Государство Израиль
Министерство просвещения
Тип экзамена: на аттестат зрелости
Время проведения экзамена: лето 2026 года
Номер вопросника: 899371

מדינת ישראל
משרד החינוך
סוג הבחינה: בגרות
מועד הבחינה: קיץ תשפ"ו, 2026
מספר השאלון: 899371

Компьютеры

מדעי המחשב

Указания

הוראות

а. Продолжительность экзамена: 2 часа 30 минут.

א. משך הבחינה: שעתיים וחצי.

б. Строение вопросника и ключ к оценке:
в этом вопроснике два раздела.

Раздел первый:	(2x20)	–	40 баллов
Раздел второй:	(2x30)	–	60 баллов
	Всего	–	100 баллов

ב. מבנה השאלון ומפתח ההערכה:
בשאלון זה שני פרקים.

פרק ראשון	(20×2)	– 40	נקודות
פרק שני	(30×2)	– 60	נקודות
סך הכול		– 100	נקודות

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. Особое указание:
Все программы, которые вы должны написать на языке программирования, пишите на одном и том же языке – Java или C# .

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

ד. הוראה מיוחדת:
את כל התוכניות שיש לכתוב בשפת מחשב כִּתבו בשפה אחת בלבד – Java או C# .

הערה: לא יורדו נקודות אם תכתבו בתוכניות אות גדולה במקום אות קטנה או להפך.

יש לכתוב במחברת הבחינה בלבד. יש לרשום "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 черновых записей на листах, не относящихся к экзаменационной тетради, может привести к тому, что экзамен будет аннулирован.

Желаем успеха!

בהצלחה!

В о п р о с ы

В этом вопроснике два раздела.

Вы должны ответить на вопросы двух разделов в соответствии с указаниями в каждом разделе.

Примечание: в каждом вопросе, где требуется ввод, нет необходимости проверять корректность вводимых данных.

Для решающих на языке Java: в каждом вопросе, где требуется ввод, считайте, что в программе написана команда:

Scanner scan = new Scanner (System.in);

РАЗДЕЛ ПЕРВЫЙ (40 баллов)

Ответьте на два из вопросов 1–3 (за каждый вопрос – 20 баллов).

1. Ниже приведен фрагмент программы, написанный на языках Java и C# .

Java	C#

int n = 23;
int steps = 0;
while (n > 0) {
  int root = (int)(Math.sqrt(n));
  int square = (int)(Math.pow(root, 2));
  n = n – square;
  steps ++;
}
System.out.println(steps);
	int n = 23;
	int steps = 0;
	while (n > 0) {
	  int root = (int)(Math.Sqrt(n));
	  int square = (int)(Math.Pow(root, 2));
	  n = n – square;
	  steps ++;
	}
	Console.WriteLine(steps);

(א) При помощи таблицы трассировки [טבלת המעקב] выполните трассировку фрагмента программы и напишите число, которое этот фрагмент выводит.
Таблица трассировки должна содержать столбец для каждой из следующих переменных: square , root , steps , n .

(ב) Напишите пример числа больше 1 , которое нужно присвоить n (в начале фрагмента), для того чтобы этот фрагмент программы вывел 1 (не нужно показывать трассировку).

/продолжение на странице 3/

2. Параметр max – положительное целое число, обозначающее максимальную разрешенную скорость в км/ч для автомобилей на определенном участке шоссе.
Параметр min – положительное целое число, обозначающее минимальную разрешенную скорость в км/ч для автомобилей на том же участке шоссе.

(א) Напишите внешний метод getStatus на языке Java или GetStatus на языке C#, который принимает положительное целое число speed , обозначающее скорость какого-либо автомобиля в км/ч на данном участке шоссе, а также параметры max и min .
Метод возвращает целое число согласно следующим условиям:
- если скорость автомобиля не менее чем на 10 км/ч выше максимальной разрешенной скорости, метод возвращает 1.
- если скорость автомобиля не менее чем на 10 км/ч ниже минимальной разрешенной скорости, метод возвращает –1.
- если не выполняется ни одно из двух указанных выше условий, метод возвращает 0.

speeds – массив с элементами целочисленного типа, содержащий информацию о скоростях разных автомобилей на этом участке шоссе (каждая ячейка массива представляет скорость другого автомобиля). Массив никак не упорядочен.

(ב) Напишите внешний метод getAllStatus на языке Java или GetAllStatus на языке C#, который принимает массив speeds и параметры max и min .
Метод возвращает новый массив с элементами целочисленного типа, равный по длине массиву speeds , ячейки которого содержат 1 или –1 , или 0 , в зависимости от автомобиля, расположенного под тем же индексом в массиве speeds (в соответствии с условиями в пункте (א)).
Пример: для max = 70 , min = 30 и массива speeds:

	0	1	2	3	4	5	6
speeds	80	50	79	90	20	100	25

будет возвращен такой массив:

0	1	2	3	4	5	6
1	0	0	1	–1	1	0

Примечания: – используйте метод, который вы написали в пункте (א).
– нельзя изменять массив speeds

(ג) Автомобиль будет оштрафован, если скорость его движения на этом участке шоссе выше максимальной разрешенной скорости не менее чем на 10 км/ч или ниже минимальной разрешенной скорости не менее чем на 10 км/ч.
Напишите внешний метод mostFined на языке Java или MostFined на языке C#, который принимает массив speeds и параметры max и min .
Метод возвращает true , если по меньшей мере половина из автомобилей в массиве speeds будет оштрафована, иначе метод возвращает false .
Пример: для массива, данного в пункте (ב), метод вернет true , поскольку из семи автомобилей в массиве четыре будут оштрафованы (автомобили с индексами 0 , 3 , 4 , 5).
Примечание: используйте метод, который вы написали в пункте (ב).

/продолжение на странице 4/

3. Когда играют в игру "**в яблочко**", то заранее выбирают секретную строку, состоящую из букв A–Z , расположенных в произвольном порядке. Каждая из букв может появляться в секретной строке не более одного раза (таким образом, длина строки находится в пределах от 1 до 26).

Правила игры:

игрок отгадывает какую-то строку (полагая, что это и есть секретная строка). За каждую букву в строке он получает оценку в соответствии с точностью его отгадки:

– оценка "**в яблочко**" – за отгадывание буквы и ее места в секретной строке.
– оценка "**попадание**" – за отгадывание буквы в секретной строке, но на неправильном месте.
– оценка "**промах**" – за отгадывание буквы, которой нет в секретной строке.

Пример: за отгадку "A G C F" для секретной строки "F G A V" игрок получит следующие оценки:

одну оценку "в яблочко" за отгадывание буквы G на правильном месте в секретной строке (индекс 1).

две оценки "попадание" за отгадывание букв A и F в секретной строке, но на неправильных местах.

одну оценку "промах" за неправильное отгадывание буквы C , которой нет в секретной строке.

Напишите внешний метод guess на языке Java или Guess на языке C#, который принимает два параметра:

секретную строку secretStr и строку-отгадку playerStr .

Предположите, что обе строки корректны (то есть, в них есть только буквы A–Z, и каждая буква появляется в строке не более одного раза).

Если длина строки-отгадки отличается от длины секретной строки, то метод выведет сообщение об ошибке: "Size Error".

Если длина строки-отгадки равна длине секретной строки, то метод выведет количество оценок "в яблочко", количество оценок "попадание" и количество оценок "промах" для строки, которую отгадал игрок (каждое число будет выведено с новой строки вывода).

/продолжение на странице 5/

РАЗДЕЛ ВТОРОЙ (60 баллов)

Ответьте на два из вопросов 4–6 (за каждый вопрос – 30 баллов).

4. Дан класс **Order** , представляющий информацию о заказе какого-либо товара; в этом классе четыре атрибута:

- productID – идентификационный код товара от 0 до 99 (включительно), целочисленного типа
- price – цена товара, действительного типа
- quantity – количество единиц товара, целочисленного типа
- delivery –включает ли заказ доставку на дом, булева типа (true – для заказа, включающего доставку, false – для заказа, не включающего доставку)

Покупатель, заказавший доставку на дом, должен заплатить **дополнительные** 100 шекелей. Если один заказ содержит не менее 10 единиц товара и суммарная стоимость этого заказа (не включая доставку) выше 170.00 шекелей, то доставка будет **бесплатной**.

Считайте, что для атрибутов этого класса определены методы get/Get и set/Set.

(א) Напишите конструктор класса **Order ,** который получает идентификационный код товара productID и цену товара price . Конструктор инициализирует атрибуты класса, создавая заказ товара productID по цене price , в количестве одной единицы, с доставкой на дом.

(ב) **(1)** Напишите в классе **Order** внутренний метод с названием totalPrice на языке Java или TotalPrice на языке C# , который возвращает общую стоимость всех товаров в заказе (действительное число) **без стоимости доставки.**

(2) Напишите в классе **Order** внутренний метод с названием totalCost на языке Java или TotalCost на языке C# ; метод возвращает общую сумму, которую должен заплатить покупатель (действительное число), **включающую стоимость доставки** (если заказ не включает доставку на дом или доставка бесплатна, то метод вернет стоимость заказа без стоимости доставки).

Примечание: используйте метод, который вы написали в подпункте (ב)(1).

(ג) Дан массив allOrders с элементами типа **Order** , содержащий заказы, выполненные в определенную неделю. Массив никак не упорядочен, и в нем нет значений null .
Напишите внешний метод с названием sumQuantity на языке Java или SumQuantity на языке C#, который получает массив allOrders . Метод возвращает новый массив длиной 100, с элементами целочисленного типа, в каждой ячейке которого находится суммарное количество заказанных единиц определенного товара, идентификационный код которого равен индексу ячейки.
Например, если для товара с идентификационным кодом 0 заказаны 15 единиц в одном заказе, 20 единиц в другом и 3 единицы в третьем, то в ячейке с индексом 0 будет находиться число 38 .

/продолжение на странице 6/

5. Дан класс **Birthdate** – дата рождения; в этом классе три атрибута:

- day – день (от 1 до 31), целочисленного типа
- month – месяц (от 1 до 12), целочисленного типа
- year – год (положительное число), целочисленного типа

Считайте, что для атрибутов этого класса определены методы get/Get и set/Set.

(א) Напишите в классе **Birthdate** внутренний метод с названием sameDate на языке Java или SameDate на языке C# , который получает другой объект other типа **Birthdate**.
Метод возвращает true , если даты рождения одинаковы, иначе метод возвращает false .

В IT-компании "Социальная сеть" разработали систему, которая вычисляет сходство между клиентами компании. Для этой цели в компании определили класс **Client** – клиент;
в этом классе три атрибута:

- firstNameInitial – первая буква имени (большая английская буква), символьного типа (char)
- lastNameInitial – первая буква фамилии (большая английская буква), символьного типа
- birthdate – дата рождения, типа **Birthdate**

Считайте, что для атрибутов этого класса определены методы get/Get и set/Set.

(ב) **(1)** Инициалы двух клиентов одинаковы, если одинаковы первые буквы их имен и одинаковы первые буквы их фамилий.
Напишите в классе **Client** внутренний метод с названием sameInitials на языке Java или SameInitials на языке C# , который получает другой объект other типа **Client.** Метод возвращает true , если инициалы обоих клиентов одинаковы, иначе метод возвращает false .

(2) В этой системе два клиента считаются похожими, если у них совпадают **как** инициалы, **так и** даты рождения (если выполняется только одно из условий, то клиенты не похожи).
Напишите в классе **Client** внутренний метод с названием areSame на языке Java или AreSame на языке C# , который получает другой объект other типа **Client .** Метод возвращает true , если два клиента похожи, иначе метод возвращает false .
Используйте методы, которые вы написали в пункте (א) и в подпункте (ב)(1).

(ג) В этой системе клиент считается уникальным, если нет ни одного похожего на него клиента (в соответствии с определением в подпункте (ב)(2)).
Напишите внешний метод с названием uniques на языке Java или Uniques на языке C#, который получает массив всех клиентов в системе clients с элементами типа **Client**
Массив никак не упорядочен, и в нем нет значений null . Метод выводит первую букву имени уникальных клиентов, находящихся в массиве (каждая буква на новой строчке).
Используйте метод, который вы написали в подпункте (ב)(2).

/продолжение на странице 7/

6. **(א)** Реализуйте следующий метод:

Java – public static int leftDigits (int num, int k)
C# – public static int LeftDigits (int num, int k)

num – положительное число, а k – положительное число, которое меньше количества цифр в num (то есть длины числа num).

Метод возвращает число, состоящее из k цифр с левой стороны числа num .

Пример: для k = 5 и num = **26004**3100 метод вернет число 26004 .

(ב) Реализуйте следующий метод:

Java – public static int rightDigits (int num, int k)
C# – public static int RightDigits (int num, int k)

num – положительное число, а k – положительное число, которое меньше количества цифр в num (то есть длины числа num).

Метод возвращает число, состоящее из k цифр с правой стороны числа num (считайте, что возвращаемое число корректно и не начинается с цифры 0).

Пример: для k = 5 и num = 2600**43100** метод вернет число 43100 .

(ג) Положительное целое число называется "сбалансированное число Капрекара", если выполняется следущее условие:
когда число возводят в квадрат, а результат разделяют посередине на две стороны равной длины (то есть результат – это число с четным количеством цифр, и количество цифр с каждой стороны одинаково), то сумма этих двух сторон равна исходному числу.

Два примера: числа **9** и **4950** являются сбалансированными числами Капрекара.

Объяснение:

Число	Квадрат числа	Левая сторона	Правая сторона	Сумма
9	$9^2 = 81$	8	1	$8 + 1 =$ **9**
4950	$4950^2 = 24502500$	2450	2500	$2450 + 2500 =$ **4950**

Напишите внешний метод с названием isKaprekar на языке Java или IsKaprekar на языке C#, который получает положительное целое число num . Метод возвращает true , если num – это сбалансированное число Капрекара, иначе метод возвращает false .

Используйте методы, которые вы написали в пунктах (א) и (ב).

Считайте, что при возведении num в квадрат получается результат с четным количеством цифр и обе его стороны – корректные числа.

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.