

Компьютеры

מדעי המחשב

Указания экзаменующимся

- а. Продолжительность экзамена: три часа.
 б. Строение вопросника и ключ к оценке:
 в этом вопроснике три раздела.
 Раздел первый: $(1 \times 10) + (1 \times 15)$ – 25 баллов
 Раздел второй: (2×25) – 50 баллов
 Раздел третий: (1×25) – 25 баллов
 Всего – 100 баллов

הוראות לנבחן

- א. משך הבחינה: שלוש שעות.
 ב. מבנה השאלון ומפתח ההערכה:
 בשאלון זה שלושה פרקים.
 פרק ראשון – $(10 \times 1) + (15 \times 1)$ – 25 נקודות
 פרק שני – (25×2) – 50 נקודות
 פרק שלישי – (25×1) – 25 נקודות
 סה"כ – 100 נקודות
 ג. חומר עזר מותר בשימוש:
 כל חומר עזר, חוץ ממחשבון שיש בו אפשרות תכנות.

г. Особые указания:

1. Все программы, которые вы должны написать на языке программирования в первом и втором разделах, пишите только на одном языке – Java или C# .
 2. На внешней обложке тетради укажите язык, который вы используете: Java или C# .
 3. На внешней обложке тетради укажите название курса, который **вы изучали**, одного из четырех следующих курсов: “Компьютерные системы и ассемблер”, “Введение в исследование операций”, “Вычислительные модели”, “Объектно-ориентированное программирование”.

ד. הוראות מיוחדות:

1. את כל התכניות שאתה נדרש לכתוב בשפת מחשב בפרקים הראשון והשני כתוב בשפה אחת בלבד – Java או C# .
 2. רשום על הכריכה החיצונית של המחברת באיזו שפה אתה כותב – Java או C# .
 3. רשום על הכריכה החיצונית של המחברת את שם המסלול שלמדת. המסלול הוא אחד מארבעת המסלולים האלה: מערכות מחשב ואסמבלי, מבוא לחקר ביצועים, מודלים חישוביים, תכנות מונחה עצמים.

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

הערה: בתכניות שאתה כותב לא יורדו לך נקודות, אם תכתוב אות גדולה במקום אות קטנה או להפך.

כתוב במחברת הבחינה בלבד. רשום “טייטה” בראש כל עמוד המשמש טייטה.

כתיבת טייטה בדפים שאינם במחברת הבחינה עלולה לגרום לפסילת הבחינה.

Пишите только в экзаменационной тетради. Напишите слово «טייטה» 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!

Обратите внимание: продолжение экзамена на странице 3

Вопросы

В этом вопроснике три раздела.

Вы должны ответить на вопросы трех этих разделов согласно инструкциям каждого раздела.

Примечание: в каждом вопросе, в котором требуется ввод,

нет необходимости проверять корректность вводимых данных.

Для решающих на языке Java: в каждом вопросе, в котором требуется ввод,

считайте, что в программе написана команда:

```
Scanner input = new Scanner (System.in);
```

РАЗДЕЛ ПЕРВЫЙ (25 баллов)

Ответьте на вопрос 1 – обязательный (10 баллов).

1. Напишите внешний метод `above` на языке Java или `Above` на языке C#, получающий массив `arr` целого типа и число `num` целого типа. Метод вернет первый индекс такой ячейки, что сумма чисел от начала массива до этой ячейки (включительно) больше, чем число `num`. Если такой ячейки нет, метод вернет `-1`.

Пример: в приведенном ниже массиве `arr` для числа `num`, равного 9, метод вернет число 4, так как это первый индекс ячейки, сумма чисел от начала массива до которой (включительно) больше, чем 9.

0	1	2	3	4	5
3	-2	6	2	1	3



Дополнительный пример: в приведенном ниже массиве `arr` для числа `num`, равного 11, метод вернет число `-1`, так как в массиве нет ячейки, сумма чисел от начала массива до которой (включительно) больше, чем 11.

0	1	2	3	4	5
1	2	-2	0	7	3

Ответьте на один из вопросов 2-3 (15 баллов).

2. Дан класс **Weight** – вес, имеющий два атрибута [תכונות]:

kilo – число килограммов, целое число, принимающее значение от 0 и выше.

gram – число граммов, целое число, принимающее значение от 0 до 999 (включительно).

Считайте, что для каждого из атрибутов в данном классе определены методы get и set на языке Java и методы Get и Set на языке C# .

Перед вами интерфейс класса **Weight** на языках Java и C# :

Заголовок метода	Описание метода
public Weight ()	Конструктор класса [פונקציה בונה], присваивающий значение 0 атрибутам kilo и gram.
public Weight (int kilo, int gram)	Конструктор класса, получающий значение килограммов и граммов. Считайте, что число килограммов в переменной kilo больше или равно 0 , а число граммов в переменной gram находится в промежутке от 0 до 999 (включительно).
public Weight (int totalGram)	Конструктор класса, получающий значение в граммах. Число граммов в переменной totalGram может быть больше, чем 999 . Метод вычислит и присвоит атрибуту kilo число килограммов, а атрибуту gram –число граммов. <u>Пример:</u> если число граммов в переменной totalGram равно 3215 , то атрибут kilo получит значение 3 , а атрибут gram получит значение 215 . Считайте, что число граммов в переменной totalGram больше или равно 0 .
на языке Java — public void add (Weight other) на языке C# — public void Add (Weight other)	Метод, получающий другой объект типа Weight и добавляющий значение его веса к весу имеющегося объекта. Считайте, что объект other не равен null .
на языке Java — public boolean less (Weight other) на языке C# — public bool Less (Weight other)	Метод, получающий другой объект типа Weight и возвращающий true , если вес имеющегося объекта <u>меньше</u> , чем вес другого (полученного) объекта . Считайте, что объект other не равен null .

Обратите внимание: продолжение вопроса на следующей странице.

/продолжение на странице 5/

- (א) Реализуйте два конструктора класса **Weight** : `public Weight (int kilo, int gram)` и `public Weight (int totalGram)` (нет необходимости реализовывать конструктор без параметров).
- (ב) Реализуйте методы `add` и `less` на языке Java или `Add` и `Less` на языке C# .
- (ג) Дан класс **AllWeights** с одним атрибутом: `arr` – массив типа **Weight** .

Перед вами метод из интерфейса класса **AllWeights** на языке Java и на языке C# :

Заголовок метода	Описание метода
на языке Java — <code>public Weight sum ()</code> на языке C# — <code>public Weight Sum ()</code>	Метод возвращает объект типа Weight , вес которого равен сумме веса всех элементов этого массива. <u>Примечание:</u> запрещено изменять элементы массива.

Реализуйте метод `sum` на языке Java или метод `Sum` на языке C# .

Считайте, что ячейки массива не равны `null` .

Внимание: можно использовать любые методы из интерфейса класса **Weight** .

3. В хостеле "Доброй ночи" 3 этажа, 1-3. Всего в хостеле 200 комнат.

В хостеле два типа комнат: одноместные и двухместные. Одноместная комната стоит 50 шекелей за ночь, а двухместная – 100 шекелей за ночь.

В компьютерной системе хостела определен класс **Room** – комната, с тремя атрибутами:

- roomNum – номер комнаты, целое число из трех цифр.

Первая цифра номера обозначает этаж, на котором находится комната.

Например: комната номер 231 находится на этаже номер 2, а комната номер 145 находится на этаже номер 1.

- roomType – тип комнаты, целое число: 1 – одноместная комната, 2 – двухместная комната.
- nightsReserved – количество ночей, на которое комната заказана, начиная с данного момента, целое число.

Если nightsReserved равно 0, то комната свободна, а если больше, чем 0, то комната занята.

Считайте, что для каждого атрибута класса определены методы get и set на языке Java и методы Get и Set на языке C#.

- (x) Напишите метод income на языке Java или Income на языке C# в классе **Room**, возвращающий прибыль, получаемую от сдачи комнаты. Прибыль рассчитывается как число заказанных ночей, умноженное на цену комнаты.

- (z) В компьютерной системе хостела также определен класс **Hostel** – хостел, с одним атрибутом:

allRooms – массив из 200 объектов типа **Room** (комнаты в массиве никак не отсортированы).

Считайте, что ячейки массива не равны null.

Обратите внимание: продолжение вопроса на следующей странице.

Перед вами частичный интерфейс класса **Hostel** на языках Java и C# .

Заголовок метода	Описание метода
на языке Java — <code>public int orderRoom (int type, int nights)</code> на языке C# — <code>public int OrderRoom (int type, int nights)</code>	Метод находит первую комнату в массиве allRooms , которая соответствует следующим требованиям: свободна и имеет тип type . Метод запишет в соответствующем атрибуте комнаты требуемое количество ночей, указанное в переменной nights , и вернет номер комнаты. Если в массиве нет свободной комнаты требуемого типа, метод вернет – 1 .
на языке Java — <code>public int [] floorIncome ()</code> на языке C# — <code>public int [] FloorIncome ()</code>	Метод вернет массив из 3 элементов, содержащий суммы прибыли от всех комнат на каждом из этажей хостела. Ячейка с индексом 0 содержит прибыль от комнат на этаже 1 , ячейка с индексом 1 содержит прибыль от комнат на этаже 2 , ячейка с индексом 2 содержит прибыль от комнат на этаже 3 .

- (1) Реализуйте метод orderRoom на языке Java или OrderRoom на языке C# в классе **Hostel**.
 - (2) Реализуйте метод floorIncome на языке Java или FloorIncome на языке C# в классе **Hostel**.
- Вы должны использовать метод, написанный вами в пункте (х).

РАЗДЕЛ ВТОРОЙ (50 баллов)

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

Ответьте на два из вопросов 4-6 (за каждый вопрос – 25 баллов)

4. (א) Напишите внешний метод [פעולה חיצונית] isExist языке Java или IsExist на языке C#. Метод получит число num целого типа со значением от 0 до 9 (включительно), и стек stk целого типа. Метод вернет true, если в стеке имеется число, цифра единиц в котором равно числу num, иначе – метод вернет false. Считайте, что числа в стеке stk не отрицательные. Пример: для числа num, равного 8, и приведенного ниже стека stk:

162	← голова стека
251	
568	
77	

метод вернет true, так как в стеке находится число 568, в котором цифра единиц 8. Примечание: после завершения выполнения метода строение стека должно быть сохранено.

- (ב) Только для решения данного пункта можно использовать приведенный ниже метод, не реализуя его.

Заголовок метода	Описание метода
на языке Java — public static Stack <Integer> clone (Stack <Integer> s)	Метод получает стек целого типа и возвращает точную копию этого стека, не изменяя исходный стек.
на языке C# — public static Stack <int> Clone (Stack <int> s)	

Определим: значимой цифрой числа является его самая левая цифра.

Например, цифра 3 – значимая цифра числа 32, а цифра 5 – значимая цифра числа 541.

Напишите внешний метод allExist на языке Java или AllExist на языке C#, получающий непустой стек stk целого типа. Метод вернет true, если все значимые цифры чисел в стеке появляются в цифрах единиц каких-либо чисел в стеке, иначе – метод вернет false.

Считайте, что числа в стеке stk не отрицательные.

Обратите внимание: продолжение вопроса на следующей странице.

/продолжение на странице 9/

Пример: для приведенного ниже стека `stk` :

122	← голова стека
251	
565	
12334	
28	
7	

метод вернет `true` , так как все значимые цифры чисел в стеке – 1 , 2 , 5 , 7 – появляются в цифрах единиц чисел в стеке.

Дополнительный пример: для приведенного ниже стека `stk` :

1223	← голова стека
245	
521	
12334	

метод вернет `false` , так как из всех значимых цифр чисел в стеке – 1 , 2 , 5 – цифра 2 не появляется в цифре единиц ни в одном из чисел в стеке.

Примечание: можно использовать метод, написанный вами для пункта (ж).

5. Для проведения марафона определен класс **Competitor** , представляющий участника, добежавшего марафон до конца.

В классе три атрибута [לוחות]:

- minutes – число минут, потребовавшихся участнику для завершения марафона (число минут может быть больше, чем 60), целого типа.
- seconds – число секунд, потребовавшихся участнику для завершения марафона (число секунд может быть до 59 включительно), целого типа.
- name – имя участника, типа строка.

Считайте, что для каждого атрибута определены методы get и set на языке Java и методы Get и Set на языке C# .

Кроме того, определен класс **Race** , хранящий набор всех участников, добежавших марафон до конца. Количество таких участников неизвестно.

Считайте, что нет двух участников, пробежавших марафон за одно и то же время.

Перед вами частичный интерфейс класса **Race** на языках Java и C# :

Заголовок метода	Описание метода	Временная сложность выполнения
На языке Java — public void add (Competitor x) На языке C# — public void Add (Competitor x)	Метод получает объект типа Competitor и добавляет его к набору участников. <u>Внимание</u>: следующий метод интерфейса (rank на языке Java или Rank на языке C#) и требования к временной сложности его выполнения влияют на то, каким образом должен быть реализован данный метод.	O(n)
На языке Java — public String rank (int x) На языке C# — public string Rank (int x)	Метод получает место участника и возвращает имя участника, занявшего это место в марафоне. <u>Пояснение</u>: место 1 занимает участник, пробежавший марафон за самое <u>короткое</u> время; следующий по времени участник занимает место 2 , и так далее. Считайте, что имеется участник, занявший требуемое место. <u>Примечание</u>: запрещается удалять элементы из набора.	O(n)

Обратите внимание: продолжение вопроса на следующей странице.

/продолжение на странице 11/

Обратите внимание:

- для каждого метода определены требования к временной сложности его выполнения [סיבוכיות זמן ריצה].
- n – число участников в наборе.
- **вы должны удовлетворить требования к временной сложности выполнения методов.**

Для пунктов (א)-(ב) вы должны использовать соответствующую структуру данных, отвечающую требованиям вопроса.

Вы можете использовать любые структуры данных, которые вы изучали.

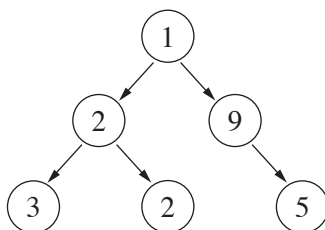
Можно писать методы в классе **Competitor** .

(א) Напишите атрибуты класса **Race** .

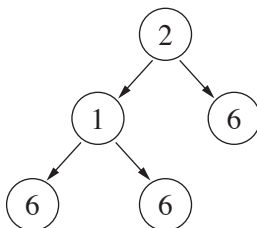
(ב) Реализуйте методы класса **Race**, описанные в интерфейсе класса в данном вопросе.

6. Определим: "дерево чисел" – это двоичное дерево целого типа, каждый узел которого содержит цифру от 1 до 9 (включительно) и каждый маршрут в дереве от корня до листа представляет собой число: лист соответствует числу единиц, уровень над ним – числу десятков, и так далее, до корня дерева.

Пример: в приведенном ниже дереве чисел представлены числа 123 , 122 , 195 .



Дополнительный пример: в приведенном ниже дереве чисел представлены числа 216 , 216 , 26 .



Напишите внешний метод `printAll` на языке Java или `PrintAll` на языке C# . Метод получит дерево чисел `tree` целого типа и выведет на экран все числа, представленные маршрутами в дереве.

Если `tree` равно `null` , метод не выведет на экран ничего.

Примечание: порядок, в котором числа выводятся на экран, не имеет значения.

РАЗДЕЛ ТРЕТИЙ (25 баллов)

В этом разделе вопросы четырёх разных курсов:

- Компьютерные системы и ассемблер [מערכות מחשב ואסמבלי], стр. 13-15.
- Введение в исследование операций [מבוא לחקר ביצועים], стр. 16-20.
- Вычислительные модели [מודלים חישוביים], стр. 21
- Объектно-ориентированное программирование [תכנות מונחה עצמים] на языке Java, стр. 22-25;
объектно-ориентированное программирование на языке C#, стр. 26-29.

Ответьте на один вопрос из того курса, который вы изучали.

Компьютерные системы и ассемблер [מערכות מחשב ואסמבלי]

Если вы изучали этот курс, ответьте на один из вопросов 7-8 (25 баллов).

7. В этом вопросе три пункта (א)-(ג). Между ними нет связи. Ответьте на вопросы всех этих пунктов.

(א) Перед вами фрагмент программы на ассемблере.

```
MOV BX, 50H
MOV CL, 8
L1: MOV AX, [BX]
    ROL AX, CL
    MOV [BX], AX
    ADD BX, 2
    CMP BX, 57H
    JBE L1
```

Перед вами карта ячеек памяти от 50H до 57H, перед началом выполнения данного фрагмента программы.

адрес ячейки	50H	51H	52H	53H	54H	55H	56H	57H
содержимое ячейки	11H	22H	33H	44H	55H	66H	77H	88H

Примечание: обратите внимание, существует различие между переносом данных из ячеек памяти в 8-битные регистры и переносом данных из ячеек памяти в 16-битные регистры.

- (1) Проследите с помощью таблицы трассировки за выполнением этого фрагмента программы. Включите в таблицу трассировки столбец для каждого из регистров, представленных в этом фрагменте программы. Кроме того, начертите соответствующую карту памяти.
- (2) Объясните, что выполняет этот фрагмент программы.

Обратите внимание: продолжение вопроса на следующей странице.

/продолжение на странице 14/

- (ב) Перед вами фрагмент программы, написанный на языках Java и C#. Напишите соответствующий фрагмент на языке ассемблер.

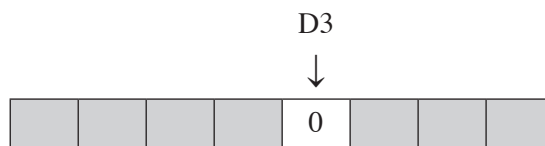
```
while ( a > 0 || b <= c )  
{  
    a = b + c ;  
    c -- ;  
}
```

Считайте, что переменные a , b , c являются знаковыми (signed) и хранятся в регистрах AX, BX, и CX соответственно.

- (א) Перед вами два утверждения о регистре AH. Для каждого из утверждений определите, верно оно или неверно. Если утверждение верно, обоснуйте ваш ответ. Если утверждение неверно, поместите в регистр AH число, доказывающее, что это утверждение не выполняется.
- i. Если в бите D3 (четвертый бит справа) регистра AH хранится значение 0, то обе приведенные ниже команды 1 и 2 выполняют одно и то же действие. Иными словами, значение регистра AH будет одинаковым после выполнения этих команд.

команда 2	команда 1
SUB AH, 8	AND AH, 11110111b

Воспользуйтесь приведенной ниже иллюстрацией:



- ii. Если в бите D3 (четвертый бит справа) регистра AH хранится значение 1, то обе приведенные ниже команды 1 и 2 выполняют одно и то же действие. Иными словами, значение регистра AH будет одинаковым после выполнения этих команд.

команда 2	команда 1
SUB AH, 8	AND AH, 11110111b

Воспользуйтесь приведенной ниже иллюстрацией:



/продолжение на странице 15/

8. В этом вопросе два пункта (א)-(ב). Между ними нет связи. Ответьте на вопросы обоих пунктов.

(א) Перед вами фрагмент программы на Ассемблере.

```
MOV AX, 620H
```

```
SUB AH, 76H
```

Определите, какими будут значения флагов ZF , CF , SF после выполнения этого фрагмента программы. Обоснуйте ваш ответ.

(ב) Во фрагменте данных определены следующие данные:

```
ARR DB 10 DUP (?)
```

Напишите процедуру под названием TEST , которая получит через стек два целых числа. Первое число, передаваемое через стек – это значение элемента из массива ARR . Второе число, передаваемое через стек – индекс этого элемента массива.

Если от начала массива до переданного индекса (не включительно) есть хотя бы один элемент, значение которого равно переданному значению элемента, то процедура запишет число 1 в регистр AL , иначе – процедура запишет число 0 в регистр AL .

Пример: в массив ARR переданы число 5 и индекс 4.

index	0	1	2	3	4	5	6	7	8	9
value	3	16	0	3	5	3	5	16	5	0

В этом примере процедура запишет в регистр AL число 0 , так как от начала массива и до индекса 4 значение 5 не встречается.

Дополнительный пример: в массив ARR переданы число 5 и индекс 8.

index	0	1	2	3	4	5	6	7	8	9
value	3	16	0	3	5	3	5	16	5	0

В этом примере процедура запишет в регистр AL число 1 , так как от начала массива и до индекса 8 есть элементы со значением 5 .

Введение в исследование операций [מבוא לחקר ביצועים]

Если вы изучали этот курс, ответьте на один из вопросов 9-10 (25 баллов).

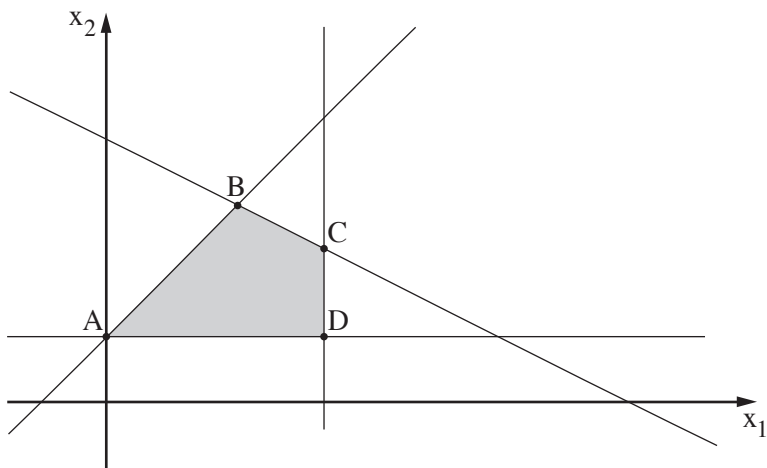
9. Дана задача линейного программирования:

$$\max \{z = ax_1 + 10x_2\}$$

при следующих ограничениях:

- (1) $x_1 + 3 \geq x_2$
- (2) $x_2 \geq 3$
- (3) $-0.5x_1 + 12 \geq x_2$
- (4) $x_1 \leq 10$

Перед вами чертеж области возможных решений данной задачи.



Обратите внимание: продолжение вопроса на следующей странице.

- (א) Найдите координаты точек A , B , C , D .
- (ב) На всей длине отрезка BC значение целевой функции одинаково. Вычислите a .
- (ג) Перед вами четыре высказывания:
- i. Есть только одно оптимальное решение.
 - ii. Есть бесконечное множество оптимальных решений.
 - iii. Оптимальное решение не ограничено.
 - iv. Нет оптимального решения.

Определите, какое из высказываний i-iv верно. Перепишите верное высказывание в вашу тетрадь и обоснуйте свой ответ.

- (ד) Отменяют второе ограничение: $x_2 \geq 3$.
- (1) Начертите новую область возможных решений.
- (2) Изменилось ли оптимальное решение после отмены этого ограничения?
Обоснуйте свой ответ.
- (3) Изменяют также целевую функцию на функцию минимума $\min \{z = ax_1 + 10x_2\}$.
Изменилось ли оптимальное решение после этого изменения? Обоснуйте свой ответ.

10. В этом вопросе два пункта, (א)-(ב). Между ними нет связи. Ответьте на вопросы обоих пунктов.

(א) (1) В таблице ниже дано решение транспортной задачи [בעיית תובלה]. Является ли это решение возможным базовым решением по методу северо-западного угла? Обоснуйте свой ответ.

Пункты отправле- ния	Пункты назначения				Предло- жение
	A	B	C	D	
1	5 100	3	8 100	7 100	300
2	9	5 300	10 100	12 100	500
3	8	6	7	3 200	200
Спрос	100	300	200	400	

Обратите внимание: продолжение вопроса на следующей странице.

- (2) В таблице ниже дано возможное базовое решение, полученное после k итераций для транспортной задачи, и дано значение v_2 :

Пункты отправления	Пункты назначения			Предложение	u_i
	1	2	3		
A	10	3 100	8	100	
B	9 100	7 100	6	200	
C	18 50	9	4 350	400	
D	11	6 300	8	300	
Спрос	150	500	350		
v_j		0			

- Перенесите эту таблицу в тетрадь и дополните ее значениями u_1 , u_2 , u_3 , u_4 , v_1 , v_3 .
- Объясните, почему данное решение не является оптимальным.
- Вы должны произвести дополнительную итерацию, то есть итерацию $k + 1$. Какая переменная выходит из базы в этой итерации?
- Начертите в тетради новую таблицу и запишите в ней решение, полученное после этой итерации.
- Уменьшилась ли общая стоимость этой транспортной задачи?

Обратите внимание: пункт (2) этого вопроса на следующей странице.

- (а) Перед вами 6 городов a, b, c, d, e, f , соединенных сетью дорог. Общая стоимость проезда по дороге состоит из двух элементов: стоимость топлива и тариф за пользование дорогой.

Перед вами матрица смежностей, представляющая стоимость топлива в шекелях для проезда между любыми двумя городами.

	a	b	c	d	e	f
a	0	6	3	∞	∞	∞
b	6	0	7	2	11	∞
c	3	7	0	10	2	2
d	∞	2	10	0	8	2
e	∞	11	2	8	0	6
f	∞	∞	2	2	6	0

Перед вами матрица смежностей, представляющая тариф в шекелях за пользование дорогами для проезда между любыми двумя городами.

	a	b	c	d	e	f
a	0	4	20	∞	∞	∞
b	4	0	2	9	1	∞
c	20	2	0	10	13	5
d	∞	9	10	0	3	13
e	∞	1	13	3	0	3
f	∞	∞	5	13	3	0

Найдите с помощью алгоритма Дейкстры самые дешевые маршруты для проезда из города a в каждый из городов b, c, d, e, f . Запишите минимальную общую стоимость проезда из города a в каждый из других городов.

Вычислительные модели [מודלים חישוביים]

Если вы изучали этот курс, ответьте на один из вопросов 11-12 (25 баллов).

- 11.** Постройте машину Тьюринга, получающую в качестве ввода числа x , y , z с начала ленты (числа не отрицательные). x – это первое число (с начала ленты), y – второе, а z – третье.

Каждое из чисел записано в унарной форме.

Числа отделены друг от друга символом $\#$.

Пример: если введено 2 для x , 3 для y и 1 для z , то лента будет выглядеть так:

┌	1	1	#	1	1	1	#	1	△	△	△	
---	---	---	---	---	---	---	---	---	---	---	---	------

Дополнительный пример: если введено 0 для x , 1 для y и 0 для z , то лента будет выглядеть так:

┌	#	1	#	△	△	△	
---	---	---	---	---	---	---	------

Машина вычислит значение приведенной ниже функции:

$$f(x, y, z) = \begin{cases} y + z & x = 0 \\ x + y & x - \text{четное число, а также } x > 0 \\ x & x - \text{нечетное число} \end{cases}$$

Вывод будет записан на ленте в некотором месте в виде унарного значения между двумя символами $\$$.

- 12. (а)** Постройте стек-автомат для языка L_1 над алфавитом $\{a, b, c\}$, состоящего из сочетаний вида $a^n b^k c^n$ таких, что n – нечетное число, а остаток от деления k на 3 равен единице. Между каждой парой сочетаний расположена буква b .

Примеры слов, принадлежащих языку L_1 :

abbbbcbbaabccs, abc

Примеры слов, не принадлежащих языку L_1 :

abbc – так как буква b появляется 2 раза, а остаток от деления 2 на 3 равен 2.

abcabc – так как буква b не расположена между двумя сочетаниями.

abccc – так как буква a появляется не столько же раз, сколько буква c .

aabcc – так как буква a и буква c появляются четное число раз.

- (б)** Дан язык L_2 над алфавитом $\{a, b, c\}$.

$$L_2 = \{a^k b^m c^x \mid 0 \leq k < 5, 0 \leq m < 5, 0 \leq x\}$$

Определим $L_3 = L_2 \cap L_1$.

Запишите все слова языка L_3 .

/продолжение на странице 22/

Объектно-ориентированное программирование [תכנות מונחה עצמים] на языке Java

Если вы изучали этот курс и пишете на языке Java, ответьте на один из вопросов 13-14 (25 баллов).

13. Даны классы: **A**, **B**, **C**, **D**, **E**.

```
public class A
{
    protected int x ;
    public A ()
    {
        this.x = 9 ;
        System.out.println ( "A. x = " +this.x ) ;
    }
    public A ( int x )
    {
        this.x = x ;
        System.out.println ( "A. x = " + this.x ) ;
    }
    public int getX () { return this.x ; }
    public int foo () { return this.x ; }
}

public class B extends A
{
    public B () { super () ; }
    public B ( int x ) { super ( x ) ; }
    public int foo () { return this.x +1 ; }
}

public class C extends B
{
    public C () { super () ; }
    public C ( int x ) { super ( x ) ; }
    public int foo () { return this.x + 2 ; }
    public int bar () { return this.x ; }
}
```

```
public class D extends C
{
    public D ()
    {
        super () ;
        this.x ++ ;
        System.out.println ( "D. x = " + this.x ) ;
    }
    public D ( int x )
    {
        super ( x ) ;
        System.out.println ( "D. x = " + this.x ) ;
    }
    public D ( int x, int y )
    {
        super ();
        this.x = this.x + x + y;
        System.out.println ( "D. x = " + this.x ) ;
    }
    public int foo () { return this.x - 1 ; }
}

public class E extends C
{
    public E () { super () ; }
    public int bar () { return this.x +1 ; }
}
```

Обратите внимание: продолжение вопроса на следующей странице.

(⌘) Начертите схему иерархии между классами A , B , C , D , E . Наследование от класса обозначьте при помощи стрелки $\longrightarrow$.

(⌘) Перед вами заголовок метода:

```
public static int getType ( Object m )
```

Метод вернет 1 , если m – это объект типа A ; 2 , если m – это объект типа B ;

3 , если m – это объект типа C ; 4 , если m – это объект типа D ;

и 5 , если m – это объект типа E .

Реализуйте этот метод.

Пояснение: чтобы выяснить тип объекта, используйте методы foo , bar , getX .

Нельзя использовать метод instanceof и методы класса Object , а также нельзя изменять классы A , B , C , D , E .

Считайте, что объект m принадлежит к одному из классов A , B , C , D , E и не равен null .

(⌘) Перед вами класс **Tester** :

```
public class Tester

{
    public static void main ( String [] args )

    {
        A a1 = new B () ;
        A a2 = new E () ;
        A a3 = new D () ;
        A a4 = new D ( 5 ) ;
        A a5 = new D ( 3, 7 ) ;
    }
}
```

Запишите вывод метода main.

14. В фирме сотовой связи "Слышим будущее" есть 4 типа работников. Каждый работник принадлежит только к одному типу.

Обычный работник – обычный работник фирмы.

Член профсоюза – работник, являющийся членом профессионального союза.

Техник – работник, ответственный за несколько принадлежащих фирме компьютеров (каждый техник отвечает за разное число компьютеров).

Куратор проекта – работник, руководящий особыми проектами в фирме. Такой работник отвечает за 15 работников, среди которых обычные работники, члены профсоюза и техники.

Решения в фирме принимаются посредством голосования при участии всех работников фирмы. Подсчет голосов происходит при помощи баллов. Каждый работник фирмы получает разное число баллов в зависимости от типа работника.

Перед вами таблица с данными о числе баллов, получаемых работниками:

Тип работника	Число баллов
Обычный работник	Получает 4 балла как работник фирмы, а также дополнительный балл за каждый год работы в фирме. <u>Пример:</u> обычный работник, проработавший в фирме 5 лет, получает 9 баллов: 4 балла как работник фирмы и еще 5 баллов – по баллу за каждый год работы.
Член профсоюза	Получает в 2 раза больше баллов, чем обычный работник, а также 2 дополнительных балла за каждый год его членства в профсоюзе. <u>Пример:</u> член профсоюза, проработавший в фирме 5 лет, из которых 3 года он является членом профсоюза, получает 24 балла: 18 баллов – вдвое больше, чем обычный работник, проработавший в фирме 5 лет, и еще 6 баллов – по 2 балла за каждый год его членства в профсоюзе.
Техник	Получает такое же количество баллов, как обычный работник, а также один дополнительный балл за каждый компьютер, за который он отвечает. <u>Пример:</u> техник, проработавший в фирме 5 лет и ответственный за 4 компьютера, получает 13 баллов: 9 баллов – как обычный работник, проработавший в фирме 5 лет, и еще 4 балла – по одному баллу за каждый компьютер, за который он отвечает.
Куратор проекта	Получает такое же количество баллов, как обычный работник, а также дополнительное количество баллов, равное сумме баллов всех 15 работников в его подчинении. <u>Пример:</u> куратор проекта, проработавший в фирме 5 лет и сумма баллов 15-ти подчиненных которого равна 80, получит 89 баллов: 9 баллов – как обычный работник, проработавший в фирме 5 лет, и еще 80 баллов – сумма баллов 15 работников, за которых он отвечает.

Обратите внимание: продолжение вопроса на следующей странице.

В фирме планируют создание компьютерной системы для подсчета баллов каждого работника. Для этого определили 4 класса:

Employee – обычный работник, **UnionMember** – член профсоюза, **Technician** – техник, **Supervisor** – куратор проекта.

Каждый работник принадлежит только одному классу.

Приведенные ниже пункты необходимо решить в максимальном соответствии с принципами объектно-ориентированного программирования.

(а) Начертите схему иерархии классов.

Наследование [ירושה] изобразите стрелкой ,

а включение [הכללה] изобразите символом .

(ב) Для каждого из классов напишите заголовок класса, его атрибуты и метод `getScore ()`, возвращающий число баллов при голосовании. Нет необходимости определять конструктор класса и методы `get` и `set`.

Указание: постоянные данные (константы) количества баллов не являются атрибутами классов [תכונות המחלקות], и их необходимо использовать только при написании метода `getScore`.

(ג) Возникло предложение изменить название фирмы на "Звонок в будущее". Все обычные работники и техники поддержали предложение, а все члены профсоюза и кураторы проектов были против. Предложение будет принято только если сумма баллов за предложение больше, чем сумма баллов против него. Иначе предложение принято не будет.

Дан заголовок метода, получающего массив всех работников и возвращающего `true`, если предложение будет принято, а иначе – возвращающего `false`.

```
public static boolean isAccepted (Object [] arr)
```

Реализуйте этот метод.

Объектно-ориентированное программирование [תכנות מונחה עצמים] на языке C#

Если вы изучали этот курс и пишете на языке C#, ответьте на один из вопросов 15-16 (25 баллов).

15. Даны классы: **A**, **B**, **C**, **D**, **E**.

```
public class A
{
    protected int x ;
    public A ()
    {
        this.x = 9 ;
        Console.WriteLine ( "A. x = " +this.x ) ;
    }
    public A ( int x )
    {
        this.x = x ;
        Console.WriteLine ( "A. x = " + this.x ) ;
    }
    public int GetX () { return this.x ; }
    public virtual int Foo () { return this.x ; }
}

public class B : A
{
    public B () : base () {}
    public B ( int x ) : base ( x ) {}
    public override int Foo () { return this.x +1 ; }
}

public class C : B
{
    public C () : base () {}
    public C ( int x ) : base ( x ) {}
    public override int Foo () { return this.x +2 ; }
    public virtual int Bar () { return this.x ; }
}
```

```
public class D : C
{
    public D () : base ()
    {
        this.x ++ ;
        Console.WriteLine ( "D. x = " + this.x ) ;
    }
    public D ( int x ) : base ( x )
    {
        Console.WriteLine ( "D. x = " + this.x ) ;
    }
    public D ( int x, int y ) : base ()
    {
        this.x = this.x + x + y ;
        Console.WriteLine ( "D. x = " + this.x ) ;
    }
    public override int Foo () { return this.x - 1 ; }
}

public class E : C
{
    public E () : base () {}
    public override int Bar () { return this.x +1 ; }
}
```

Обратите внимание: продолжение вопроса на следующей странице.

(⌘) Начертите схему иерархии между классами A , B , C , D , E . Наследование от класса обозначьте при помощи стрелки $\longrightarrow$.

(⌘) Перед вами заголовок метода:

```
public static int GetType ( Object m )
```

Метод вернет 1 , если m – это объект типа A , 2 если m – это объект типа B , 3 если m – это объект типа C , 4 если m – это объект типа D , и 5 если m – это объект типа E .

Реализуйте данный метод.

Пояснение: чтобы выяснить тип объекта, используйте методы Foo , Bar , GetX .

Нельзя использовать метод is и as и методы класса Object , а также нельзя изменять классы A , B , C , D , E .

Считайте, что объект m принадлежит к одному из классов A , B , C , D , E и не равен null .

(⌘) Перед вами класс **Tester** :

```
public class Tester

{
    public static void Main ( string [] args )

    {
        A a1 = new B () ;
        A a2 = new E () ;
        A a3 = new D () ;
        A a4 = new D ( 5 ) ;
        A a5 = new D ( 3, 7 ) ;
    }
}
```

Запишите вывод метода Main.

16. В фирме сотовой связи "Слышим будущее" есть 4 типа работников. Каждый работник принадлежит только к одному типу.

Обычный работник – обычный работник фирмы.

Член профсоюза – работник, являющийся членом профессионального союза.

Техник – работник, ответственный за несколько принадлежащих фирме компьютеров (каждый техник отвечает за разное число компьютеров).

Куратор проекта – работник, руководящий особыми проектами в фирме. Такой работник отвечает за 15 работников, среди которых обычные работники, члены профсоюза и техники.

Решения в фирме принимаются посредством голосования при участии всех работников фирмы. Подсчет голосов происходит при помощи баллов. Каждый работник фирмы получает разное число баллов, в зависимости от типа работника.

Перед вами таблица с данными о числе баллов, получаемых работниками:

Тип работника	Число баллов
Обычный работник	Получает 4 балла как работник фирмы, а также дополнительный балл за каждый год работы в фирме. <u>Пример:</u> обычный работник, проработавший в фирме 5 лет, получает 9 баллов: 4 балла как работник фирмы и еще 5 баллов – по баллу за каждый год работы.
Член профсоюза	Получает в 2 раза больше баллов, чем обычный работник, а также 2 дополнительных балла за каждый год его членства в профсоюзе. <u>Пример:</u> член профсоюза, проработавший в фирме 5 лет, из которых 3 года он является членом профсоюза, получает 24 балла: 18 баллов – вдвое больше, чем обычный работник, проработавший в фирме 5 лет, и еще 6 баллов – по 2 балла за каждый год его членства в профсоюзе.
Техник	Получает такое же количество баллов, как обычный работник, а также один дополнительный балл за каждый компьютер, за который он отвечает. <u>Пример:</u> техник, проработавший в фирме 5 лет и ответственный за 4 компьютера, получает 13 баллов: 9 баллов – как обычный работник, проработавший в фирме 5 лет, и еще 4 балла – по одному баллу за каждый компьютер, за который он отвечает.
Куратор проекта	Получает такое же количество баллов, как обычный работник, а также дополнительное количество баллов, равное сумме баллов всех 15 работников в его подчинении. <u>Пример:</u> куратор проекта, проработавший в фирме 5 лет и сумма баллов 15-ти подчиненных которого равна 80, получит 89 баллов: 9 баллов – как обычный работник, проработавший в фирме 5 лет, и еще 80 баллов – сумма баллов 15 работников, за которых он отвечает.

Обратите внимание: продолжение вопроса на следующей странице.

/продолжение на странице 29/

В фирме планируют создание компьютерной системы для подсчета баллов каждого работника. Для этого определили 4 класса:

Employee – обычный работник, **UnionMember** – член профсоюза, **Technician** – техник, **Supervisor** – куратор проекта.

Каждый работник принадлежит только одному классу.

Приведенные ниже пункты необходимо решить в максимальном соответствии с принципами объектно-ориентированного программирования.

(א) Начертите схему иерархии классов.

Наследование [ירושה] изобразите стрелкой  ,

а включение [הכלה] изобразите символом  .

(ב) Для каждого из классов напишите заголовок класса, его атрибуты и метод GetScore () , возвращающий число баллов при голосовании. Нет необходимости определять конструктор класса и методы Get и Set .

Указание: постоянные данные (константы) количества баллов не являются атрибутами классов [תכונות המחלקות], и их необходимо использовать только при написании метода GetScore .

(ג) Возникло предложение изменить название фирмы на "Звонок в будущее". Все обычные работники и техники поддержали предложение, а все члены профсоюза и кураторы проектов были против. Предложение будет принято только если сумма баллов за предложение больше, чем сумма баллов против него. Иначе предложение принято не будет.

Дан заголовок метода, получающего массив всех работников и возвращающего true , если предложение будет принято, а иначе – возвращающего false .

```
public static bool IsAccepted (Object [] arr)
```

Реализуйте этот метод.

Желаем успеха!

Авторские права принадлежат Государству Израиль.
Копировать или публиковать можно только
с разрешения Министерства просвещения.

בהצלחה!

זכות היוצרים שמורה למדינת ישראל.
אין להעתיק או לפרסם
אלא ברשות משרד החינוך.