Государство Израиль
Министерство просвещения
Тип экзамена: на аттестат зрелости
Время проведения экзамена: лето 2021 года
Номер вопросника: 899381
Перевод на русский язык (5)

מועד קיץ

מדינת ישראל
משרד החינוך
סוג הבחינה: בגרות
מועד הבחינה: קיץ תשפ"א, 2021
מספר השאלון: 899381
תרגום לרוסית (5)

Обратите внимание: в этом вопроснике есть специальные инструкции.
Отвечайте на вопросы, следуя этим инструкциям.

שים לב: בבחינה זו יש הנחיות מיוחדות.
יש לענות על השאלות על פי הנחיות אלה.

Компьютеры

מדעי המחשב

Указания экзаменующимся

הוראות לנבחן

а. Продолжительность экзамена: три часа.

א. משך הבחינה: שלוש שעות.

б. Строение вопросника и ключ к оценке:
в этом вопроснике три раздела.
Вы должны выбрать пять вопросов из этих трех разделов.
За каждый вопрос – 20 баллов (5 × 20).
Всего – 100 баллов.
Обратите внимание: если вы отвечаете на вопросы третьего раздела, выбирайте вопросы только одного курса.

ב. מבנה השאלון ומפתח ההערכה:
בשאלון זה שלושה פרקים.
יש לבחור בחמש שאלות מתוך שלושת הפרקים.
לכל שאלה — 20 נקודות (5×20).
סך הכול — 100 נקודות.
שים לב: אם תבחר לענות על שאלות מן הפרק השלישי, בחר בשאלות מתוך מסלול אחד בלבד.

в. Разрешенный вспомогательный материал:
любой вспомогательный материал, за исключением калькулятора с возможностью программирования.

ג. חומר עזר מותר בשימוש:
כל חומר עזר, חוץ ממחשבון שיש בו אפשרות תכנות.

г. Особые указания:

(1) **На внешней обложке** тетради **укажите** название **курса**, который **вы изучали,** если вы отвечаете на вопросы третьего раздела, а если нет – напишите **"ללא מסלול"**.
Курс – это один из из четырех следующих курсов: "Компьютерные системы и ассемблер", "Введение в исследование операций", "Вычислительные модели", "Объектно-ориентированное программирование".

ד. הוראות מיוחדות:

(1) **רשום על הכריכה החיצונית** של המחברת את **שם המסלול שלמדת**, אם תבחר לענות על שאלות מן הפרק השלישי, ואם לא — רשום **"ללא מסלול"**.
המסלול הוא אחד מארבעת המסלולים האלה: מערכות מחשב ואסמבלי, מבוא לחקר ביצועים, מודלים חישוביים, תכנות מונחה עצמים.

(2) Все программы, которые вы должны написать на языке программирования в первом и втором разделах, пишите только на одном языке – Java или C# .

(2) את כל התוכניות שאתה נדרש לכתוב בשפת מחשב בפרקים הראשון והשני כתוב בשפה אחת בלבד — Java או C# .

Примечание: баллы сниматься не будут, если в написанных вами программах вы напишете большую букву вместо маленькой или наоборот.

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

כתוב במחברת הבחינה בלבד. רשום "טיוטה" בראש כל עמוד המשמש טיוטה.
כתיבת טיוטה בדפים שאינם במחברת הבחינה עלולה לגרום לפסילת הבחינה.
Пишите только в экзаменационной тетради. Напишите слово «טיוטה» 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

/продолжение на странице 3/

В о п р о с ы

В этом вопроснике три раздела.
Вы должны выбрать пять вопросов из трех этих разделов. За каждый вопрос – 20 баллов.
На внешней обложке тетради укажите название курса, который вы изучали,
если вы отвечаете на вопросы третьего раздела, иначе – напишите "ללא מסלול".

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

1. Напишите внешний метод [פעולה חיצונית] filter на языке Java или Filter на языке C#, получающий массив arr целого типа и число num целого типа. Метод вернет новый массив целого типа, содержащий только те числа из массива arr , которые не равны num .

Пример: для числа num , равного 9, и массива arr длиной 7 , приведенных ниже:

	0	1	2	3	4	5	6
arr	6	**9**	2	2	**9**	4	– 3

метод вернет массив длиной 5, который выглядит так:

0	1	2	3	4
6	2	2	4	– 3

Считайте, что в массиве arr существует по меньшей мере одно число, которое не равно num , и по меньшей мере одно число, которое равно num .

Обратите внимание: длина возвращаемого массива – это число элементов в массиве arr , которые не равны num .

Примечание: порядок чисел в возвращаемом массиве не имеет значения.

/продолжение на странице 4/

2. Обратите внимание: этот вопрос сформулирован дважды: на языке Java на страницах 4-5 и на языке C# на страницах 6-7.

Для решающих на языке Java

Дан класс [מחלקה] **Subject** , представляющий предмет в аттестате ученика, и в нем два атрибута [תכונות]: название предмета – subName и оценка – grade (от 0 до 100).

```
public class Subject {
        private String subName;
        private int grade;
}
```

Также дан класс **ReportCard** , представляющий аттестат ученика, и в нем два атрибута: имя ученика – stuName и массив объектов – subArray типа **Subject**, длина которого равна числу предметов, которые изучал ученик.

```
public class ReportCard {
        private String stuName;
        private Subject [] subArray;
 }
```

Считайте, что для всех атрибутов в двух данных классах определены методы get и set .
Перед вами частичный интерфейс класса **ReportCard** :

Заголовок метода	Описание метода
public ReportCard (String name, int num)	Конструктор [פעולה בונה], получающий имя ученика и число предметов, которые он изучал, и инициализирующий атрибуты класса. Массив subArray инициализируется как пустой массив длиной num .
public double average ()	Метод, возвращающий среднее арифметическое [ממוצע] всех оценок в аттестате ученика.
public boolean isExcellent ()	Метод, возвращающий true , если ученик считается отличником. Иначе метод возвращает false. Отличник: ученик, среднее арифметическое оценок которого 85 или больше, все его оценки больше, чем 54, и по меньшей мере одна из его оценок равна 100.

Обратите внимание: продолжение вопроса на следующей странице.

/продолжение на странице 5/

(א) Реализуйте конструктор [פעולה בונה] класса **ReportCard** .

(ב) Реализуйте метод isExcellent в классе **ReportCard**.

Можно пользоваться методом average класса **ReportCard** , не реализуя его.

Считайте, что элементы массива subArray не равны null.

(ג) Напишите внешний метод [פעולה חיצונית] printExcellent , принимающий массив объектов – array , типа **ReportCard**, представляющий учеников, учившихся в определенном классе.

Этот метод выведет на экран имена отличников.

Вы должны использовать метод isExcellent класса **ReportCard.**

Считайте, что элементы массива array не равны null.

/продолжение на странице 6/

Для решающих на языке C#

Дан класс **Subject** , представляющий предмет в аттестате ученика, и в нем два атрибута [תכונות]: название предмета – subName и оценка – grade (от 0 до 100).

```
public class Subject {
        private string subName;
        private int grade;
}
```

Также дан класс **ReportCard** , представляющий аттестат ученика, и в нем два атрибута: имя ученика – stuName и массив объектов – subArray типа **Subject**, длина которого равна числу предметов, которые изучал ученик.

```
public class ReportCard {
        private string stuName;
        private Subject [] subArray;
 }
```

Считайте, что для всех атрибутов в двух данных классах определены методы Get и Set .
Перед вами частичный интерфейс класса **ReportCard** :

Заголовок метода	Описание метода
public ReportCard (string name, int num)	Конструктор [פעולה בונה], получающий имя ученика и число предметов, которые он изучал, и инициализирующий атрибуты класса. Массив subArray инициализируется как пустой массив длиной num .
public double Average ()	Метод, возвращающий среднее арифметическое [ממוצע] всех оценок в аттестате ученика.
public bool IsExcellent ()	Метод, возвращающий true , если ученик считается отличником. Иначе метод возвращает false. Отличник: ученик, среднее арифметическое оценок которого 85 или больше, все его оценки больше, чем 54, и по меньшей мере одна из его оценок равна 100.

/продолжение на странице 7/

(א) Реализуйте конструктор [פעולה בונה] класса **ReportCard** .

(ב) Реализуйте метод IsExcellent в классе **ReportCard**.

Можно пользоваться методом Average класса **ReportCard** , не реализуя его.

Считайте, что элементы массива subArray не равны null.

(ג) Напишите внешний метод [פעולה חיצונית] PrintExcellent , принимающий массив объектов – array , типа **ReportCard**, представляющий учеников, учившихся в определенном классе.

Этот метод выведет на экран имена отличников.

Вы должны использовать метод IsExcellent класса **ReportCard.**

Считайте, что элементы массива array не равны null.

/продолжение на странице 8/

3. Обратите внимание: этот вопрос сформулирован дважды: на языке Java на страницах 8-9 и на языке C# на страницах 10-11.

Для решающих на языке Java

"Особая строка" – это строка, в которой одинаковые символы появляются подряд, или строка, в которой нет одинаковых символов (в том числе пустая строка).

Примеры "особой строки": "ccabbb" , "aaaa" , "a" , "*&$" , "**1zz" .

Примеры строк, которые не являются "особой строкой": "aba" , "aabcb" , "**33%3" .

Дан класс **MyString**.

```
public class MyString {
    private String str;
}
```

Перед вами интерфейс класса **MyString**.

Вы должны пользоваться методами этого интерфейса, нет необходимости реализовывать их.

Заголовок метода	Описание метода
public MyString ()	Конструктор [פעולה בונה], инициализирующий пустую строку.
public int countChar (char ch)	Метод, возвращающий число появлений символа ch в строке str .
public void removeChar (char ch)	Метод, получающий символ ch и убирающий из строки str все его появления.
public void appendChar (char ch)	Метод, добавляющий в конец строки str символ ch .
public char firstChar ()	Метод, возвращающий первый символ в строке str .
public boolean isEmpty ()	Метод, возвращающий true , если строка str пуста, иначе возвращающий false .

Обратите внимание: продолжение вопроса на следующей странице.

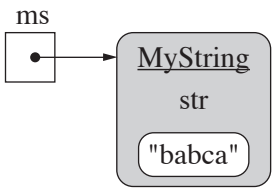
/продолжение на странице 9/

Перед вами заголовок внешнего метода [פעולה חיצונית] special:

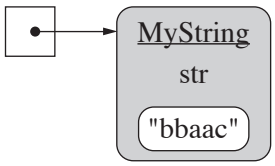
```
public static MyString special (MyString ms)
```

Метод special возвращает новый объект типа **MyString** , атрибут которого, str , является "особой строкой", состоящей из всех символов атрибута str объекта ms .
Реализуйте метод special только посредством методов интерфейса класса **MyString** .
Запрещается добавлять методы, в том числе запрещается добавлять методы get и set, к классу **MyString** .

Например:
для приведенного ниже объекта ms , в котором обычная строка:



метод special вернет новый объект, в котором "особая строка":



Указания:
– нет необходимости сохранять неизменной строку str объекта ms .
– порядок символов в новом объекте не имеет значения, если соблюдается определение "особой строки".

/продолжение на странице 10/

Для решающих на языке C#

"Особая строка" – это строка, в которой одинаковые символы появляются подряд, или строка, в которой нет одинаковых символов (в том числе пустая строка).

Примеры "особой строки": "ccabbb" , "aaaa" , "a" , "*&$" , "**1zz" .

Примеры строк, которые не являются "особой строкой": "aba" , "aabcb" , "**33%3" .

Дан класс **MyString**.

```
public class MyString {
    private string str;
}
```

Перед вами интерфейс класса **MyString**.
Вы должны пользоваться методами этого интерфейса, нет необходимости реализовывать их.

Заголовок метода	**Описание метода**
public MyString ()	Конструктор [פעולה בונה], инициализирующий пустую строку.
public int CountChar (char ch)	Метод, возвращающий число появлений символа ch в строке str .
public void RemoveChar (char ch)	Метод, получающий символ ch и убирающий из строки str все его появления.
public void AppendChar (char ch)	Метод, добавляющий в конец строки str символ ch .
public char FirstChar ()	Метод, возвращающий первый символ в строке str .
public boolean IsEmpty ()	Метод, возвращающий true , если строка str пуста, иначе возвращающий false .

Обратите внимание: продолжение вопроса на следующей странице.

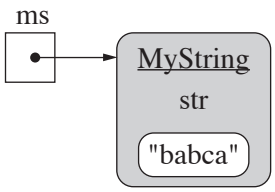
/продолжение на странице 11/

Перед вами заголовок внешнего метода [פעולה חיצונית] Special:

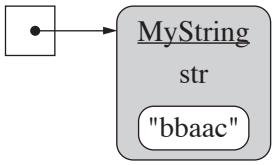
```
public static MyString Special (MyString ms)
```

Метод Special возвращает новый объект типа **MyString** , атрибут которого, str , является "особой строкой", состоящей из всех символов атрибута str объекта ms .
Реализуйте метод Special только посредством методов интерфейса класса **MyString** .
Запрещается добавлять методы, в том числе запрещается добавлять методы Get и Set, к классу **MyString** .

Например:
для приведенного ниже объекта ms , в котором обычная строка:



метод Special вернет новый объект, в котором "особая строка":



Указания:
– нет необходимости сохранять неизменной строку str объекта ms .
– порядок символов в новом объекте не имеет значения, если соблюдается определение "особой строки".

/продолжение на странице 12/

РАЗДЕЛ ВТОРОЙ

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 [לממש].

4. В этом вопросе можно использовать приведенный ниже внешний метод [פעולה חיצונית], не реализуя его.

Заголовок метода	Описание метода
На языке **Java** public static Node<Integer> delete (int num, Node<Integer> lst) На языке **C#** public static Node<int> Delete (int num, Node<int> lst)	Метод получает число — num и указатель [הפנָיה] на начало цепочки звеньев [שרשרת חוליות] — lst . Метод удаляет звенья со значением num и возвращает указатель на начало цепочки звеньев.

Дан класс **BiList** – двойная цепочка с двумя атрибутами:

- lst1 – указатель на начало цепочки звеньев целого типа
- lst2 – указатель на начало цепочки звеньев целого типа

Перед вами частичный интерфейс класса **BiList** на языках **Java** и **C#** .
Вы должны пользоваться методами этого интерфейса, нет необходимости реализовывать их.

Заголовок метода	Описание метода
public BiList ()	Конструктор [פעולה הבונה] объекта с указателями на две пустые цепочки.
На языке **Java** public void addNum (int num, int codeList) На языке **C#** public void AddNum (int num, int codeList)	Метод добавляет звено со значением num в конец цепочки lst1 или в конец цепочки lst2 в соответствии с codeList : Когда codeList = 1 , num добавится в lst1 , когда codeList = 2 , num добавится в lst2 . Считайте, что значение параметра codeList корректно.

Обратите внимание: продолжение вопроса на следующей странице.

/продолжение на странице 13/

Напишите внешний метод [פעולה חיצונית] с заголовком generateBilist на языке Java или GenerateBilist на языке C#, принимающий цепочку звеньев целых чисел – lst. Количество звеньев в lst четное [זוגי], и числа в ее звеньях отличаются друг от друга. Метод вернет объект типа **Bilist** , для которого выполняются следующие условия:

– каждое из чисел из цепочки lst появится в одной из цепочек lst1 и lst2 .

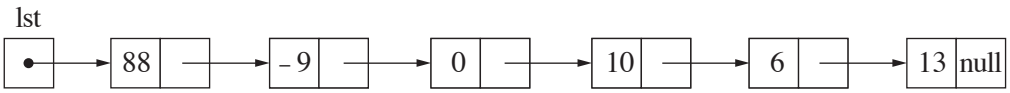
– все числа в цепочке lst1 будут больше всех чисел в цепочке lst2 .

– количество звеньев в обеих цепочках lst1 и lst2 будет одинаковым.

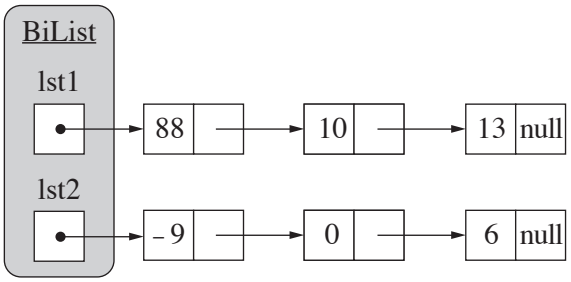
Обратите внимание: запрещается добавлять методы в класс **Bilist** , в том числе запрещается добавлять методы get и set на языке Java и методы Get и Set на языке C# .

Пример:

для приведенной ниже цепочки lst :



метод вернет этот объект:



Примечание:

– нет необходимости сохранять неизменной цепочку lst .

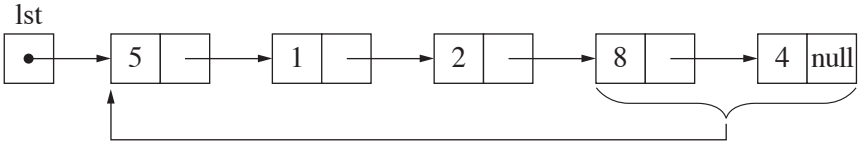
– порядок элементов в цепочке lst1 и в цепочке lst2 не имеет значения.

/продолжение на странице 14/

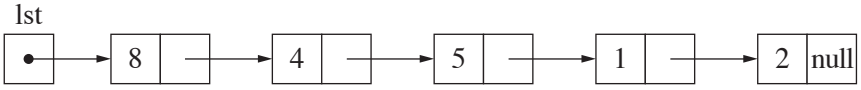
5. В цепочке звеньев [שרשרת חוליות] "круговой перенос n звеньев" – это перенос последних n звеньев в начало цепочки (без изменения порядка их появления).

Перед вами пример для n = 2 : переносим два последних звена в начало цепочки.

Цепочка звеньев до переноса



Цепочка звеньев после переноса



(א) Напишите внешний метод [פעולה חיצונית] с заголовком move на языке Java или Move на языке C# , принимающий цепочку звеньев – lst целого типа и число n целого типа. Метод вернет цепочку звеньев после "кругового переноса n звеньев".

Считайте, что $n \geq 0$ и число звеньев в цепочке больше, чем n .

(ב) Какова временная сложность выполнения [סיבוכיות זמן הריצה] метода, который вы написали в пункте (א)? Обоснуйте свой ответ.

/продолжение на странице 15/

6. Обратите внимание: этот вопрос сформулирован дважды: на языке Java на странице 15 и на языке C# на странице 16.

На языке Java

(א) Перед вами метод sod1.

```
public static boolean sod1 (int[] arr, int x, int i)
{
 if ( i == − 1) return false;
 if (arr[i] == x) return true;
 return sod1(arr, x, i − 1);
}
```

(1) Запишите значение, возвращаемое вызовом метода sod1(a, 8, a.length − 1) для массива a, представленного ниже. Необходимо показать трассировку [מעקב].

	0	1	2	3	4
a	5	4	15	12	2

(2) Для некоторого массива a и числа x, какова цель метода sod1(a, x, a.length − 1) ?

(3) Какова временная сложность выполнения [סיבוכיות זמן הריצה] метода sod1? Обоснуйте свой ответ.

(ב) Перед вами метод sod2.

```
public static  boolean sod2 (int[] arr, int x, int i)
{
 if ( i == 0) return false;
 if (sod1(arr, x – arr[i], i − 1)) return true;
 return sod2(arr, x, i − 1);
}
```

(1) Запишите значение, возвращаемое вызовом метода sod2(a, 16, a.length − 1) для массива a, представленного ниже. Необходимо показать трассировку [מעקב].

	0	1	2	3	4
a	5	4	15	12	2

В этом пункте нет необходимости выполнять трассировку метода sod1.

(2) Для некоторого массива a и числа x, какова цель метода sod2(a, x, a.length − 1) ?

(3) Какова временная сложность выполнения [סיבוכיות זמן הריצה] метода sod2? Обоснуйте свой ответ.

/продолжение на странице 16/

На языке C#

(א) Перед вами метод Sod1.

```
public static bool Sod1 (int[] arr, int x, int i)
{
 if ( i == − 1) return false;
 if (arr[i] == x) return true;
 return Sod1(arr, x, i − 1);
}
```

(1) Запишите значение, возвращаемое вызовом метода Sod1(a, 8, a.Length − 1) для массива a , представленного ниже. Необходимо показать трассировку [מעקב].

	0	1	2	3	4
a	5	4	15	12	2

(2) Для некоторого массива a и числа x , какова цель метода Sod1(a, x, a.Length − 1) ?

(3) Какова временная сложность выполнения [סיבוכיות זמן הריצה] метода Sod1? Обоснуйте свой ответ.

(ב) Перед вами метод Sod2.

```
public static  boolean Sod2 (int[] arr, int x, int i)
{
 if ( i == 0) return false;
 if (Sod1(arr, x – arr[i], i − 1)) return true;
 return Sod2(arr, x, i − 1);
}
```

(1) Запишите значение, возвращаемое вызовом метода Sod2(a, 16, a.Length − 1) для массива a, представленного ниже. Необходимо показать трассировку [מעקב].

	0	1	2	3	4
a	5	4	15	12	2

В этом пункте нет необходимости выполнять трассировку метода Sod1.

(2) Для некоторого массива a и числа x , какова цель метода Sod2(a, x, a.Length − 1) ?

(3) Какова временная сложность выполнения [סיבוכיות זמן הריצה] метода Sod2? Обоснуйте свой ответ.

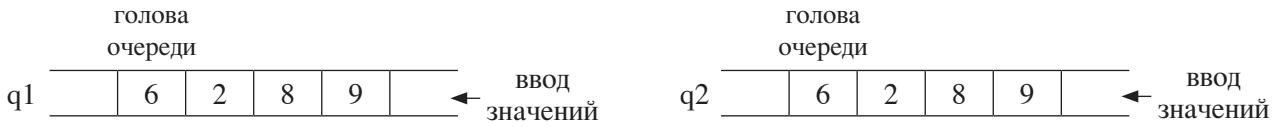
/продолжение на странице 17/

7. В этом вопросе вы можете пользоваться внешним методом [פעולה חיצונית], приведенным ниже, не реализуя его.

Заголовок метода	Описание метода
На языке **Java** — public static int size (Queue<Integer> q) На языке **C#** — public static int Size (Queue<int> q)	Метод возвращает число элементов в очереди q , не изменяя очереди.

(א) Две очереди, q1 и q2 , будут "одинаковыми очередями", если число элементов в обеих очередях одинаково и в обеих очередях присутствуют в точности одинаковые значения и в том же самом порядке.

Пример двух одинаковых очередей:



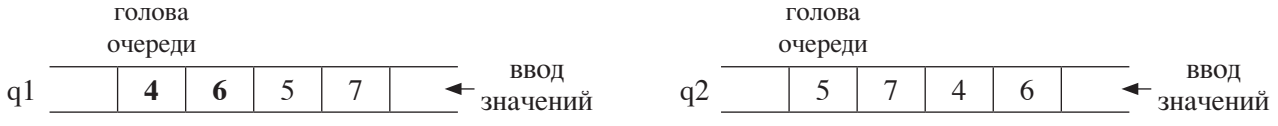
Напишите внешний метод с заголовком isIdentical на языке Java или IsIdentical на языке C#, который получает две очереди целого типа q1 и q2 и возвращает true, если эти очереди одинаковы, иначе – метод возвращает false.

Примечание: по окончании этого метода необходимо сохранить полученное исходное строение очередей.

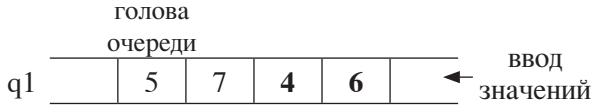
(ב) "Перенос из начала в конец" – это перенос числа из головы очереди в ее конец.
Пример:

Напишите внешний метод с заголовком isSimilar на языке Java или IsSimilar на языке C#, который получает две очереди целого типа q1 и q2 и возвращает true, если очереди q1 и q2 одинаковы или в своем исходном виде, или после того, как в очереди q1 выполняется "перенос из начала в конец", один раз или более. Иначе – метод возвращает false.

Пример: метод вернет true для двух очередей q1 и q2 , представленных ниже:



Причина этого в том, что после того, как в очереди q1 выполняется дважды "перенос из начала в конец", она и очередь q2 станут одинаковыми, и очередь q1 будет выглядеть так:



Примечание:

– необходимо использовать метод, который вы написали в пункте (א) .
– нет необходимости сохранять исходное полученное строение очередей.

/продолжение на странице 18/

РАЗДЕЛ ТРЕТИЙ

В этом разделе вопросы четырех разных курсов:
- Компьютерные системы и ассемблер [מערכות מחשב ואסמבלי], стр. 18-19.
- Введение в исследование операций [מבוא לחקר ביצועים], стр. 20-22.
- Вычислительные модели [מודלים חישוביים], стр. 23
- Объектно-ориентированное программирование [תכנות מונחה עצמים] на языке Java, стр. 24-27; объектно-ориентированное программирование на языке C#, стр. 28-31.

Если вы отвечаете на вопросы этого раздела, выберите вопросы только одного курса.

Компьютерные системы и ассемблер [מערכות מחשב ואסמבלי]

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   AL , 6
        MOV   BX , 5CH
        MOV   DL , [BX]
        DEC   BX
A1:     MOV   CL , [BX]
        INC   CL
        MOV   [BX+1] , CL
        DEC   BX
        DEC   AL
        CMP   AL , 0
        JA    A1
        INC   BX
        INC   DL
        MOV   [BX] , DL
```

Дана карта ячеек памяти от 56H до 5CH , перед началом выполнения данного фрагмента программы.

address	56H	57H	58H	59H	5AH	5BH	5CH
value	59H	E2H	C5H	71H	ABH	3FH	8DH

(1) Проследите с помощью таблицы трассировки за выполнением этого фрагмента программы. Включите в таблицу трассировки столбец для каждого из регистров DL , CL , AL , BX . Кроме того, напишите соответствующую карту памяти.

(2) Объясните, что выполняет этот фрагмент программы.

(ב) Во фрагменте данных определена переменная REZ длиной в слово [מילה]: REZ DW ?

Дано, что в стеке [מחסנית] (в SP и в SP+2) хранятся два знаковых (signed) числа длиной в слово.

Напишите фрагмент программы, которая сохранит в переменной REZ число, абсолютное значение [ערך מוחלט] которого – большее из этих двух чисел.

Примечания: – предположите, что оба числа не равны 8000H ;
– предположите, что абсолютные значения у этих двух чисел разные.

/продолжение на странице 19/

9. Во фрагменте данных определены следующие данные:

ARR DB 10 DUP (?)
VAL DB ?
IND DB ?

Напишите фрагмент программы, в котором элементы массива ARR начиная с индекса IND и до конца массива так сдвигают вправо, что самый правый элемент выходит из массива и значение VAL вводится в индекс IND .
Считайте, что массив полон чисел, что VAL инициализирован и что $0 \le IND \le 9$.

Например: для VAL = **15** , IND = **5** , и следующего массива ARR :

index	0	1	2	3	4	**5**	6	7	8	9
value	− 3	11	0	1	8	**− 9**	**5**	**6**	**33**	**73**

После выполнения фрагмента программы будет получен следующий массив ARR :

index	0	1	2	3	4	**5**	6	7	8	9
value	− 3	11	0	1	8	**15**	**− 9**	**5**	**6**	**33**

/продолжение на странице 20/

Введение в исследование операций [מבוא לחקר ביצועים]

10. (א) (1) В таблице ниже дана транспортная задача [בעיית תובלה].

Пункты отправления	**Пункты назначения**				**Предложение**
	1	2	3	4	
1	10	7	9	3	200
2	15	12	4	5	300
3	17	13	6	7	500
Спрос	100	150	250	80	

Перенесите эту таблицу в тетрадь и предложите возможное базовое решение по методу северо-западного угла.

(2) В таблице ниже дано возможное базовое решение транспортной задачи и дано значение u_1 .

Пункты отправления	**Пункты назначения**				**Предложение**	$\mathbf{u_i}$
	1	2	3	4		
1	10 (100)	7 (100)	9	3	200	0
2	15	12 (50)	4 (250)	5	300	
3	17	13	6 (20)	7 (180)	200	
Спрос	100	150	270	180		
$\mathbf{v_j}$						

I. Перенесите эту таблицу в тетрадь и дополните ее значениями u_2 , u_3 , v_1 , v_2 , v_3 , v_4 .

II. Докажите, что решение, данное в таблице, не является оптимальным.

III. Измените цену (c_{ij}) в одной из не-базовых переменных таким образом, чтобы данное в таблице решение транспортной задачи стало бы оптимальным.

Обратите внимание: продолжение вопроса на следующей странице.

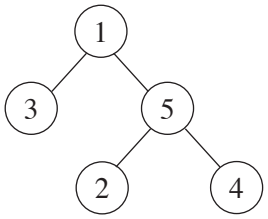
/продолжение на странице 21/

(ב) Граф $G = (V , E)$ – не ориентированный [לא מכוון] , связный и простой – будет называться "разделяющимся" графом, если в нем выполняются следующие условия:

– у него есть по меньшей мере две вершины;

– можно разделить его вершины на две группы, группу א и группу ב, таким образом, что в одной и той же группе не будет двух вершин, соединенных ребром.

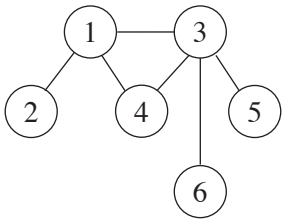
Пример:



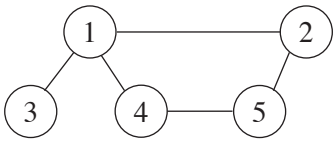
Этот граф является "разделяющимся" графом, потому что можно разделить его вершины таким образом, что группа א будет включать вершины $\{3, 5\}$, а группа ב будет включать вершины $\{1, 2, 4\}$.

(1) Перед вами три графа: G_a , G_b , G_c .

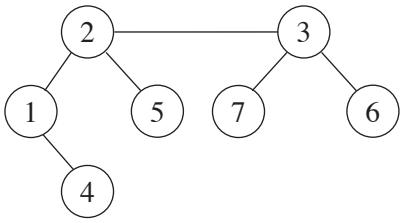
G_c



G_b

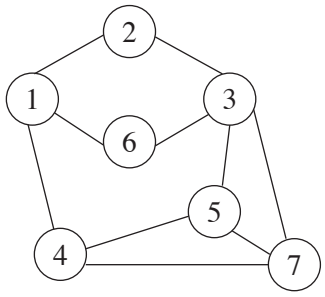


G_a



Укажите, какой из графов является "разделяющимся" графом, а какой – не является. Для каждого "разделяющегося" графа покажите две группы его вершин.

(2) Перед вами не "разделяющийся" граф. Каково минимальное число ребер, которые нужно убрать, чтобы граф считался "разделяющимся"?
Напишите, какие ребра необходимо убрать, и укажите две группы вершин.



/продолжение на странице 22/

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

(א) Перед вами матрица смежностей, представляющая ориентированный [מכוון] граф $G = (V , E)$.

	a	**b**	**c**	**d**	**e**	**f**
a	0	1	0	0	0	0
b	1	0	1	1	0	0
c	1	0	0	1	1	0
d	1	0	0	0	1	1
e	0	0	0	0	0	1
f	0	0	0	0	0	0

(1) Начертите граф G в графической форме.

(2) Найдите компоненты сильной связности [רכיבי קשירות חזקה] (רק"ח) в данном графе.

(3) Каково минимальное число ребер [קשתות], которые нужно добавить в граф G , чтобы он содержал только один компонент сильной связности (רק"ח)? Укажите ребра, которые требуется добавить.

(ב) Перед вами матрица смежностей, представляющая вес пути [משקל מסלול] между вершинами ориентированного и взвешенного [משוקלל] графа $G = (V , E)$.

	a	**b**	**c**	**d**	**e**
a	0	1	3	∞	∞
b	2	0	1	3	12
c	3	∞	0	5	∞
d	4	∞	∞	0	7
e	1	2	∞	8	0

Даны две возможные начальные точки: a или c . Вам нужно попасть в вершину e из одной из двух этих точек.

(1) Какова начальная точка, из которой вес пути в вершину e будет минимальным? Необходимо использовать алгоритм Дейкстры [דייקסטרה] и показать трассировку [מעקב] в каждой итерации.

Трассировка будет включать в каждой итерации группу постоянных вершин (P) и группу временных вершин (T).

(2) Если мы изменим вес ребра от вершины a к вершине c на вес k , $k \geq 0$, изменится ли решение? Объясните.

/продолжение на странице 23/

Вычислительные модели [מודלים חישוביים]

12. L – это язык из всех слов над алфавитом {a,b,c}, которые удовлетворяют всем перечисленным ниже требованиям:

- предпоследняя буква в слове – это буква a .
- слово содержит последовательность bc по меньшей мере два раза.
- буква b встречается в слове четное [זוגי] число раз.
- в слове не встречается последовательность bb .

(א) **(1)** Приведите пример одного слова, принадлежащего языку L .

(2) Приведите пример одного слова, не принадлежащего языку L .

(ב) Докажите, что язык L является регулярным.

Примечание: можно использовать свойства замкнутости [תכונות סגירות].

13. Даны языки L_1 и L_2 :

$L_1 = \{ a^n b^m c^k \mid n, k \geq 0, m = 2k \}$

$L_2 = \{ a^n b^m c^k \mid k , m , n$ больше, чем 0, и должны быть или все четными, или все нечетными $\}$

(א) Является ли язык L_1 регулярным? Если да, постройте полный конечный детерминированный автомат, принимающий этот язык. А если нет, постройте стек-автомат, принимающий этот язык.

(ב) Является ли язык L_2 регулярным? Если да, постройте полный конечный детерминированный автомат, принимающий этот язык. А если нет, постройте стек-автомат, принимающий этот язык.

/продолжение на странице 24/

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

Вопросы 14-15 предназначены для пишущих на языке Java.

14. Для планирования событий в фирме "Завтра" написали класс **Date** – Дата, с тремя атрибутами [תכונות] целого типа: day – день , month – месяц , year – год.

Кроме того, для планирования событий был написан класс **Event** – Событие.

В фирме происходят три вида событий – встречи, телефонные разговоры и задания.

Чтобы хранить информацию о каждом виде событий, были написаны следующие классы:

Meeting – Встреча, **PhoneCall** – Телефонный разговор, **Task** – Задание.

Атрибуты классов определены в соответствии с приведенными ниже требованиями:

Информация, которую нужно хранить о встрече – **Meeting** :

date – типа **Date**

hour – время начала, целого типа

arrNames – имена участников встречи, в массиве типа строка (длина массива равна числу участников)

duration – длительность встречи в минутах, целого типа

location – место встречи, типа строка

Информация, которую нужно хранить о телефонном разговоре – **PhoneCall** :

date – типа **Date**

hour - время начала, целого типа

phoneNumber – номер телефона, по которому следует звонить, типа строка

name – имя человека, которому звонят, типа строка

Информация, которую нужно хранить о задании – **Task :**

date – типа **Date**

hour – время начала, целого типа

title – название задания, типа строка

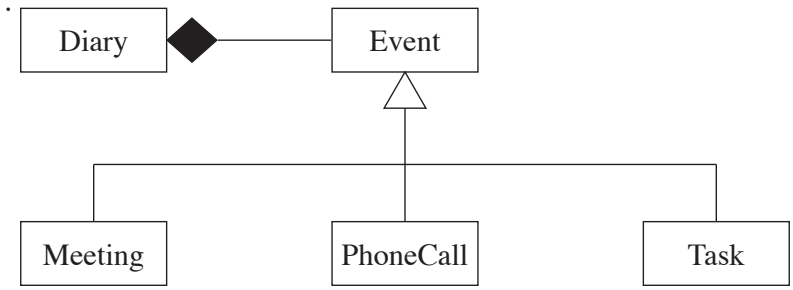
Информация, которую нужно хранить о событии – **Event :**

те требования к хранению информации, которые перечислены выше для разных видов событий и которые должны храниться в классе Event в соответствии с принципами объектно-ориентированного программирования.

В дополнение к этим классам был создан класс **Diary** – Дневник (календарь), хранящий до 1000 событий **Event** в массиве arr длиной 1000 типа **Event .**

/продолжение на странице 25/

Перед вами частичная схема иерархии классов:



Внимание:

наследование обозначается стрелкой ◁—— , а включение обозначается символом ◆——

На вопросы следующих пунктов требуется ответить в соответствии с принципами объектно-ориентированного программирования.

(א) Скопируйте схему в свою тетрадь и добавьте в схему класс **Date**. Запишите заголовок класса Событие – **Event** и его атрибуты.
Считайте, что во всех остальных классах определены соответствующие атрибуты, и что во всех классах определены конструкторы и методы get и set .

(ב) Напишите в классе **Diary** метод с именем allCalls , получающий объект date типа **Date** и возвращающий массив типа **PhoneCall** длиной 100, содержащий все телефонные разговоры в день date.

Внимание:

– Считайте, что в классе **Date** определен метод

public boolean same (Date other)

Этот метод возвращает true, если дата такая же, как other, иначе метод возвращает false. Можно использовать этот метод, не реализуя его.

– Считайте, что в день – date есть ровно 100 телефонных разговоров.

– Считайте, что во время вызова этого метода массив событий arr в классе **Diary** заполнен объектами (без null).

(ג) Перед вами заголовок внутреннего метода match в классе **Event** :

public boolean match (String name)

Вызов этого метода вернет true, если событие (**Event**) – это встреча (**Meeting**), в которой участвует человек, имя которого равно name, или если событие – это телефонный разговор (**PhoneCall**) с человеком, имя которого равно name . Иначе – метод вернет false . Реализуйте этот метод и добавьте методы в соответствующие классы, чтобы выполнялось требуемое действие.
Для каждого метода укажите, к какому классу он добавлен.

Примечание:

в этом пункте запрещено использовать метод instanceof и методы класса Object.

/продолжение на странице 26/

15. Перед вами классы A , B , Tester :

```
public class A {
    public A() { System.out.println("ctor A"); }
    public void foo(int x) {
        System.out.println("A foo int " + x); }
    public void foo(double y) {
        System.out.println("A foo double " + y); }
    public void bar(int x) {
        System.out.println("A bar " + x);
        foo(x);
    }
}

public class B extends A {
    public B() { System.out.println("ctor B"); }
    public void foo(int x) {
        System.out.println("B foo int " + x); }
    public void bar() {
        System.out.println("B bar");
        foo(2);
    }
    public void another(int x) {
        System.out.println("B another " + x);
        super.foo(x);
        foo(2.0 * x);
    }
}

public class Tester {
    public static void main (String [] args)
    {
    }
}
```

Обратите внимание: продолжение вопроса на следующей странице.

/продолжение на странице 27/

Перед вами 15 фрагментов кода. Выберите десять из них.

Подставьте каждый из выбранных фрагментов кода в метод main класса Tester , запишите номер выбранного фрагмента в вашей тетради и укажите, является ли код корректным [תקין] или некорректным [אינו תקין]. Если код корректен – запишите вывод [פלט], а если не корректен – объясните ошибку.

Фрагменты кода не связаны друг с другом.

1. A a = new A(); a.foo(2);	**2.** A a = new A(); ((B)a).foo(3);	**3.** A a = new A(); B b = a; b.foo(2);
4. A x = new B(); x.foo(2);	**5.** A x = new B(); x.bar();	**6.** A a = new A(); a.bar(3);
7. A a = new A(); a.bar();	**8.** B b = new B(); b.bar();	**9.** B b = new B(); b.bar(3);

10.
B b = new B();
b.foo(2);
b.foo(2.0);
	11.
	A a = new A();
	a.foo(2);
	a.foo(2.0);
		12.
		B b = new A();
		b.another(2);

13. A a = new A(); a.another(2);	**14.** B x = new B(); x.another(2);	**15.** A x = new B(); x.another(2);

/продолжение на странице 28/

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

Вопросы 16-17 предназначены для пишущих на языке C#

16. Для планирования событий в фирме "Завтра" написали класс **Date** – Дата, с тремя атрибутами [תכונות] целого типа: day – день , month – месяц , year – год.

Кроме того, для планирования событий был написан класс **Event** – Событие.

В фирме происходят три вида событий – встречи, телефонные разговоры и задания.

Чтобы хранить информацию о каждом виде событий, были написаны следующие классы:

Meeting – Встреча, **PhoneCall** – Телефонный разговор, **Task** – Задание.

Атрибуты классов определены в соответствии с приведенными ниже требованиями:

Информация, которую нужно хранить о встрече – **Meeting** :

date – типа **Date**

hour – время начала, целого типа

arrNames – имена участников встречи, в массиве типа строка (длина массива равна числу участников)

duration – длительность встречи в минутах, целого типа

location – место встречи, типа строка

Информация, которую нужно хранить о телефонном разговоре – **PhoneCall** :

date – типа **Date**

hour - время начала, целого типа

phoneNumber – номер телефона, по которому следует звонить, типа строка

name – имя человека, которому звонят, типа строка

Информация, которую нужно хранить о задании – **Task :**

date – типа **Date**

hour – время начала, целого типа

title – название задания, типа строка

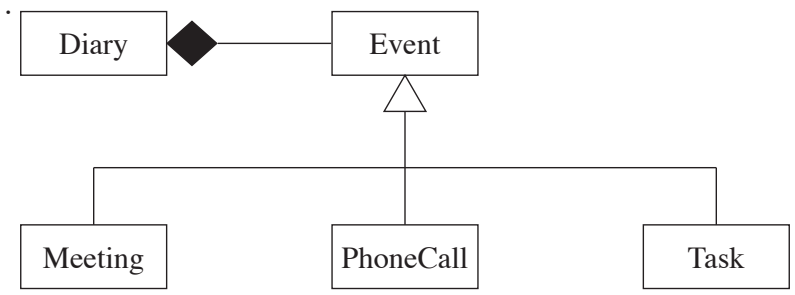
Информация, которую нужно хранить о событии – **Event :**

те требования к хранению информации, которые перечислены выше для разных видов событий и которые должны храниться в классе Event в соответствии с принципами объектно-ориентированного программирования.

В дополнение к этим классам был создан класс **Diary** – Дневник (календарь), хранящий до 1000 событий **Event** в массиве arr длиной 1000 типа **Event .**

/продолжение на странице 29/

Перед вами частичная схема иерархии классов:



Внимание:

наследование обозначается стрелкой ◁—— , а включение обозначается символом ◆——

На вопросы следующих пунктов требуется ответить в соответствии с принципами объектно-ориентированного программирования.

(א) Скопируйте схему в свою тетрадь и добавьте в схему класс **Date**. Запишите заголовок класса Событие – **Event** и его атрибуты.
Считайте, что во всех остальных классах определены соответствующие атрибуты, и что во всех классах определены конструкторы и методы get и set .

(ב) Напишите в классе **Diary** метод с именем AllCalls , получающий объект date типа **Date** и возвращающий массив типа **PhoneCall** длиной 100, содержащий все телефонные разговоры в день date.

Внимание:

– Считайте, что в классе **Date** определен метод

public bool Same (Date other)

Этот метод возвращает true, если дата такая же, как other, иначе метод возвращает false. Можно использовать этот метод, не реализуя его.

– Считайте, что в день – date есть ровно 100 телефонных разговоров.

– Считайте, что во время вызова этого метода массив событий arr в классе **Diary** заполнен объектами (без null).

(ג) Перед вами заголовок внутреннего метода Match в классе **Event** :

public virtual bool Match (string name)

Вызов этого метода вернет true, если событие (**Event**) – это встреча (**Meeting**) в которой участвует человек, имя которого равно name , или если событие – это телефонный разговор (**PhoneCall**) с человеком, имя которого равно name . Иначе – метод вернет false . Реализуйте этот метод и добавьте методы в соответствующие классы, чтобы выполнялось требуемое действие.
Для каждого метода укажите, к какому классу он добавлен.

Примечание:

в этом пункте запрещено использовать метод is и as и методы класса Object.

16. Перед вами классы A , B , Tester :

```
public class A {
    public A() { Console.WriteLine("ctor A"); }
    public virtual void Foo(int x) {
        Console.WriteLine("A Foo int " + x); }
    public void Foo(double y) {
        Console.WriteLine("A Foo double " + y); }
    public void Bar(int x) {
        Console.WriteLine("A Bar " + x);
        Foo(x);
    }
}

public class B : A {
    public B() { Console.WriteLine("ctor B"); }
    public override void Foo(int x) {
        Console.WriteLine("B Foo int " + x); }
    public void Bar() {
        Console.WriteLine("B Bar");
        Foo(2);
    }
    public void Another(int x) {
        Console.WriteLine("B Another " + x);
        base.Foo(x);
        Foo(2.0 * x);
    }
}

public class Tester {
    public static void Main (string [] args)
    {
    }
}
```

Обратите внимание: продолжение вопроса на следующей странице.

Перед вами 15 фрагментов кода. Выберите десять из них.

Подставьте каждый из выбранных фрагментов кода в метод Main класса Tester , запишите номер выбранного фрагмента в вашей тетради и укажите, является ли код корректным [תקין] или некорректным [אינו תקין]. Если код корректен – запишите вывод [פלט], а если не корректен – объясните ошибку.

Фрагменты кода не связаны друг с другом.

1. A a = new A(); a.Foo(2);	**2.** A a = new A(); ((B)a).Foo(3);	**3.** A a = new A(); B b = a; b.Foo(2);
4. A x = new B(); x.Foo(2);	**5.** A x = new B(); x.Bar();	**6.** A a = new A(); a.Bar(3);
7. A a = new A(); a.Bar();	**8.** B b = new B(); b.bar();	**9.** B b = new B(); b.Bar(3);

10.
B b = new B();
b.Foo(2);
b.Foo(2.0);
	11.
	A a = new A();
	a.Foo(2);
	a.Foo(2.0);
		12.
		B b = new A();
		b.Another(2);

13. A a = new A(); a.Another(2);	**14.** B x = new B(); x.Another(2);	**15.** A x = new B(); x.Another(2);

Желаем успеха!