шаг 1: Добавьте к чертежу в вашей тетради проекцию высоты целевой функции [היטל הגובה של פונקציית המטרה] для $z = 5$.

[1 תוצ] Вычислите координаты точек пересечения этой проекции с осями x_1 и x_2 и обозначьте их на чертеже.

шаг 2: Добавьте к чертежу в вашей тетради проекцию высоты целевой функции для $z = 15$.

[2 תוצ] Вычислите координаты точек пересечения этой проекции с осями x_1 и x_2 и обозначьте их на чертеже.

шаг 3: При помощи стрелки обозначьте на чертеже в вашей тетради направление возрастания целевой функции.

[3 תוצ] Показывают ли шаги 1-3, что решение $(3, 1)$ является единственным оптимальным решением? Обоснуйте ваш ответ.

Обратите внимание: продолжение вопроса на следующей странице

/продолжение на странице 18/

- (ב) К задаче, описанной в начале вопроса, добавили условие:

$$x_1 - x_2 \geq 2$$

Начертите в вашей тетради новую систему осей и обозначьте на ней только область возможных решений, полученную после добавления этого условия.

- (ג) К задаче, описанной в начале вопроса, добавляют некоторое условие, после чего областью возможных решений становится отрезок [уор] между точками (3, 1) и (3, 4). Изменится ли данное оптимальное решение (3, 1)? Обоснуйте ваш ответ.

- (ד) К задаче, описанной в начале вопроса, добавляют новое условие:

$$3x_1 + 2x_2 \leq 14.$$

Ученик утверждал, что данное оптимальное решение (3, 1) не изменится.

Прав ли был ученик? Обоснуйте ваш ответ, не изображая на чертеже новую область возможных решений, полученную после добавления этого условия.

- (ה) В задаче, описанной в начале вопроса, изменяют только целевую функцию на $z = 3x_1$.

Перед вами утверждения i-iv, только одно из которых является верным. Перепишите в вашу тетрадь верное утверждение и обоснуйте ваш выбор.

i Единственным оптимальным решением будет (3, 1).

ii Единственным оптимальным решением будет (3, 4).

iii Новое оптимальное решение будет таким:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \lambda \begin{pmatrix} 3 \\ 1 \end{pmatrix} + (1 - \lambda) \begin{pmatrix} 3 \\ 4 \end{pmatrix}$$

где $0 \leq \lambda \leq 1$

iv Оптимальное решение будет неограниченным [לא חסום].

/продолжение на странице 19/

10. В данном вопросе шесть не связанных между собой пунктов (א)-(ו).
Ответьте на вопросы всех пунктов.

(א) Перед вами транспортная задача [בעיית תובלה] с 2-мя пунктами отправления [מקורות] и 4-мя пунктами назначения [ימים]. Стоимости перевозки единицы груза из каждого пункта отправления в каждый пункт назначения приведены в таблице:

Пункты отправления	Пункты назначения				Предложение
	1	2	3	4	
1	8	9	4	6	100
2	14	12	13	8	80
Спрос	10	40	30	70	

В соответствии с этой таблицей, невозможно применить метод северо-западного угла [שיטת הפינה הצפון-מערבית], чтобы найти возможное решение данной транспортной задачи. Объясните почему.

- (ב) В приведенной ниже таблице дана часть возможного базового решения транспортной проблемы: $x_{12} = 2$, $x_{11} = 10$.

Пункты отправления	Пункты назначения			Предложение
	1	2	3	
1	4 10	5 2	10	12
2	6	3	6	12
3	6	2	6	6
Спрос	10	10	10	

Перепишите эту таблицу в вашу тетрадь и дополните ее в соответствии с методом северо-западного угла .

Обратите внимание: продолжение вопроса на следующей странице
/продолжение на странице 20/

- (ג) В приведенной ниже таблице дано возможное базовое решение транспортной проблемы. Значение, установленное для u_1 , равно 0. Перепишите эту таблицу в вашу тетрадь, вычислите значения v_1, v_2, v_3, u_2, u_3 и запишите полученные значения в соответствующие ячейки таблицы.

Пункты отправления	Пункты назначения			Предло- жение	u_i
	1	2	3		
1	4 10	5 2	10	12	0
2	6	3 2	6 10	12	
3	6	2 6	6	6	
Спрос	10	10	10		
v_j					

- (7) В приведенной ниже таблице дано возможное базовое решение транспортной проблемы и даны значения $v_1, v_2, v_3, u_1, u_2, u_3$.

Пункты отправления	Пункты назначения			Предло- жение	u_i
	1	2	3		
1	4 20	5 4	10	24	0
2	2	6 16	3 4	20	1
3	6	2	6 8	8	4
Спрос	20	20	12		
v_j	4	5	2		

Является ли это решение оптимальным? Обоснуйте ваш ответ.

Обратите внимание: продолжение вопроса на следующей странице

- (ה) В приведенной ниже таблице дано решение транспортной проблемы и даны значения $v_1, v_2, v_3, u_1, u_2, u_3$, соответствующие этому решению.

Пункты отправления	Пункты назначения			Предло- жение	u_i
	1	2	3		
1	10 20	25	30	20	10
2	10 30	22	14 20	50	10
3	16	20 40	20 20	60	16
Спрос	50	40	40		
v_j	0	4	4		

Перед вами утверждения i-iv, только одно из которых является верным. Перепишите в вашу тетрадь верное утверждение и обоснуйте ваш выбор.

- i Данное решение не является возможным решением.
- ii Данное решение является возможным базовым решением, но не является оптимальным.
- iii Данное решение является единственным оптимальным решением.
- iv Данное решение является оптимальным, но это не единственное оптимальное решение.

- (ו) В приведенной ниже таблице дана часть возможного базового решения транспортной проблемы, вычисленного методом северо-западного угла, и даны значения $v_1, v_2, v_3, u_1, u_2, u_3$, вычисленные в соответствии с этим решением.

Пункты отправления	Пункты назначения			Поставка	u_i
	1	2	3		
1	12 20	15	17	20	2
2	10	18 10	14	10	0
3	20	10 5	18 10	15	-8
Спрос	20	15	10		
v_j	10	18	26		

Что нужно добавить в эту таблицу, чтобы получить возможное базовое решение, соответствующее всем представленным в таблице данным?

Выберите среди предложенных вариантов i-iv правильный ответ и перепишите его в вашу тетрадь. Обоснуйте ваш ответ.

i $X_{12} = 0$.

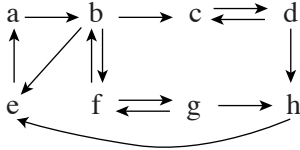
ii $X_{21} = 0$.

iii $X_{13} = 0$.

iv $X_{31} = 0$.

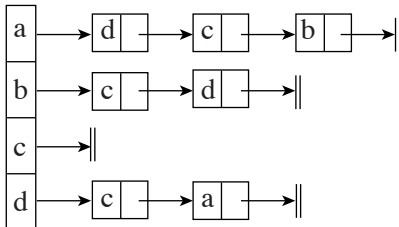
11. В данном вопросе три не связанных между собой пункта (א)-(ג).
Ответьте на вопросы всех пунктов.

(א) Перед вами ориентированный [מכוון] граф G .



- (i) Найдите компоненту/компоненты сильной связности [רכיב/רכיבי הקשירות החזקה (רק"ח/רק"חים)] в данном графе.
Для каждой найденной вами компоненты сильной связности [רק"ח] запишите группу её вершин [קודקודים].
- (ii) Является ли данный граф сильно связанным ориентированным графом [גרף קשיר מכוון בחזקה (קשיר היטב)]? Обоснуйте ваш ответ.

(ב) $G = (V, E)$ – ориентированный граф, где V – множество вершин графа, а E – множество ребер графа. Граф G представлен при помощи списка смежностей, приведенного здесь:



- (i) Примените алгоритм поиска в глубину (DFS) в данном графе, начиная с вершины a .
Начертите в вашей тетради полученное покрывающее дерево [העץ הכוסה] (DFS) / покрывающий лес [היער הכוסה] (DFS).
- (ii) Примените алгоритм поиска в глубину (DFS) в данном графе, начиная с вершины c .
Начертите в вашей тетради полученное покрывающее дерево (DFS) / покрывающий лес (DFS).
- (iii) Примените алгоритм поиска в ширину (BFS) в данном графе, начиная с вершины d .
Начертите в вашей тетради полученное покрывающее дерево (BFS) / покрывающий лес (BFS).

/продолжение на странице 25/

- (ג) $G = (V, E)$ – ориентированный граф, где V – множество вершин графа, а E – множество ребер графа. Граф G представлен при помощи списка смежностей.

Перед вами описание алгоритма, проверяющего, есть ли в графе G цикл [מגל], и выводящего на экран соответствующее сообщение. В алгоритме отсутствуют два выражения [ביטויים], обозначенные как (1) и (2).

Описание алгоритма

Шаг 1: Применить к графу G алгоритм поиска компонент сильной связности (SCC).

(Алгоритм возвращает компоненты сильной связности, найденные в графе).

Шаг 2: Если (1) (2), то
вывести: в графе нет циклов

Шаг 3: Иначе [ללא],
вывести: в графе есть цикл

Перед вами три возможных варианта выражения (1) и три возможных варианта выражения (2).

Варианты выражений:

(1) одно из:

- нет компоненты сильной связности, содержащей
- как минимум одна компонента сильной связности содержит
- каждая из компонент сильной связности содержит

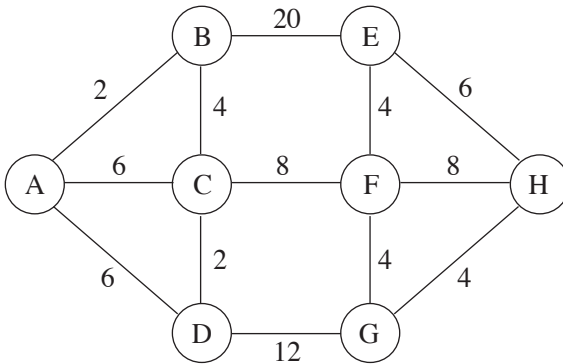
(2) одно из:

- одну единственную вершину
- как минимум две вершины
- все вершины графа

Выберите один из вариантов выражения (1) и один из вариантов выражения (2) так, чтобы алгоритм выполнял требуемые действия. Запишите в вашей тетради оба выбранных выражения.

12. В данном вопросе два не связанных между собой пункта. Ответьте на оба пункта (א) и (ב).

(א) Перед вами сеть [רשת].



Запишите в вашей тетради пять маршрутов минимальной длины от вершины A к вершине H внутри данной сети. Опишите каждый из этих маршрутов в отдельности.

(ב) G – связный неориентированный граф, определенный как $G = (V, E)$, где V – множество вершин графа, а E – множество ребер графа. Функция веса $W: E \rightarrow \mathbb{R}^+$ определяет вес (число) для каждого ребра в графе G .

X, Y, Z – вершины в графе G .

Пусть каждое ребро графа покрашено в красный или в синий цвет.

Перед вами описание алгоритма, проверяющего, проходят ли все маршруты минимальной длины из вершины X в вершину Y , составленные только из синих ребер, через вершину Z .

Если да – алгоритм возвращает значение “правда” (TRUE).

Иначе [אחרת] – алгоритм возвращает значение “ложь” (FALSE).

В алгоритме отсутствуют 5 выражений [ביטויים], обозначенных (1)-(5).

Описание алгоритма

Шаг 1: Удалим из графа все ребра (1) и получим новый граф $G_1 = (V, E_1)$.

Шаг 2: При помощи алгоритма Дейкстры вычислим длину самого короткого маршрута из X в Y в графе (2). Обозначим полученную длину M_1 .

- Шаг 3: Удалим из графа G_1 вершину (3) и все связанные с ней ребра и получим новый граф $G_2 = (V_2, E_2)$.
- Шаг 4: При помощи алгоритма Дейкстры вычислим длину самого короткого маршрута из X в Y в графе (4) . Обозначим полученную длину M_2 .
- Шаг 5: Если (5) , то вернуть значение “правда”.
Иначе [אחרת], вернуть значение “ложь”.

Перепишите в вашу тетрадь приведенную ниже таблицу и запишите пропущенные выражения.

	выражение (1) [ביטוי]
	выражение (2) [ביטוי]
	выражение (3) [ביטוי]
	выражение (4) [ביטוי]
	выражение (5) [ביטוי]

16. Имеется машина Тьюринга, вычисляющая функцию $f(m, n)$.

Машина находит меньшее из двух чисел m и n .

Машина получает в качестве ввода [קלט] два целых числа m и n , больших, чем 0. Два числа записаны на ленте как унарные числа

[מספרים אנריים] (унарное число m – это число, представленное как m единиц), между двумя числами записан символ $\#$.

Машина возвращает меньшее из двух чисел. Результат записывается на ленту как унарное число между двумя символами $\$$. Левый символ $\$$ будет записан вместо символа $\#$, и число будет записано справа от него.

В процессе работы машина может пользоваться символами a и b .

Например: для ввода $m = 3$ и $n = 5$ лента памяти перед началом вычисления будет иметь следующий вид:

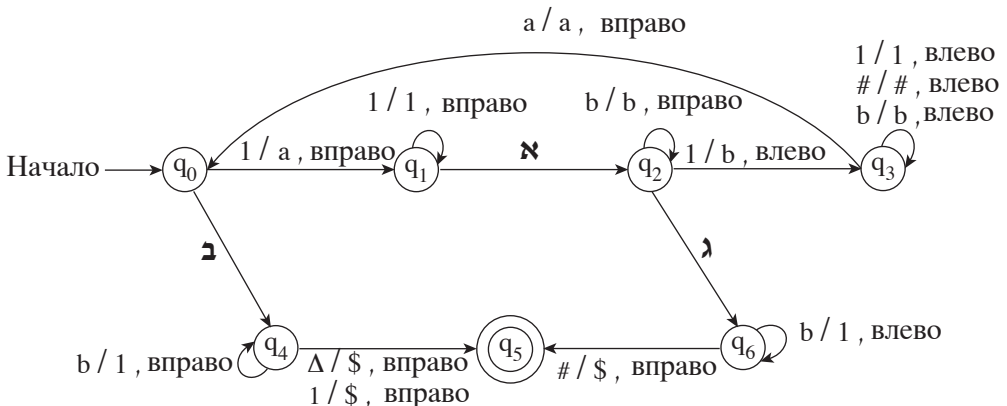
⊢	1	1	1	#	1	1	1	1	1	Δ	Δ	Δ	...
---	---	---	---	---	---	---	---	---	---	---	---	---	-----

После окончания вычислений, лента будет иметь следующий вид:

⊢	...	...	...	\$	1	1	1	\$	...
---	-----	-----	-----	----	---	---	---	----	-----

Примечание: ячейки, помеченные как ... – это ячейки, содержимое которых не важно.

Перед вами частичный чертеж этой машины.



(κ) На чертеже есть три перехода, обозначенные как κ-λ. Для этих переходов не указаны входные символы [סימני הקלט] и команды [ההוראות].

Перенесите чертеж в вашу тетрадь и дополните три недостающих перехода так, чтобы машина вычисляла функцию $f(m, n)$.

(б) Покажите процесс вычисления машины для ввода: $m = 1$, $n = 1$.

На каждом этапе записывайте состояние ленты, обозначайте место нахождения головки и указывайте, в каком из состояний $q_0 - q_6$ находится машина.

/продолжение на странице 30/

Объектно-ориентированное программирование [תכנות מונחה עצמים]

Если вы изучали этот курс и пишете на языке Java, ответьте на два вопроса из вопросов 17-20 (за каждый вопрос – 25 баллов).

17. Перед вами часть проекта, занимающегося транспортными средствами и содержащего следующие классы:

класс Vehicle	Представляет транспортные средства
класс Train	Представляет поезд, который является транспортным средством
класс Boat	Представляет лодку, которая является транспортным средством
класс Airplane	Представляет самолет, который является транспортным средством
класс TransportationCompany	Представляет компанию, владеющую разными видами транспортных средств

```
public class Vehicle
{
    private String type;           // среда (суша / вода / воздух)
    private String way;           // тип дороги (шоссе / рельсы / река / ...)
    private int maxSpeed;         // максимальная скорость

    public Vehicle(String type, String way, int maxSpeed)
    {
        this.type = type;
        this.way = way;
        this.maxSpeed = maxSpeed;
    }
}
```

Обратите внимание: продолжение вопроса на следующих страницах.

```
public class Train extends Vehicle
{
    private int numOfCarriages;      // количество вагонов
    public Train(int maxSpeed, int numOfCarriages)
    {
        super("land", "tracks", maxSpeed);
        this.numOfCarriages = numOfCarriages;
    }

    public void incNumOfCarriages(int n) // увеличивает на n число вагонов поезда
    {
        this.numOfCarriages = this.numOfCarriages + n;
    }
}
```

```
public class Boat extends Vehicle
{
    public Boat(String way, int maxSpeed)
    {
        super("water", way, maxSpeed);
    }
}
```

```
public class Airplane extends Vehicle
{
    private int maxHeight; // максимальная высота полета
    public Airplane(int maxSpeed, int maxHeight)
    {
        super("sky", "air", maxSpeed);
        this.maxHeight = maxHeight;
    }
}
```

```
public class TransportationCompany
{
    private Vehicle[] vehicles = new Vehicle[50]; // массив транспортных
                                                    средств компании
    private int counter = 0;                       // число имеющихся в
                                                    действительности транспортных средств
    public TransportationCompany()
    {
    }

    public void addVehicle (Vehicle v) // добавляет транспортное средство в массив
                                        транспортных средств компании

    {
        // считайте, что место для добавления
        транспортного средства выделено

        this.vehicles[counter] = v;
        this.counter++;
    }
}
```

- (א) Реализуйте на языке Java главный класс [מחלקה ראשית] Program, содержащий главный метод, который выполнит следующие задачи:
- (i) Создаст экземпляр класса “Транспортная компания” - **TransportationCompany**, с именем company1.
 - (ii) Добавит в company1 одну лодку и один поезд.
Выберите для атрибутов класса [תכונות] значения [ערכים] по Вашему желанию.

- (ב) В классе **TransportationCompany** определен следующий метод:

```
public void display()
{
    for (int i=0; i<this.counter; i++)
    {
        System.out.println((i+1) + ":" + this.vehicles[i]);
    }
}
```


Реализуйте на языке Java методы, которые обеспечат корректное выполнение метода `display()` так, чтобы для каждого транспортного средства на экран выводились все его атрибуты. Определите эти методы в максимальном соответствии с принципами объектно-ориентированного программирования (*encapsulation* - инкапсуляция, *inheritance* - наследование, *polymorphism* - полиморфизм).

Для каждого выполненного Вами метода укажите, к какому классу он принадлежит.

Нельзя менять метод `display()`.

- (ג) Напишите на языке Java метод, который получит целое число `n` и добавит `n` вагонов ко всем поездам, принадлежащим компании, которая владеет разными видами транспортных средств. Снабдите метод комментариями и укажите, в каком классе он должен быть определен.

Нельзя менять имеющиеся в проекте методы.

18. Перед вами классы **AA** и **BB** :

```
public class AA
{
    private String st;

    public AA()                { this.st = "excellent"; }
    public AA(String st)       { this.st = st;           }
    public String getSt()      { return this.st;         }
    public void setSt(String st) { this.st = st;         }
    public String toString()   { return "st = " + this.st; }
}

public class BB extends AA
{
    private int num;

    public BB()                { super(); this.num = 1; }
    public BB(int num , String st) { super(st); this.num = Math.abs(num); }
    public int getNum()         { return this.num;       }
    public void setNum(int num) { this.num = num;        }
    public String toString()     { return super.toString() + " num = " + this.num; }
}
```

- (א) В классе **AA** определите метод `isLike(Object obj)` , получающий экземпляр [פזז] `obj` типа `Object`. Если экземпляр `obj` принадлежит типу **AA** и содержимое строки `st` в `obj` такое же, как содержимое строки `st` в данном экземпляре класса, то метод вернет `true`, иначе – метод вернет `false`.
- (ב) В классе **BB** определите метод, замещающий [דורס] метод, определенный вами в пункте (א). Если экземпляр `obj` принадлежит типу **BB** и значение атрибута `num` в нем такое же, как значение атрибута `num` в данном экземпляре класса, то метод вернет `true`, иначе – метод вернет `false`.

(ג) Перед вами фрагмент главного метода [פעולה ראשית]:

```
AA a = new AA("excellent");  
BB b = new BB();  
a = b;  
if (a.isLike(b)) System.out.println(a);
```

Является ли данный фрагмент корректным [תקין]?

Если да – каким будет вывод данного фрагмента [פלט]? Запишите, какая версия метода isLike будет применена – версия из AA или из BB. Если нет – объясните, в чем заключается ошибка и когда она обнаружится: во время компиляции или во время выполнения [זמן ריצה].

(ד) Перед вами фрагмент главного метода:

```
AA aa = new AA();  
BB bb = new BB(2, "excellent");  
bb = aa;  
if (bb.isLike(aa)) System.out.println(bb);
```

Является ли данный фрагмент корректным [תקין]?

Если да – каким будет вывод данного фрагмента [פלט]? Запишите, какая версия метода isLike будет применена – версия из AA или из BB. Если нет – объясните, в чем заключается ошибка и когда она обнаружится: во время компиляции или во время выполнения [זמן ריצה].

(ה) Напишите внешний метод [פעולה חיצונית] longString, получающий массив [מערך] экземпляров типа Object. Метод возвращает строку, представляющую собой склейку [שרשרת] атрибутов st экземпляров типа AA в данном массиве следующим образом:

- Если у экземпляра есть только атрибут st, строка из атрибута st будет добавлена один раз.
- Если у экземпляра есть также атрибут num, то строка из атрибута st будет добавлена num раз.
- Если в массиве нет ни одного экземпляра типа AA, будет возвращена пустая строка.

19. Компьютерная фирма разработала программу для работы с последовательностями целых чисел.

Разработка проходила в несколько этапов.

Для каждой последовательности чисел интерес представляют:

- (1) Первый элемент последовательности с порядковым номером 1 .
- (2) Элемент последовательности с порядковым номером n .
- (3) Вывод на экран n первых элементов последовательности.

На **первом этапе** были разработаны два класса:

Арифметическая прогрессия [סדרה חשבונית] (**ASeq**) – последовательность, в которой разность [הפרש] между каждым элементом и его предшественником является постоянной величиной.

Геометрическая прогрессия [סדרה הנדסית] (**GSeq**) – последовательность, в которой частное [מנה] каждого элемента и его предшественника является постоянной величиной.

Перед Вами классы, разработанные на первом этапе:

```
public class ASeq
{
    private int first;
    private int difference;

    public ASeq(int first, int difference)
    {
        this.first = first;
        this.difference = difference;
    }

    public int theNElement(int n)
    {
        return this.first + (n-1) * this.difference;
    }
}
```

```
public void displayNElements(int n)
{
    System.out.print("The sequence elements: ");
    for (int i = 0; i < n - 1; i++)
        System.out.print(this.theNElement(i+1) + " , ");
    System.out.println(this.theNElement(n));
}
}
```

```
public class GSeq
{
    private int first;
    private int product;

    public GSeq(int first , int product)
    {
        this.first = first;
        this.product = product;
    }

    public int theNElement(int n)
    {
        return this.first * (int)Math.pow(this.product , n - 1);
    }

    public void displayNElements(int n)
    {
        System.out.print("The sequence elements: ");
        for (int i = 0 ; i < n - 1; i++)
            System.out.print(this.theNElement(i+1) + " , ");
        System.out.println(this.theNElement(n));
    }
}
```

- (א) Проследите за выполнением приведенного ниже фрагмента программы. В трассировке [מעקב] покажите экземпляр класса [עצם], который был создан, его атрибуты [תכונות] и вывод [פלט].

```
ASeq aSeq = new ASeq(2, 3);  
System.out.println(aSeq.theNElement(4));  
aSeq.displayNElements(5);
```

На **втором этапе** разработки было решено, что целесообразно разработать новый класс, описывающий постоянную последовательность [סדרה קבועה] **Sequence**, чтобы классы **ASeq** и **GSeq** наследовали [יירשו] от нового класса. В постоянной последовательности определено значение первого элемента, а все остальные элементы такие же, как первый элемент.

- (ב) Завершите разработку второго этапа в максимальном соответствии с принципами объектно-ориентированного программирования и в соответствии с инструкциями (i)-(ii):

- (i) Реализуйте [משש] полностью класс **Sequence**. Класс должен обрабатывать:
- (1) Первый элемент последовательности с порядковым номером 1 .
 - (2) Элемент последовательности с порядковым номером n .
 - (3) Вывод на экран n первых элементов последовательности.
- (ii) Реализуйте заново класс **ASeq** так, чтобы он наследовал от класса **Sequence**.

На **третьем этапе** разработки было решено расширить проект, содержащий три класса, разработанные на втором этапе (**Sequence**, **ASeq**, **GSeq**) так, чтобы для любой последовательности можно было выполнить метод, вычисляющий и возвращающий сумму n первых элементов последовательности.

Считайте, что класс **GSeq** был реализован заново так, чтобы он наследовал от класса **Sequence**.

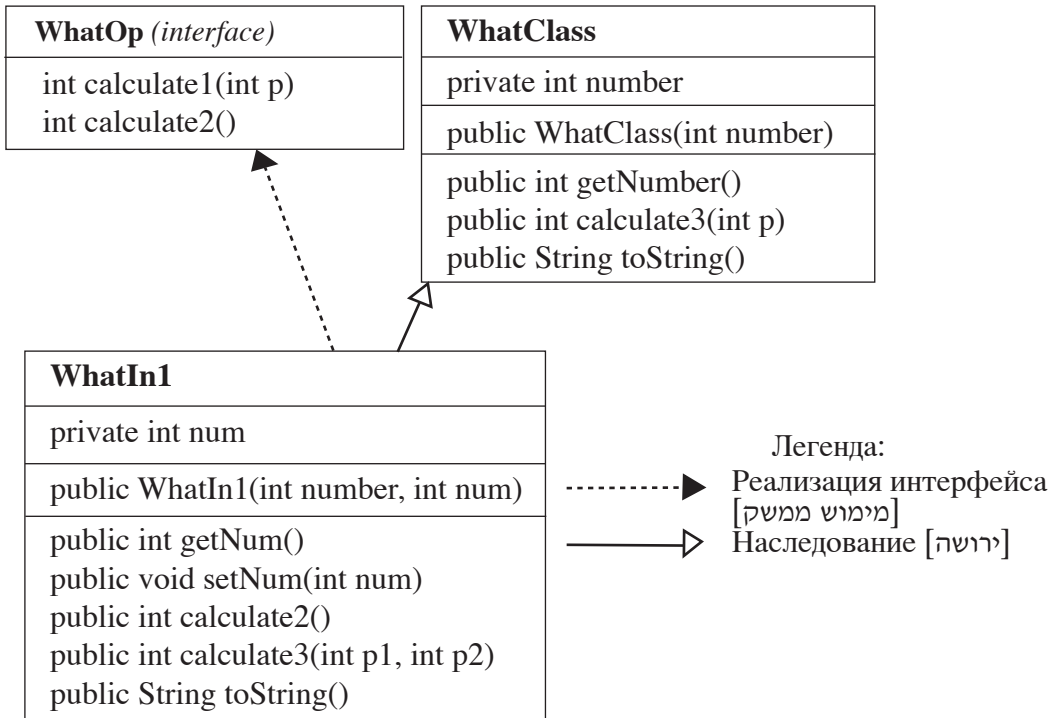
- (ג) Для каждого из классов **Sequence**, **ASeq**, **GSeq** напишите, нужно ли вносить в него изменения для того, чтобы проект отвечал требованиям разработки третьего этапа в максимальном соответствии с принципами объектно-ориентированного программирования. Если требуются изменения, детализируйте и реализуйте их [פרט וממש].

На **четвертом этапе** было решено разработать статический метод `check` [פעולה סטטית], получающий целое число `n` и два экземпляра последовательностей: один экземпляр класса **ASeq**, а другой - экземпляр класса **GSeq**.

Метод вычисляет сумму `n` первых элементов в каждой из двух последовательностей и возвращает:

- Символ 'A' – если сумма `n` первых элементов последовательности типа **ASeq** является большей из двух сумм.
- Символ 'G' – если сумма `n` первых элементов последовательности типа **GSeq** является большей из двух сумм.
- Символ 'E' – если сумма `n` первых элементов обеих последовательностей одинаковая.

(7) Реализуйте статический метод `check` в соответствии с требованиями, определенными на четвертом этапе разработки.



(א) При реализации метода `public int calculate1(int p)` в классе **WhatClass**, будет ли выполняться требование к его реализации в классе **WhatIn1**? Обоснуйте ваш ответ.

(ב) Перед вами реализация конструктора [פעולה בונה] класса **WhatIn1**:

```

public WhatIn1(int number, int num)
{
    this.number = number;
    this.num = num;
}
    
```

Корректен ли [תקין] данный конструктор? Если да – опишите выполнение конструктора, если нет – исправьте конструктор (не разрешается изменять ничего, за исключением самого конструктора). Запишите в вашей тетради исправленный конструктор.

Обратите внимание: продолжение вопроса на следующей странице

/продолжение на странице 41/

- (ג) В классе **WhatIn1** реализован метод `calculate2()` , возвращающий целое значение среднего арифметического атрибутов экземпляра класса [תכונות העצם]:

```
public int calculate2()
{    return (int)((this.number + this.num)/2);    }
```

Корректен ли [תקין] данный метод? Если да – опишите выполнение метода, если нет – исправьте метод (не разрешается изменять ничего, за исключением самого метода). Запишите в вашей тетради исправленный метод.

- (ד) Перед вами частичное описание класса **WhatIn2** :

class WhatIn2 extends WhatIn1
private int sum
public WhatIn2(int number, int num, int sum)
public int calculate3(int p1, int p2, int p3)

- (1) Можно ли полагаться на конструктор по умолчанию [ברירת מחדל] вместо того, чтобы определять конструктор в классе **WhatIn2** ? Обоснуйте ваш ответ.
- (2) Метод `calculate3` реализован в трех классах **WhatClass** , **WhatIn1** и **WhatIn2** следующим образом:

class WhatClass	public int calculate3(int p) { return this.number * p; }
class WhatIn1	public int calculate3(int p1, int p2) { return this.calculate3(p1) + this.num * p2 * p2; }
class WhatIn2	public int calculate3(int p1, int p2, int p3) { return this.calculate3(p1, p2) + this.sum * p3 * p3 * p3;}

Считайте, что `obj` – экземпляр класса **WhatIn2** и его атрибуты имеют следующие значения:

```
number = 1
num = 2
sum = 3
```

Перед вами команда из главного метода [הפעולה הראשית]:

```
System.out.println(obj.calculate3(1000, 100, 10));
```

Проследите за выполнением [הראה את המעקב] данной команды. При ответе укажите вызовы методов [זימון פעולות] и значения атрибутов экземпляра класса. Запишите результирующий вывод [פלט].

/продолжение на странице 42/

Объектно-ориентированное программирование
[תכנות מונחה עצמים]

Если вы изучали этот курс и пишете на языке C#, ответьте на два вопроса из вопросов 21-24 (за каждый вопрос – 25 баллов).

21. Перед вами часть проекта, занимающегося транспортными средствами и содержащего следующие классы:

класс Vehicle	Представляет транспортные средства
класс Train	Представляет поезд, который является транспортным средством
класс Boat	Представляет лодку, которая является транспортным средством
класс Airplane	Представляет самолет, который является транспортным средством
класс TransportationCompany	Представляет компанию, владеющую разными видами транспортных средств

```
public class Vehicle
```

```
{  
    private string type;      // среда (суша / вода / воздух)  
    private string way;      // тип дороги (шоссе / рельсы / река / ...)  
    private int maxSpeed;    // максимальная скорость  
  
    public Vehicle(string type, string way, int maxSpeed)  
    {  
        this.type = type;  
        this.way = way;  
        this.maxSpeed = maxSpeed;  
    }  
}
```

```
public class Train : Vehicle
{
    private int numOfCarriages; // количество вагонов
    public Train(int maxSpeed, int numOfCarriages) : base("land", "tracks", maxSpeed)
    {
        this.numOfCarriages = numOfCarriages;
    }
    public void IncNumOfCarriages(int n) // увеличивает на n число вагонов в
                                         поезде
    {
        this.numOfCarriages = this.numOfCarriages + n;
    }
}
```

```
public class Boat : Vehicle
{
    public Boat(string way, int maxSpeed) : base("water", way, maxSpeed)
    {

    }
}
```

```
public class Airplane : Vehicle
{
    private int maxHeight; // максимальная высота полета
    public Airplane(int maxSpeed, int maxHeight) : base("sky", "air", maxSpeed)
    {
        this.maxHeight = maxHeight;
    }
}
```

```
public class TransportationCompany
{
    private Vehicle[] vehicles = new Vehicle[50]; // массив транспортных
                                                    средств компании

    private int counter = 0; // число имеющихся в
                            // действительности транспортных средств

    public TransportationCompany()
    {
    }

    public void addVehicle (Vehicle v) // добавляет транспортное средство в массив
                                        // считайте, что место для добавления
                                        // транспортного средства выделено
    {
        this.vehicles[counter] = v;
        this.counter++;
    }
}
```

- (א) Реализуйте на языке C# главный класс [מחלקה ראשית] Program, содержащий главный метод, который выполнит следующие задачи:
- (i) Создаст экземпляр класса “Транспортная компания” – **TransportationCompany**, с именем company1.
 - (ii) Добавит в company1 одну лодку и один поезд.
Выберите для атрибутов класса [תכונות] значения [ערכים] по Вашему желанию.

- (ב) В классе **TransportationCompany** определен следующий метод:

```
public void Display()
{
    for (int i=0; i<this.counter; i++)
    {
        Console.WriteLine ((i+1) + ":" + this.vehicles[i]);
    }
}
```

Реализуйте на языке C# методы, которые обеспечат корректное выполнение метода Display() так, чтобы для каждого транспортного средства на экран выводились все его атрибуты. Определите эти методы в максимальном соответствии с принципами объектно-ориентированного программирования (encapsulation - инкапсуляция, inheritance - наследование, polymorphism - полиморфизм).

Для каждого выполненного Вами метода укажите, к какому классу он принадлежит.

Нельзя менять метод Display().

- (ג) Напишите на языке C# метод, который получит целое число n и добавит n вагонов ко всем поездам, принадлежащим компании, которая владеет разными видами транспортных средств. Снабдите метод комментариями и укажите, в каком классе он должен быть определен.

Нельзя менять имеющиеся в проекте методы.

22. Перед вами классы **AA** и **BB**:

```
public class AA
{
    private string st;

    public AA() { this.st = "excellent"; }
    public AA(string st) { this.st = st; }
    public string GetSt() { return this.st; }
    public void SetSt (string st) { this.st = st; }
    public override string ToString() { return "st = " + this.st; }
}

public class BB : AA
{
    private int num;

    public BB() : base() { this.num = 1; }
    public BB(int num, string st) : base(st) { this.num = Math.Abs(num); }
    public int GetNum() { return this.num; }
    public void SetNum(int num) { this.num = num; }
    public override string ToString() { return base.ToString() + " num = "
                                      + this.num; }
}
```

(א) В классе **AA** определите метод `IsLike(Object obj)`, получающий экземпляр [פזז] `obj` типа `Object`. Если экземпляр `obj` принадлежит типу **AA** и содержимое строки `st` в `obj` такое же, как содержимое строки `st` в данном экземпляре класса, то метод вернет `true`, иначе – метод вернет `false`.

(ב) В классе **BB** определите метод, замещающий [סדר] метод, определенный вами в пункте (א). Если экземпляр `obj` принадлежит типу **BB** и значение атрибута `num` в нем такое же, как значение атрибута `num` в данном экземпляре класса, то метод вернет `true`, иначе – метод вернет `false`.

/продолжение на странице 47/

(ג) Перед вами фрагмент главного метода [פעולה ראשית]:

```
AA a = new AA("excellent");  
BB b = new BB();  
a = b;  
if (a.IsLike(b)) Console.WriteLine(a);
```

Является ли данный фрагмент корректным [תקין]?

Если да – каким будет вывод данного фрагмента [פלט]? Запишите, какая версия метода IsLike будет применена – версия из AA или из BB. Если нет – объясните, в чем заключается ошибка и когда она обнаружится: во время компиляции или во время выполнения [זמן ריצה].

(ד) Перед вами фрагмент главного метода:

```
AA aa = new AA();  
BB bb = new BB(2, "excellent");  
bb = aa;  
if (bb.IsLike(aa)) Console.WriteLine(bb);
```

Является ли данный фрагмент корректным [תקין]?

Если да – каким будет вывод данного фрагмента [פלט]? Запишите, какая версия метода IsLike будет применена – версия из AA или из BB. Если нет – объясните, в чем заключается ошибка и когда она обнаружится: во время компиляции или во время выполнения [זמן ריצה].

(ה) Напишите внешний метод [פעולה חיצונית] LongString, получающий массив [מערך] экземпляров типа Object. Метод возвращает строку, представляющую собой склейку [שרשרת] атрибутов st экземпляров типа AA в данном массиве следующим образом:

- Если у экземпляра есть только атрибут st, строка из атрибута st будет добавлена один раз.
- Если у экземпляра есть также атрибут num, то строка из атрибута st будет добавлена num раз.
- Если в массиве нет ни одного экземпляра типа AA, будет возвращена пустая строка.

23. Компьютерная фирма разработала программу для работы с последовательностями целых чисел.

Разработка проходила в несколько этапов.

Для каждой последовательности чисел интерес представляют:

- (1) Первый элемент последовательности с порядковым номером 1 .
- (2) Элемент последовательности с порядковым номером n .
- (3) Вывод на экран n первых элементов последовательности.

На **первом этапе** были разработаны два класса:

Арифметическая прогрессия [סדרה חשבונית] (**ASeq**) – последовательность, в которой разность [הפרש] между каждым элементом и его предшественником является постоянной величиной.

Геометрическая прогрессия [סדרה הנדסית] (**GSeq**) – последовательность, в которой частное [מנה] каждого элемента и его предшественника является постоянной величиной.

Перед Вами классы, разработанные на первом этапе:

```
public class ASeq
{
    private int first;
    private int difference;

    public ASeq(int first, int difference)
    {
        this.first = first;
        this.difference = difference;
    }

    public int TheNElement(int n)
    {
        return this.first + (n - 1) * this.difference;
    }
}
```



```

public void DisplayNElements(int n)
{
    Console.Write("The sequence elements: ");
    for (int i = 0; i < n - 1; i++)
        Console.Write(this.TheNElement(i+1) + " , ");
    Console.WriteLine(this.TheNElement(n));
}
}

public class GSeq
{
    private int first;
    private int product;

    public GSeq(int first, int product)
    {
        this.first = first;
        this.product = product;
    }

    public int TheNElement(int n)
    {
        return this.first * (int)Math.Pow(this.product , n-1);
    }

    public void DisplayNElements(int n)
    {
        Console.Write("The sequence elements: ");
        for (int i = 0; i < n - 1; i++)
            Console.Write(this.TheNElement(i+1) + " , ");
        Console.WriteLine (this.TheNElement(n));
    }
}

```

- (א) Проследите за выполнением приведенного ниже фрагмента программы. В трассировке [מעקב] покажите экземпляр класса [עצם], который был создан, его атрибуты [תכונות] и вывод [פלט].

```
ASeq aSeq = new ASeq(2, 3);  
Console.WriteLine(aSeq.TheNElement(4));  
aSeq.DisplayNElements(5);
```

На **втором этапе** разработки было решено, что целесообразно разработать новый класс, описывающий постоянную последовательность [סדרה קבועה] **Sequence**, чтобы классы **ASeq** и **GSeq** наследовали [יירשו] от нового класса. В постоянной последовательности определено значение первого элемента, а все остальные элементы такие же, как первый элемент.

- (ב) Завершите разработку второго этапа в максимальном соответствии с принципами объектно-ориентированного программирования и в соответствии с инструкциями (i)-(ii):
- (i) Реализуйте [משמ] полностью класс **Sequence**. Класс должен обрабатывать:
- (1) Первый элемент последовательности с порядковым номером 1.
 - (2) Элемент последовательности с порядковым номером n.
 - (3) Вывод на экран n первых элементов последовательности.
- (ii) Реализуйте заново класс **ASeq** так, чтобы он наследовал от класса **Sequence**.

На **третьем этапе** разработки было решено расширить проект, содержащий три класса, разработанные на втором этапе (**Sequence**, **ASeq**, **GSeq**) так, чтобы для любой последовательности можно было выполнить метод, вычисляющий и возвращающий сумму n первых элементов последовательности.

Считайте, что класс **GSeq** был реализован заново так, чтобы он наследовал от класса **Sequence**.

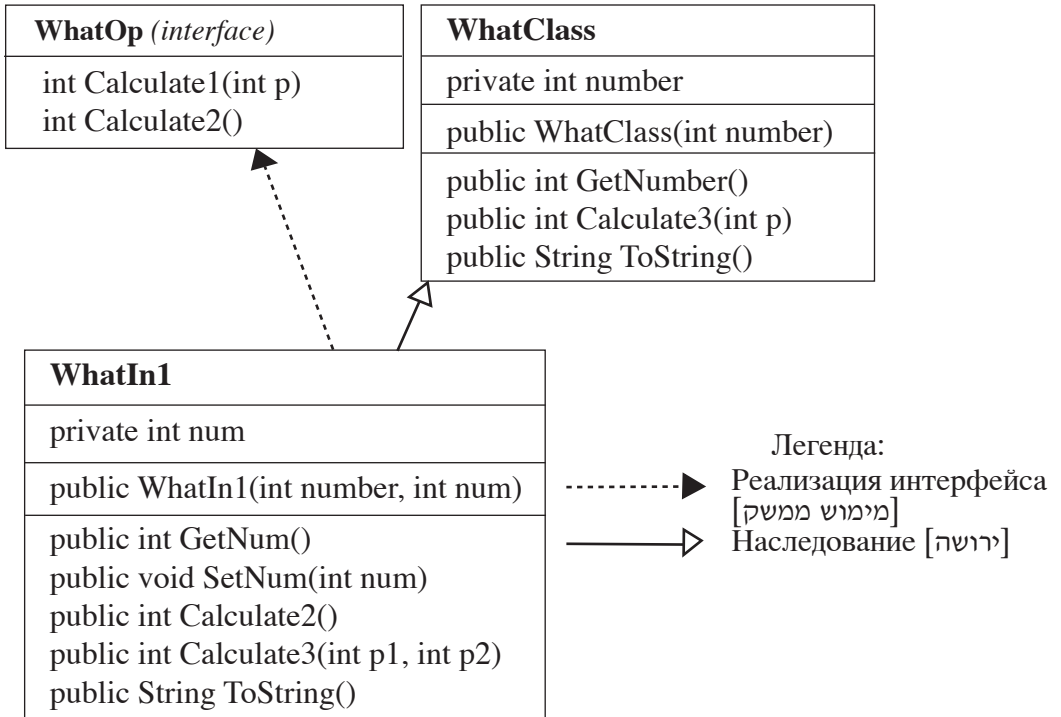
- (ג) Для каждого из классов **Sequence**, **ASeq**, **GSeq** напишите, нужно ли вносить в него изменения для того, чтобы проект отвечал требованиям разработки третьего этапа в максимальном соответствии с принципами объектно-ориентированного программирования. Если требуются изменения, детализируйте и реализуйте их [פרט וממש].

На **четвертом этапе** было решено разработать статический метод `Check` [פעולה סטטית], получающий целое число `n` и два экземпляра последовательностей: один экземпляр класса **ASeq**, а другой - экземпляр класса **GSeq**.

Метод вычисляет сумму `n` первых элементов в каждой из двух последовательностей и возвращает:

- Символ 'A' – если сумма `n` первых элементов последовательности типа **ASeq** является большей из двух сумм.
 - Символ 'G' – если сумма `n` первых элементов последовательности типа **GSeq** является большей из двух сумм.
 - Символ 'E' – если сумма `n` первых элементов обеих последовательностей одинаковая.
- (7) Реализуйте статический метод `Check` в соответствии с требованиями, определенными на четвертом этапе разработки.

24. Перед вами частичная диаграмма UML :



(א) При реализации метода `public int Calculate1(int p)` в классе **WhatClass** будет ли выполняться требование к его реализации в классе **WhatIn1** ? Обоснуйте ваш ответ.

(ב) Перед вами реализация конструктора [פעולה בונה] класса **WhatIn1** :

```

public WhatIn1(int number, int num)
{
    this.number = number;
    this.num = num;
}
    
```

Корректен ли [תקין] данный конструктор? Если да – опишите выполнение конструктора, если нет – исправьте конструктор (не разрешается изменять ничего, за исключением самого конструктора). Запишите в вашей тетради исправленный конструктор.

- (ג) В классе **WhatIn1** реализован метод `Calculate2()`, возвращающий целое значение среднего арифметического атрибутов экземпляра класса [תכונות העצם]:

```
public int Calculate2()  
{    return (int)((this.number + this.num)/2); }
```

Корректен ли [תקין] данный метод? Если да – опишите выполнение метода, если нет – исправьте метод (не разрешается изменять ничего, за исключением самого метода). Запишите в вашей тетради исправленный метод.

- (ד) Перед вами частичное описание класса **WhatIn2** :

class WhatIn2: WhatIn1
private int sum
public WhatIn2(int number, int num, int sum)
public int Calculate3(int p1, int p2, int p3)

- (1) Можно ли полагаться на конструктор по умолчанию [ברירת מחדל] вместо того, чтобы определять конструктор в классе **WhatIn2**? Обоснуйте ваш ответ.
- (2) Метод `Calculate3` реализован в трех классах **WhatClass**, **WhatIn1** и **WhatIn2** следующим образом:

class WhatClass	public int Calculate3(int p) { return this.number * p; }
class WhatIn1	public int Calculate3(int p1, int p2) { return this.Calculate3(p1) + this.num * p2 * p2; }
class WhatIn2	public int Calculate3(int p1, int p2, int p3) { return this.Calculate3(p1, p2) + this.sum * p3 * p3 * p3; }

Считайте, что `obj` – экземпляр класса **WhatIn2** и его атрибуты имеют следующие значения:

number = 1

num = 2

sum = 3

Перед вами команда из главного метода [הפעולה הראשית]:

```
Console.WriteLine(obj.Calculate3(1000, 100, 10));
```

Проследите за выполнением [הראה את המעקב] данной команды. При ответе укажите вызовы методов [זימון פעולות] и значения атрибутов экземпляра класса. Запишите результирующий вывод [פלט].

Желаем успеха!

בהצלחה!