

МИНИСТЕРСТВО НА ОБРАЗОВАНИЕТО И НАУКАТА

ДЪРЖАВЕН ЗРЕЛОСТЕН ИЗПИТ ПО ИНФОРМАТИКА

22 май 2026 г.

ПРОФИЛИРАНА ПОДГОТОВКА ВАРИАНТ 2

ЧАСТ 1 (Време за работа: 90 минути)

За задачите от 1. до 16. включително изберете точно един от отговорите!

1. Кое от следните твърдения е правилно за абстрактния тип данни „опашка“?

- А) Опашката работи на принципа LIFO и позволява достъп само до елемента в края на опашката.
- Б) Опашката работи на принципа FIFO и позволява достъп за извличане само до елемента в началото на опашката.
- В) До всеки елемент в опашката има пряк достъп и може да бъде премахнат по време на работа.
- Г) Новите елементи могат да се добавят само преди елемента в началото на опашката.

2. Какъв тип релация между таблицата, съхраняваща информация за клиенти и таблица, съхраняваща информация за поръчките, е необходима в релационна база данни, ако един клиент може да прави много поръчки, а всяка поръчка принадлежи на един клиент?

- А) Едно към едно (1 : 1)
- Б) Едно към много (1 : N)
- В) Много към много (M : N)
- Г) Няма релация между таблиците

3. Какъв ще бъде резултатът при опит за компилиране и изпълнение на дадената програма?

C#	Java
<pre>using System; class BankAccount { private double balance = 1000.5; }</pre>	<pre>class BankAccount { private double balance = 1000.5; }</pre>

<pre> class Program { static void Main() { BankAccount account = new BankAccount(); account.balance -= 500.25; Console.WriteLine((int)account.balance); } } </pre>	<pre> public class Main { public static void main(String[] args) { BankAccount account = new BankAccount(); account.balance -= 500.25; System.out.println((int)account.balance); } } </pre>
--	---

- А) На стандартния изход ще се изведе 500
- Б) Грешка по време на изпълнение
- В) Грешка при компилация
- Г) На стандартния изход ще се изведе -500

4. Прилага се двоично търсене в масива: {16, 20, 30, 35, 54, 60, 67, 69, 80, 87, 92}. Търси се числото 20. Кой е вторият елемент от списъка, с който ще бъде направено сравнение?

- А) 20
- Б) 54
- В) 35
- Г) 30

5. В списък `cities`, обект от стандартния клас `List<string>` (за C#) / `ArrayList<String>` (Java), трябва да се добави име на нов град само ако той НЕ присъства в него. Чрез кой от следните програмни фрагменти няма да се допусне дублиране на елементи в списъка?

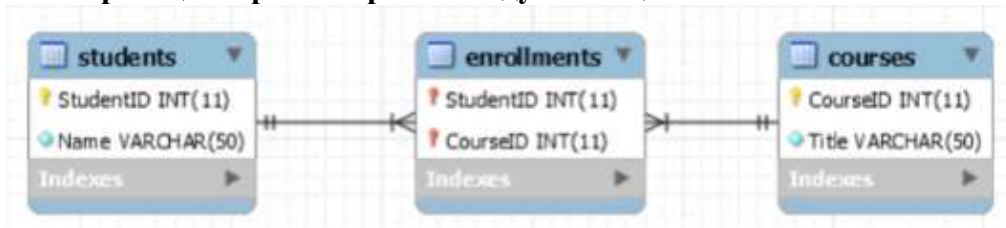
	C#	Java
А)	<code>if (cities.Contains(cityName)) cities.Add(cityName);</code>	<code>if (cities.contains(cityName)) cities.add(cityName);</code>
Б)	<code>if (!cities.Contains(cityName)) cities.Add(cityName); }</code>	<code>if (!cities.contains(cityName)) cities.add(cityName);</code>
В)	<code>if (cities.IndexOf(cityName) > 0) cities.Add(cityName);</code>	<code>if (cities.indexOf(cityName) > 0) cities.add(cityName);</code>
Г)	<code>if (cities.Find(cityName) >= 0) cities.Add(cityName);</code>	<code>if (cities.find(cityName) >= 0) cities.add(cityName);</code>

6. Коя от изброените двойки преходи между етапи в жизнения цикъл на информационна система е логически НЕправилна?

Забележка: Символа „→“ означава „следващ етап“.

- А) Анализ → Дизайн
- Б) Поддръжка → Анализ
- В) Внедряване → Поддръжка
- Г) Дизайн → Разработка

7. Какъв тип релация е реализирана между таблиците *students* и *courses*?

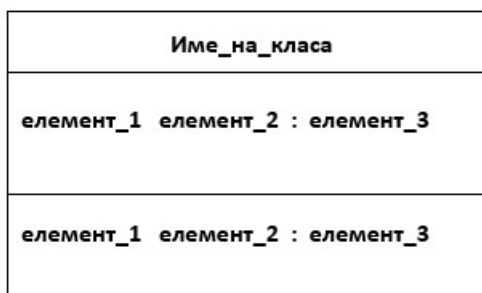


- А) Едно към едно (1 : 1)
- Б) Едно към много (1 : N)
- В) Много към едно (N : 1)
- Г) Много към много (M : N)

8. В кой от случаите е най-подходящо променливата *val* да е статична?

- А) Всяка инстанция трябва да има собствена стойност на променливата *val*
- Б) Стойността на променливата *val* трябва да бъде споделена от всички обекти
- В) Стойността на променливата *val* се променя при всяко извикване на метод
- Г) Променливата *val* е локална за метод

9. В UML диаграма на клас атрибутите и методите се описват с три елемента, както са представени на схемата. Какво е значението на елемент_1?



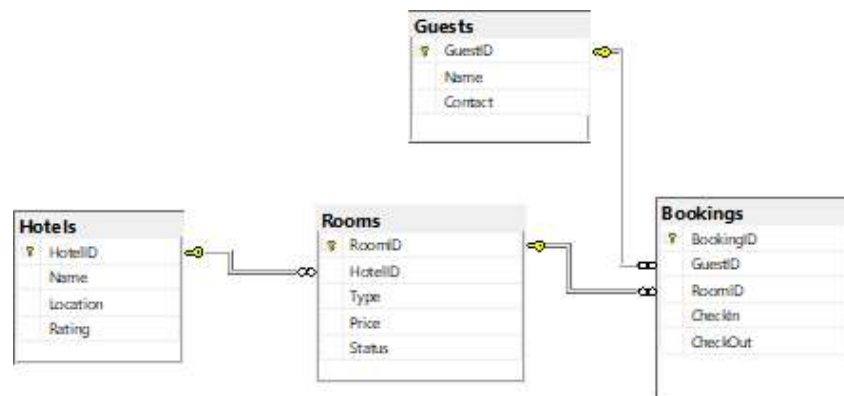
- А) тип

- Б) име
- В) приоритет
- Г) видимост

10. В коя от дадените двойки обекти първият е във връзка „има“ (has a) с втория?

- А) автомобил – електрически автомобил
- Б) електрически автомобил – батерия
- В) електрически автомобил – автомобил
- Г) батерия – електрически автомобил

11. Какъв тип е връзката между таблиците **Hotels (хотели) и **Guests** (гости) от дадената релационна база от данни.**



- А) Няма връзка
- Б) От тип „едно към едно“
- В) От тип „много към много“
- Г) От тип „едно към много“

12. Какво ще се изведе на стандартния изход в резултат от изпълнението на следния програмен фрагмент?

Гарантирано е, че всички необходими библиотеки са включени.

C#
<pre> try { Queue <int> q = new Queue<int>(); q.Enqueue(1); Console.WriteLine(q.Dequeue() + q.Dequeue()); } catch (Exception e) { </pre>

```

        Console.WriteLine("The queue is empty!");
    }
    finally
    {
        Console.WriteLine("Goodbye!");
    }
}

```

Java

```

try {
    Queue<Integer> q = new LinkedList();
    q.add(1);
    System.out.println(q.remove() + q.remove());
} catch (Exception e) {
    System.out.println("The queue is empty!");
} finally {
    System.out.println("Goodbye!");
}

```

A) 2

Б) 2

Goodbye!

В) 1

The queue is empty!

Г) The queue is empty!

Goodbye!

13. Дадена е диаграма на релационна база данни за стрийминг платформа, съдържаща таблиците Users, Videos и Likes.



Ако таблиците съдържат следните кортежи:

Users

Id	Username	Age

1	gamerX	15
2	melody	17
3	codeKid	16
4	ChillGuy	15

Videos

Id Title CreatorId		

1	Minecraft Tips	1
2	Guitar Basics	2
3	C# for Beginners	3
4	Funny Shorts	NULL

Likes

UserId VideoId	

1	1
2	1
4	1
2	2
3	3

Какъв ще е резултата от изпълнение на заявката?

```
SELECT Title FROM Videos
LEFT JOIN Likes ON Videos.Id = Likes.VideoId
LEFT JOIN Users ON Likes.UserId = Users.Id
WHERE Age < 16;
```

- A) Minecraft Tips
Minecraft Tips
- Б) Minecraft Tips
- В) Minecraft Tips
Minecraft Tips
Minecraft Tips
- Г) Няма да име върнати редове

14. Какво ще се изведе на стандартния изход при изпълнението на следния програмен фрагмент?

C#	Java
<pre>string s = "aba"; s.Replace('a', 'x'); s = s + s[0]; Console.WriteLine(s); Console.WriteLine(s.Replace("b", "").Length);</pre>	<pre>String s = "aba"; s.replace('a', 'x'); s = s + s.charAt(0); System.out.println(s); System.out.println(s.replace("b", "").length());</pre>

- A) xbx
3
- Б) aba
2
- В) abaa
3

Г) ааа
3

15. Кое от изброеното в приложение с графичен потребителски интерфейс определя дали дадена контрола с име Cont (бутон, етикет, текстово поле или др.) е видим върху формата по време на изпълнение, но не реагира на събития?

	C#	Java
A)	Cont.Visible = false;	Cont.setVisible(false);
Б)	Cont.Refresh();	Cont.redraw();
B)	Cont.Enabled = false;	Cont.setEnabled(false);
Г)	Cont.Dispose();	Cont.dispose();

16. Кое от изброените съдържа грешка при създаване на обект от класа Car, описан по дадения начин?

C#	Java
<pre>public class Car { private string Brand { get; set; } private int Year { get; set; } public Car() { } public Car(string brand,int year) { Brand = brand; Year = year; } }</pre>	<pre>class Car { private String brand; private int year; public Car() { } public Car(String brand, int year) { this.brand = brand; this.year = year; } public String getBrand() { return brand; } public void setBrand(String brand) { this.brand = brand; } public int getYear() { return year; } public void setYear(int year) { this.year = year; } }</pre>

A) Car car1 = new Car();
Б) Car car2 = new Car("2020",'7');
B) Car car3 = new Car("Audi", 2020);
Г) Car car4 = new Car(Brand = "Audi", Year = 2020);

Отговорите на задачите от 17. до 24. включително запишете в полетата за отговори!

17. В полето за отговор запишете какво ще се изведе на стандартния изход след изпълнение на програмния фрагмент?

C#	Java
<pre>int n = 934; int a = -1; int b = 0; while (n > 0) { int x = n % 10; if (x > 0) { int y = (int)Math.Sqrt(x); if (y * y == x) { b++; if (x > a) a = x; if (x == 9) break; } } n /= 10; } Console.WriteLine("a = " + a); Console.WriteLine("b = " + b);</pre>	<pre>int n = 934; int a = -1; int b = 0; while (n > 0) { int x = n % 10; if (x > 0) { int y = (int) Math.sqrt(x); if (y * y == x) { b++; if (x > a) a = x; if (x == 9) break; } } n /= 10; } System.out.println("a = " + a); System.out.println("b = " + b);</pre>

18. В полето за отговор запишете какво ще се изведе на стандартния изход, след изпълнение на програмния фрагмент. Необходимите библиотеки са включени.

C#
<pre>public static int FirstIndex(int[] arr, int target) { for (int i = 0; i < arr.Length; i++) if (arr[i] == target) return i; return -1; } public static int LastIndex(int[] arr, int target) { for (int i = arr.Length - 1; i >= 0; i--) if (arr[i] == target) return i; return -1; } static void Main()</pre>

```

{
    int[] a = { 7, 2, 7, 4, 7, 3 };
    int p1 = FirstIndex(a, 7);
    int p2 = LastIndex(a, 7);
    Array.Sort(a);
    int p3 = FirstIndex(a, 7);
    int p4 = LastIndex(a, 7);
    Console.WriteLine(p1 + "; " + p2);
    Console.WriteLine(p3 + "; " + p4);
}

```

Java

```

public static int firstIndex(int[] arr, int target) {
    for (int i = 0; i < arr.length; i++)
        if (arr[i] == target)
            return i;
    return -1;
}
public static int lastIndex(int[] arr, int target) {
    for (int i = arr.length - 1; i >= 0; i--)
        if (arr[i] == target)
            return i;
    return -1;
}
public static void main(String[] args) {
    int[] a = { 7, 2, 7, 4, 7, 3 };
    int p1 = firstIndex(a, 7);
    int p2 = lastIndex(a, 7);
    Arrays.sort(a);
    int p3 = firstIndex(a, 7);
    int p4 = lastIndex(a, 7);
    System.out.println(p1 + "; " + p2);
    System.out.println(p3 + "; " + p4);
}

```

19. В полето за отговор запишете какво ще се изведе на стандартния изход след изпълнение на програмния фрагмент.

C#	Java
<pre> int a = 0; int b = 0; for (int i = 1; i <= 5; i++) { if (i % 2 == 0) a += i; else b += 1; } </pre>	<pre> int a = 0; int b = 0; for (int i = 1; i <= 5; i++) { if (i % 2 == 0) a += i; else b += 1; } </pre>

<pre> } Console.WriteLine(a); Console.WriteLine(b); </pre>	<pre> } System.out.println(a); System.out.println(b); </pre>
--	--

20. На всеки ред в текстовия файл `data.txt` е записана информация за продажбата на някакъв продукт във формат

<продукт>;<количество>;<цена>

data.txt
Хляб;5;1.20
Мляко;3;2.50
Хляб;2;1.20
Сирене;1;8.00
Мляко;4;2.50

Програмният фрагмент трябва да изчисли общата сума от продажбите на продукт "Хляб" и да я **запише** в текстов файл **result.txt**. В кода редовете, означени с (1) и (3), съдържат грешки, а (2) – липсваща част от кода. В полетата за отговори срещу (1) и (3) запишете коригираните редове, а срещу (2) – липсващия код, за да работи фрагмента от програмата правилно.

Забележка: Приема се, че нужните библиотеки са включени и файлът `data.txt` се намира в текущата директория.

C#
<pre> try { StreamReader reader = new StreamReader("data.txt"); StreamWriter writer = StreamWriter("result.txt"); // (1) { double total = 0; string line; while ((line = ____ (2) ____ .ReadLine()) != null) // (2) { string[] parts = line.Split(";"); if (parts[0] == "Хляб") { int quantity = int.Parse(parts[1]); double price = double.Parse(parts[2]); total += quantity * price; } } reader.Close(); Console.WriteLine("Обща сума за хляб: " + total); // (3) } } </pre>

```

        writer.Close();
    }
}
catch (Exception ex)
{
    Console.WriteLine("Error: " + ex.Message);
}

```

Java

```

try {
    Scanner reader = new Scanner(new File("data.txt"));
    PrintWriter writer = PrintWriter("result.txt"); // (1)

    double total = 0;
    while (reader.hasNextLine())
    {
        String line = ____ (2) ____ .nextLine(); // (2)
        String[] parts = line.split(";");
        if (parts[0].equals("Хляб"))
        {
            int quantity = Integer.parseInt(parts[1]);
            double price = Double.parseDouble(parts[2]);
            total += quantity * price;
        }
    }
    reader.close();
    System.out.println("Обща сума за хляб: " + total); // (3)
    writer.close();
}
catch (Exception ex)
{
    System.out.println("Error: " + ex.getMessage());
}

```

21. Даденият програмен фрагмент трябва да провери дали дадено цяло число е четно, но съдържа грешки и не се компилира. В полето за отговор напишете номера на съответния ред с грешка и предложение за корекция на кода, така че фрагмента да работи коректно.

C#	Java
<pre> 1 static void IsEven(int n) 2 { 3 if (n % 2 = 0) return true; 4 return false; 5 } 6 static void Main() { 7 bool res = IsEven("5"); 8 } </pre>	<pre> 1 static void isEven(int n) 2 { 3 if (n % 2 = 0) return true; 4 return false; 5 } 6 public static void main(String[] args) { 7 boolean res = isEven("5"); 8 } </pre>

22. В полето са свободни отговори запишете какъв резултат ще се изведе на стандартния изход след изпълнение на програмния фрагмент.

C#	Java
<pre>int a = 1, b = 10, c = 0, tot = 0; do { a = a * 2; c = c + 1; b = b - a; tot = (tot + b) % (c + 1); } while (b > 0); Console.WriteLine(b); Console.WriteLine(c); Console.WriteLine(tot);</pre>	<pre>int a = 1, b = 10, c = 0, tot = 0; do { a = a * 2; c = c + 1; b = b - a; tot = (tot + b) % (c + 1); } while (b > 0); System.out.println(b); System.out.println(c); System.out.println(tot);</pre>

23. В полето за отговор запишете какво ще се изведе на стандартния изход, след изпълнение на програмния фрагмент.

C#
<pre>int n = 5738462; Stack<int> st = new Stack<int>(); Queue<int> q = new Queue<int>(); while (n != 0) { int k = n % 10; if (k % 2 == 0) st.Push(k); else q.Enqueue(k); n /= 10; } if (st.Count >= 2) { st.Pop(); st.Pop(); } while (st.Count > 0) { Console.Write(st.Pop() + " "); } Console.WriteLine(); if (q.Count > 0) { q.Dequeue(); } if (q.Count > 0) { </pre>

```

        Console.WriteLine(q.Count + "; " + q.Peek());
    }
    while (q.Count > 0)
    {
        Console.Write(q.Dequeue() + " ");
    }
}

```

Java

```

int n = 5738462;
Stack<Integer> st = new Stack<>();
Queue<Integer> q = new LinkedList<>();
while (n != 0) {
    int k = n % 10;
    if (k % 2 == 0) {
        st.push(k);
    } else {
        q.add(k);
    }
    n /= 10;
}
if (st.size() >= 2) {
    st.pop();
    st.pop();
}
while (st.size() > 0) {
    System.out.print(st.pop() + " ");
}
System.out.println();
if (q.size() > 0) {
    q.poll();
}
if (q.size() > 0) {
    System.out.println(q.size() + "; " + q.peek());
}
while (q.size() > 0) {
    System.out.print(q.poll() + " ");
}
}

```

24. В дадения фрагмент от код има редове, отбелязани с (1), (2) и (3). В полетата за отговори срещу съответния номер запишете какво ще изведе съответния ред.

Приема се, че всички необходими библиотеки са добавени.

C#

```

double[] arr = { 2.5, -5.9, 2.3, -3.1, 10, 0, -7.7, 9.2, -4.3, 6.8 };
for (int i = 0; i < arr.Length / 2; i++)
{
    arr[i] = Math.Ceiling(arr[i]);
}
for (int i = arr.Length / 2 + 1; i < arr.Length; i++)

```

```

{
    arr[i] = Math.Floor(arr[i]);
}
Console.WriteLine "[" + string.Join(", ", arr) + "];    //(1)
Array.Sort(arr);
Stack<long> stack = new Stack<long>();
Queue<long> queue = new Queue<long>();
for (int i = 0; i < arr.Length; i += 2)
{
    stack.Push((long)Math.Round(arr[i]));
}
Console.WriteLine "[" + string.Join(", ", stack) + "];    //(2)
for (int i = 1; i < arr.Length; i += 2)
{
    queue.Enqueue((long)Math.Round(arr[i]));
}
Console.WriteLine "[" + string.Join(", ", queue) + "];    //(3)

```

Java

```

double[] arr = {2.5, -5.9, 2.3, -3.1, 10, 0, -7.7, 9.2, -4.3, 6.8};
for (int i = 0; i < arr.length / 2; i++) {
    arr[i] = Math.ceil(arr[i]);
}
for (int i = arr.length / 2 + 1; i < arr.length; i++) {
    arr[i] = Math.floor(arr[i]);
}
System.out.println(Arrays.toString(arr));    //(1)
Arrays.sort(arr);
Stack<Long> stack = new Stack<>();
Queue<Long> queue = new LinkedList<>();
for (int i = 0; i < arr.length; i+=2) {
    stack.push(Math.round(arr[i]));
}
System.out.println(stack);    //(2)
for (int i = 1; i < arr.length; i+=2) {
    queue.add(Math.round(arr[i]));
}
System.out.println(queue);    //(3)

```

25. Дадената диаграма представя релационна база данни за турнир по футбол, съдържаща информация за отборите, участващи в турнира и проведените мачове.

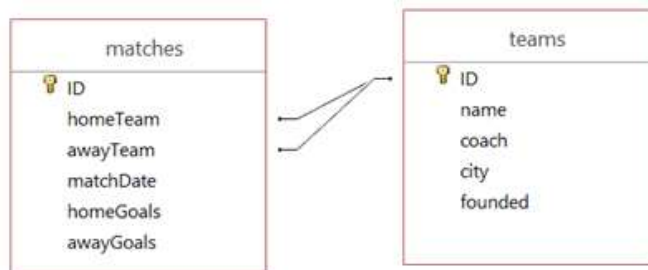
Дадената заявка извежда имената на отборите-домакини, датата на провеждане на мачовете и резултатите от тях.

```

SELECT th.name homeTeam, m.matchDate, m.homeGoals, m.awayGoals
FROM matches AS m
JOIN teams th ON m.homeTeam = th.id;

```

В полето за отговор запишете редактирана заявката, така че тя да извежда имената на отборите-домакини и отборите-гости, датата на провеждане и резултатите от мачовете, играни между отбори от един и същ град. Резултатът да е подреден в намаляващ ред по голова разлика (головете на домакините – головете на гостите), като при равенство се подрежда в намаляващ ред на головете на домакините.



МИНИСТЕРСТВО НА ОБРАЗОВАНИЕТО И НАУКАТА

ДЪРЖАВЕН ЗРЕЛОСТЕН ИЗПИТ ПО ИНФОРМАТИКА

22 май 2026 г.

ПРОФИЛИРАНА ПОДГОТОВКА ВАРИАНТ 2

ЧАСТ 2 (Време за работа: 150 минути)

Файловете с отговорите на задачите от 26. до 28. включително, качете в изпитната система, като спазите указанията в условието на задачата!

Внимание! Имената на работните файлове, които прикачвате в изпитната система НЕ трябва да съдържат текстове или символи, които могат да доведат до нарушаване на анонимността на изпитната Ви работа!

26. При работа с редици от числа е необходимо да се откриват подредици от последователни елементи, които отговарят на зададени условия.

Създайте приложение с име Zad26, което чете от стандартния вход едно естествено число N ($1 \leq N \leq 1000$) – броят на числата в редицата. От следващия ред чете редица от N естествени числа, разделени с точно един интервал. Никое от числата не надвишава 1 000 000.

Приложението трябва да намери и извежда най-дългата строго растяща подредица от **последователни елементи**.

Ако има две подредици с еднаква дължина, да се изведе тази, чиято сума на елементите е по-голяма.

Ако има две подредици с еднаква дължина и сума, да се изведе първата отляво надясно.

Приложението извежда на стандартния изход търсената информация в следния формат:

- На първия ред се извеждат две цели числа, разделени с интервал – броят на елементите на най-дългата растяща подредица и сума на тази редица.

- На втория ред се извеждат елементите на подредицата, разделени с интервал.

Пример:

Вход :

10

10 20 5 5 10 40 50 60 15 5

Изход:

5 165

5 10 40 50 60

Забележки:

Гарантирано е, че входните данни за програмата са коректни.

Приемат се и решения с графичен потребителски интерфейс (ГПИ), в които данните се въвеждат в подходящи контроли, а резултатът се извежда в етикет или нередактируемо текстово поле.

Прикачете в изпитната система архив с име **zad26.zip**, съдържащ файловете с Вашето решение.

27. Създайте релационна база от данни **Library**, която съхранява информация за управлението на книги в една библиотека..

А) Създайте в базата данни таблица **Books** със следните атрибути:

Име на колоната	Тип на данните	Ограничения
book_id	цяло число	Първичен ключ
title	текст до 45 символа	задължително съдържа стойност
author	текст на до 45 символа	задължително съдържа стойност
genre	текст до 45 символа	задължително съдържа стойност
publication_year	цяло число	задължително съдържа стойност

Б) Създайте в базата данни таблица **Readers** със следните атрибути:

Име на колоната	Тип на данните	Ограничения
reader_id	цяло число	Първичен ключ
reader_name	текст до 45 символа	задължително съдържа стойност
email	текст на до 45 символа	задължително съдържа стойност
membership_status	текст до 45 символа	задължително съдържа стойност

В) Създайте в базата данни таблица **Borrowings** със следните атрибути:

Име на колоната	Тип на данните	Ограничения
borrowing_id	цяло число	Първичен ключ
book_id	цяло число	задължително съдържа стойност, външен ключ към таблицата Books
reader_id	цяло число	задължително съдържа стойност, външен ключ към таблицата Readers
borrow_date	дата	
return_date	дата	

Г) Напишете заявка за добавяне на следните записи в таблицата **Books** (данните за таблицата са в ресурсен файл с име **resources.txt**):

Book_id	title	author	genre	Publication_year
1	Pod Igoto	Ivan Vazov	Roman	1894
2	Bai Ganio	Aleko Konstantinov	Satira	1895
3	Tiutyun	Dimitar Dimov	Roman	1951

4	Zhelezniyat svetilnik	Dimitar Talev	Istoricheski	1952
5	Istoriya slavyanobulgarska	Paisiy Hilendarski	Istoriya	1762
6	Prespanskite kambani	Dimitar Talev	Istoricheski	1960
7	Do Chicago i nazad	Aleko Konstantinov	Patepis	1894
8	Osadeni dushi	Dimitar Dimov	Drama	1945
9	Na iztok ot Zapada	Miroslav Penkov	Razkazi	2011
10	Patuvane kum sebe si	Blaga Dimitrova	Roman	1965

Д) Напишете заявка за добавяне на следните записи в таблицата **Readers** (данните за таблицата са в ресурсен файл с име **resources.txt**):

Reader_id	reader_name	email	membership_status
1	Borislav Petrov	borislav.p@email.com	Active
2	Silviya Nikolova	s.nikolova@email.com	Active
3	Georgi Todorov	georgi.t@email.com	Expired
4	Elitsa Vasileva	elitsa.v@email.com'	Active
5	Plamen Dimitrov	plamen.d@email.com	Suspended
6	Nadezhda Marinova	nadezhda.m@email.com	Active

Е) Напишете заявка за добавяне на следните записи в таблицата **Borrowings** (данните за таблицата са в ресурсен файл с име **resources.txt**):

borrowing_id	book_id	reader_id	borrow_date	return_date
1	1	1	2025-01-10	2025-01-20
2	3	2	2025-01-15	NULL
3	5	3	2025-02-01	2025-02-18
4	2	1	2025-02-05	NULL
5	4	4	2025-02-10	2025-02-22
6	8	2	2025-02-12	NULL
7	9	5	2025-02-18	2025-02-25
8	10	6	2025-02-20	NULL
9	7	4	2025-02-22	2025-03-01
10	6	1	2025-02-28	NULL

Ж) Напишете заявка, която връща заглавието и автора на книгите, публикувани между 1900 и 2000 г. и чиито заглавия започват с латинската буква „P“.

При вярно написана заявка се получава резултатът, показан на фигура 1:

title	author
Prespanskite kambani	Dimitar Talev
Patuvane kum sebe si	Blaga Dimitrova

Фигура 1

З) Напишете заявка, която връща имената и имейлите на всички читатели със статус **Active**, подредени по име в азбучен ред. При вярно написана заявка се получава резултатът, представен на фигура 2:

reader_name	email
Borislav Petrov	borislav.p@email.com
Elitsa Vasileva	elitsa.v@email.com
Nadezhda Marinova	nadezhda.m@email.com
Silviya Nikolova	s.nikolova@email.com

Фигура 2

И) Напишете заявка, която извежда заглавията на всички книги, които в момента са заети и все още не са върнати, заедно с имената на читателите, които ги държат, и датата на заемането. При вярно написана заявка се получава резултатът, представен на фигура 3.

title	reader_name	borrow_date
Tiutyun	Silviya Nikolova	2025-01-15
Bai Ganio	Borislav Petrov	2025-02-05
Osadeni dushi	Silviya Nikolova	2025-02-12
Patuvane kum sebe si	Nadezhda Marinova	2025-02-20
Prespanskite kambani	Borislav Petrov	2025-02-28

Фигура 3

К) Напишете заявка, която да извежда броя на книгите, които е заел всеки читател. При вярно написана заявка се получава резултатът, представен на фигура 4.

reader_name	borrowed books
Borislav Petrov	3
Elitsa Vasileva	2
Georgi Todorov	1
Nadezhda Marinova	1
Plamen Dimitrov	1
Silviya Nikolova	2

Фигура 4

Л) Напишете заявка, която извежда имената на читателите, които имат повече от 1 заемания, заедно с броя на заеманите от тях книги. При вярно написана заявка се получава резултатът, представен на фигура 5.

reader_name	borrowed books
Borislav Petrov	3
Elitsa Vasileva	2
Silviya Nikolova	2

Фигура 5

Прикачете в изпитната система файл с име *zad27.zip*, съдържащ създадената база от данни (при работа с *MS Access*) или заявките за създаването на база от данни (при работа със стандартен език за заявки) и написаните от Вас заявки към базата данни.

28. За нормалното функциониране на един хотел е необходимо да се поддържа информация за персонала и гостите.

Създайте приложение с име **Zad28**, което управлява информация за хората в хотел и извежда статистики по различни критерии.

Разработеното приложение трябва да съдържа класовете **Person**, **Worker**, **Guest** и **Manager**. Класовете трябва да имат описаните частни атрибути (полета) с осигурен публичен достъп-четене (за C# може да се използва идеологията „property“) и методи.

А) Абстрактен базов клас **Person** с полета:

- **firstName** (собствено име) – символен низ
- **lastName** (фамилно име) – символен низ
- **age** (възраст) – цяло положително число

Класът трябва да притежава следните методи:

- **Конструктор** с параметри, който задава стойности на всички характеристики;
Класът трябва да предоставя възможност за **валидиране** на полето **age**:
Възрастта трябва да е число по-голямо или равно на 18, иначе да се генерира изключение със съобщението: **"Age must be at least 18."**
- **Абстрактен метод Info()** (за C#) / **info()** (за Java), който извежда на стандартния

изход информацията за съответния служител;

Б) Клас **Worker** наследява **Person** и съдържа собствени полета:

- **experience** (стаж в години) – цяло число
- **salary** (заплата) – реално положително число

Класът трябва да притежава следните методи:

- **Конструктор** с параметри, който задава стойности на всички характеристики;
Класът трябва да предоставя възможност за валидиране на полето **salary**:
Заплатата трябва да е число по-голямо от 500 EUR, иначе да се генерира изключение със съобщението: **"Salary must be at least 500 EUR."**
- **Предефиниран метод Info** (за C#) и **info** (за Java) така, че да се извежда на стандартния изход информацията в следния формат:

Worker:<интервал><firstName><интервал><lastName>

Age: <интервал><Age>

Salary:<интервал><salary><интервал>EUR.

Years of experience: <интервал><yearsexperience>

Заплатата да бъде форматирана до втория знак след десетичния разделител.

В) Клас **Guest** наследява **Person** и съдържа собствени полета

- **room** (номер на стая) – цяло число
- **stay** (престой в дни) – цяло число

Класът трябва да притежава следните методи:

- **Конструктор** с параметри, който задава стойности на всички характеристики;
Класът трябва да предоставя възможност за валидиране на полето **stay**:
Престоят трябва да са число, по-голямо или равно на 1, иначе да се генерира изключение със съобщение: **"Stay duration must be greater than or equal to 1. "**
- **Предефиниран метод Info()** (за C#) и **info()** (за Java) така, че да се извежда на стандартния изход информацията в следния формат:

Guest: <интервал><firstName><интервал><lastName>

Age: <интервал><age>

Room: <интервал><room>

Stay: <интервал><stay>

Г) Клас **Manager** наследява **Worker** и съдържа собствени полета;

- **department** (отдел) – непразен низ
- **bonus** (бонус) – реално положително число

Класът трябва да притежава следните методи:

- **Конструктор** с параметри, който задава стойности на всички характеристики:

Класът трябва да предоставя възможност за валидиране на полетата **department** и **bonus**:

Отделът не трябва да е празен низ иначе да се генерира изключение със съобщението: **"Department cannot be empty."**

Бонусът трябва да е неотрицателно число, иначе да се генерира изключение със съобщението: **"Bonus cannot be negative."**

- **Предефиниран метод Info() (за C#) и info() (за Java)** така, че да се извежда на стандартния изход информацията в следния формат:

Manager:<интервал><firstName><интервал><lastName>

Age:<интервал><age>

Salary:<интервал><salary><интервал>EUR.

Years of experience:<интервал><experience>

Department: <интервал><department>

Bonus: <интервал><bonus><интервал>EUR.

Заплатата и бонусът да бъдат форматирани до втория знак след десетичния разделител.

Д) Приложението трябва да:

- Прочете от текстов файл **hotel.txt** данните за хората в хотела, като информацията за всеки човек е записана на един ред. **Всеки ред във файла е в един от следните 3 формата:**

Worker;firstName;lastName;age;experience;salary

Manager;firstName;lastName;age;experience;salary;department;bonus

Guest;firstName;lastName;age;room;stay

- Съхранява обектите в подходяща колекция.
- Прихваща и обработва изключения – при невалидни данни в ред на файла **hotel.txt** на стандартния изход се извежда съобщение с детайли за грешката (съобщенията от валидациите) и продължава обработката за останалите редове.

Приложението трябва да извежда на стандартният изход следната информация:

- Съобщението **„=== All people sorted ===“**, последвано на нов ред от списъка с информация за всички хора, сортирани по собствено име във възходящ ред, а при еднакво име – по фамилия, също във възходящ ред. След информацията се извежда един празен ред.

- Съобщението „**===The longest guest stay:<интервал><X><интервал>day(s) ===**“, където **<X>** е броят дни на най-дългия престой. След това на нови редове се извеждат име и фамилия, разделени с интервал, на гостите с най-дълъг престой. Ако във файла не се съдържа информация за гости се извежда съобщението: „**===No guests in the hotel!===**“
- Съобщението „**=== No managers without bonus.!==**“, ако всички мениджъри получават бонус. Ако има мениджъри без бонус да се изведе съобщението: „**===Managers without bonus:**
<интервал><count><интервал>===“, където **<count>** е броят на мениджърите без бонус.
- За извеждане на информацията за всеки служител се използва резултатът от метода **Info()** (за C#) / **info()** (за Java).

Приложени са 3 примерни файла. В зависимост от настройките на ОС, входът и изходът на реалните числа може да са с разделител точка или запетая. В дадените примерни файлове няма данни с дробна част, но ако тествате с ваши примери, трябва да внимавате с разделителя. Няма нужда да преправяте изхода, ако е с разделител запетая.

Изход с използване на данните от ресурсния файл с име hotel2.txt:
=== All people sorted ===

Guest: Anna Petrova
 Age: 27
 Room: 101
 Stay: 15
 Worker: Elena Georgieva
 Age: 45
 Salary: 1160,00 EUR.
 Years of experience: 20
 Guest: Elena Todorova
 Age: 28
 Room: 402
 Stay: 14
 Worker: Georgi Dimitrov
 Age: 45
 Salary: 3500,00 EUR.
 Years of experience: 20
 Guest: Ivan Georgiev

Age: 52
Room: 305
Stay: 2
Worker: Ivan Petrov
Age: 34
Salary: 1100,00 EUR.
Years of experience: 10
Guest: Maria Dimitrova
Age: 44
Room: 210
Stay: 15
Worker: Maria Ivanova
Age: 41
Salary: 1400,00 EUR.
Years of experience: 15
Worker: Petar Stoyanov
Age: 50
Salary: 4200,00 EUR.
Years of experience: 25
Guest: Stefan Hristov
Age: 36
Room: 115
Stay: 1
Worker: Stefan Kolev
Age: 38
Salary: 900,00 EUR.
Years of experience: 8

===The longest guest stay: 15 day(s) ===
Anna Petrova
Maria Dimitrova
=== No managers without bonus.!==

*Прикачете в изпитната система архив с име **zad28.zip**, съдържащ файловете с Вашето решение.*