

ДЪРЖАВЕН ЗРЕЛОСТЕН ИЗПИТ ПО

ИНФОРМАТИКА

20 май 2022 г.

ПРОФИЛИРАНА ПОДГОТОВКА

ВАРИАНТ 1

ЧАСТ 1 (Време за работа: 90 минути)

Отговорите на задачите от 1. до 16. включително отбелязвайте в листа за отговори!

1. Каква ще е стойността на променливата `count` след изпълнението на следния програмен фрагмент:

```
int n = 42, count = 0;
while (n > 1) {
    count++;
    if (n % 2 == 0)
        n = n / 2;
    else
        n = 3 * n + 1;
}
```

A) 5;

Б) 8;

В) 10;

Г) 41.

2. Нека променливите `s1`, `s2` и `s3` са инициализирани съответно с низовете "roses", "are" и "red". Оценката на кой от следните изрази е истина?

C#	Java
A) <code>string.Compare(s1, s2) < 0</code>	A) <code>s1.compareTo(s2) < 0</code>
Б) <code>string.Compare(s2, s3) != 0</code>	Б) <code>s2.compareTo(s3) != 0</code>
В) <code>string.Compare(s3, s1) > 0</code>	В) <code>s3.compareTo(s1) > 0</code>
Г) <code>string.Compare(s1, s3) == 0</code>	Г) <code>s1.compareTo(s3) == 0</code>

3. Нека са дадени следните дефиниции:

```
class Test
{
    private int x = 0;
    protected void setX(int value)
    {
        x = value;
    }
}
```

Възможно ли е наследниците на Test да извикват метода setX на Test?

- A) Не, защото методът достъпва private променлива.
- Б) Не, защото setX е protected.
- В) Не, защото x е константа със стойност 0 и не може да бъде променяна.
- Г) Да.

4. След края на try-catch конструкция може да се сложи специален блок, кодът в който се изпълнява независимо дали клаузата try е изцяло завършена, или е било хвърлено изключение и се е преминало към изпълнение на клауза catch. Този блок често се използва за почистване на ресурси. Коя е запазената дума, с която се означава той?

- A) finally
- Б) destruct
- В) cleanup
- Г) end

5. Какво ще се изведе в конзолата след изпълнение на програмата:

```
C#
static void Main(string[] args)
{
    counter(3);
}
private static void counter (int n)
{
    if (n == 0)
        return;
    Console.Write(n);
    counter(n - 1);
}
```

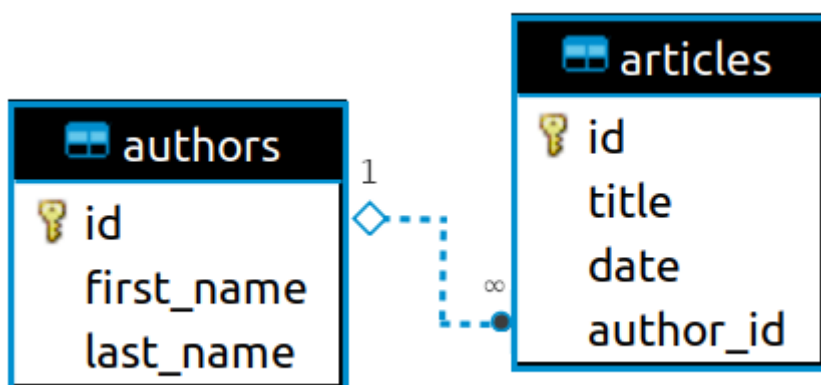
```

Java
public static void main(String[] args) {
    counter(3);
}
private static void counter(int n){
    if (n == 0)
        return;
    System.out.print(n);
    counter(n - 1);
}

```

- A) 3210; Б) 1230; В) 123; Г) 321.

6. Дадена е следната диаграма на релационна база от данни:



Кое от следните твърдения е вярно?

- А) Колоната id в authors образува първичен ключ, колоната id в articles образува първичен ключ, а колоната author_id образува външен ключ, рефериращ колоната id в authors.
- Б) Колоната id в articles образува първичен ключ, а колоната id в authors образува едновременно първичен ключ и външен ключ, рефериращ колоната author_id.
- В) Колоната author_id образува първичен ключ, колоната id в authors образува външен ключ и колоната id в articles образува външен ключ.
- Г) Колоните id в authors, id в articles и author_id образуват първични ключове.

7. Коя последователност от команди и в какъв ред трябва да се изпълни, за да може да се прочете информацията, записана в текстов файл?

1. създаване на променлива от тип текстов файл със създадения текстов файл;
2. прочитане на данни от текстовия файл;

3. отваряне на файла за четене;

4. отваряне на файла за запис.

A) 1, 2, 3;

Б) 1, 3, 2;

В) 1, 4, 3;

Г) 3, 2, 4.

8. Какво ще се отпечата на стандартния изход след изпълнение на програмния фрагмент?

<pre>C# static void Main(string[] args) { Stack<int> numbers = new Stack<int>(); numbers.Push(5); numbers.Push(7); numbers.Push(2); numbers.Pop(); numbers.Push(3); numbers.Push(4); numbers.Pop(); Console.Write(numbers.Pop() + " "); Console.Write(numbers.Peek() + " "); Console.Write(numbers.Pop() + " "); Console.WriteLine(numbers.Pop()); }</pre>
<pre>Java public static void main(String[] args) { Stack<Integer> numbers = new Stack<>(); numbers.push(5); numbers.push(7); numbers.push(2); numbers.pop(); numbers.push(3); numbers.push(4); numbers.pop(); System.out.print(numbers.pop() + " "); System.out.print(numbers.peek() + " "); System.out.print(numbers.pop() + " "); System.out.println(numbers.pop()); }</pre>

A) 4 3 3 2;

Б) 3 7 7 5;

В) 5 7 7 3;

Г) 5 7 2 3.

9. Дадена е следната диаграма на релационна база от данни:



Колко общо са външните ключове в представените таблици?

- А) един; Б) два; В) три; Г) четири.

10. Кой етап от жизнения цикъл на една информационна система е най-продължителният?

- А) експлоатация; Б) разработка; В) тестване; Г) синтезиране.

11. За коя от структурите от данни добавянето на елемент на посоченото място е с най-ниска сложност по време?

- А) В началото на свързан списък.
Б) Във върха на стек.
В) В края на опашка.
Г) Трите посочени по-горе операции са с еднаква сложност по време.

12. Даден е фрагмент от програма, обработващ елементите на едномерен целочислен масив A с индекси от 0 до 9.

C#	Java
<pre>int[] A = {3, 5, 6, 2, 1, 6, 3, 1, 8, 3}; int s, k, t; s = 0; for (k = 1; k < 10; ++k) { if (A[k] < A[k - 1]) { t = A[k]; A[k] = A[k - 1]; A[k - 1] = t; s = s + k; } } Console.WriteLine(s);</pre>	<pre>int[] A = {3, 5, 6, 2, 1, 6, 3, 1, 8, 3}; int s, k, t; s = 0; for (k = 1; k < 10; ++k) { if (A[k] < A[k - 1]) { t = A[k]; A[k] = A[k - 1]; A[k - 1] = t; s = s + k; } } System.out.println(s);</pre>

Стойността на променливата s, след изпълнението на фрагмента, е:

- A) 28; Б) 30; В) 31; Г) 29.

13. Следните твърдения се описват с връзки „има“ (has-a) и „е“ (is-a):

Кучетата са животни.

Пудел е порода куче.

Шаро е пудел.

Посочете коя от следните конструкции в език за обектно-ориентирано програмиране се доближава най-точно до тях?

А) Дефиниран е клас с име Dog („Куче“). От него са създадени два обекта с имена Poodle („Пудел“) и Sharo („Шаро“).

Б) Дефиниран е клас с име Dog („Куче“). Той има за наследник клас с име Poodle („Пудел“). Sharo („Шаро“) е име на обект от тип Poodle.

В) Дефиниран е клас с име Animal („Животно“). От него са създадени обекти с имена Dog („Куче“), Poodle („Пудел“) и Sharo („Шаро“).

Г) Дефиниран е клас с име Dog („Куче“). Той има наследник с име Poodle („Пудел“), който на свой ред има за наследник клас с име Sharo („Шаро“).

14. Отбележете отговора, при който на стандартния изход ще се изведе следният текст: "Harry Potter" - J. K. Rowling

C#

- A) Console.WriteLine("Harry Potter" - J. K. Rowling);
- Б) Console.WriteLine("Harry Potter\" - J. K. Rowling");
- В) Console.WriteLine("\Harry Potter\" - J. K. Rowling");
- Г) Console.WriteLine("Harry Potter - J. K. Rowling").

Java

- A) System.out.println("Harry Potter" - J. K. Rowling);
- Б) System.out.println("Harry Potter\" - J. K. Rowling");
- В) System.out.println("\Harry Potter\" - J. K. Rowling");
- Г) System.out.println("Harry Potter - J. K. Rowling").

15. Дадена е структура на таблицата Persons:

Име на колона	Тип данни	Ограничения
Id	int	Primary key, Auto Increment
FirstName	varchar(50)	Not Null
LastName	varchar(50)	-
Age	int	Age >= 18

Отбележете отговора, съдържащ програмният ред, чрез който може да се изберат всички записи от таблица "Persons", където "FirstName" е "Ivan" и "LastName" е "Petrov"?

- A) SELECT FirstName = 'Ivan', LastName = 'Petrov' FROM Persons
- Б) SELECT * FROM Persons WHERE FirstName <> 'Ivan' AND LastName <> 'Petrov'
- В) SELECT * FROM Persons WHERE FirstName = 'Ivan' AND LastName = 'Petrov'
- Г) SELECT Persons WHERE FirstName = 'Ivan' AND LastName = 'Petrov'

16. Каква ще е числовата стойност на променливата **n** след изпълнението на следния програмен фрагмент?

C#	Java
<pre>int s = 3, n = 1; int[] a = {1, 7, 13}; while (s < 80) { s = s + a[2]; n = n * 2; }</pre>	<pre>int s = 3, n = 1; int[] a = {1, 7, 13}; while (s < 80) { s = s + a[2]; n = n * 2; }</pre>

A) 6; Б) 32; В) 64; Г) 81.

Отговорите на задачите от 17. до 24. вкл. запишете в листите за отговори!

17. Каква конструкция трябва да се постави на мястото на многоточието, за да се създаде обект с име „hash“, представящ абстрактната структура от данни хеш-таблица с ключове числа и стойности символни низове, така че да е валиден кодът?

C#
<pre>..... hash.Add(3, "Abc"); int size = hash.Count;</pre>
Java
<pre>..... hash.put(3, "Abc"); int size = hash.size();</pre>

18. Какъв ще е върнатият резултат след извикването **function(99)** на следния метод:

<pre>static int function(int n) { if (n > 100) return n - 10; return function(function(n + 11)); }</pre>

19. На кой от редовете на следния програмен фрагмент е допусната грешка?

```
1  int n = 7;
2  double x = 2.3, y;
3  y = x + n;
4  if (5 > y)
5      n - 1 = n;
6  else
7      n = 15 / n;
8  while (n > 0) {
9      n = n + 1;
10     x = y * 2;
11     y = y;
12 }
```

20. Дадена е таблицата **brands** със следните данни:

id	name	market_share
1	BestCam	0.35
2	Affordables	0.25
3	Camera Max	0.19

Дадена е и таблицата **cameras** със следните данни:

id	model	brand_id	mpix	zoom	price
1	CM51	3	18	25	1000
2	A234	1	24	40	2000
3	Z1	1	16	22	800
4	J11	2	8	5	180
5	A345	1	15	18	1250
6	U25	2	14	21	800

Запишете в листа за отговори (1) колко ще бъдат редовете и (2) колко ще бъдат колоните в резултата от изпълнението на следната заявка?

```
SELECT *
FROM cameras
JOIN brands ON brand_id = brands.id
WHERE market_share > 0.2 AND zoom >= 20;
```

21. Какво ще изведе на стандартния изход дадената по-долу програма? В листите за отговори напишете точно това, което ще се изведе от програмата, без да слагате допълнителни символи или коментари.

```
C#
using System;

class A
{
    private static int m_counter = 0;

    public A() { m_counter++; }

    public static int getCounter() {
        return m_counter;
    }
}

class Program
{
    static void Main(string[] args)
    {
        A obj = new A();
        Console.Write(A.getCounter() + ";");

        for (int i = 0; i < 3; i++) {
            A temp = new A();
        }
        Console.Write(A.getCounter() + ";");
        func(obj);
    }

    public static void func(A param) {
        Console.Write(A.getCounter() + ";");
    }
}
}
```

Java

```
class A {
    private static int m_counter = 0;

    public A() { m_counter++; }

    public static int getCounter() {
        return m_counter;
    }
}

class Application {
    public static void main(String[] args) {
        A obj = new A();
        System.out.print(A.getCounter() + "");

        for(int i = 0; i < 3; i++) {
            A temp = new A();
        }
        System.out.print(A.getCounter() + "");
        func(obj);
    }

    public static void func(A param) {
        System.out.print(A.getCounter() + "");
    }
}
```

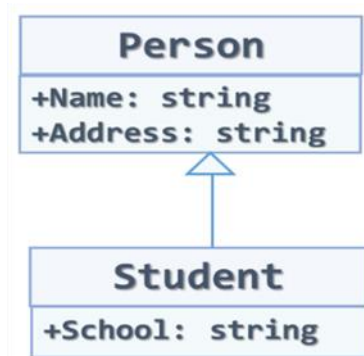
22. Какъв текст ще се изведе на стандартния изход след последователното въвеждане на символите '3' и 'x' на отделни редове при изпълнението на следната програма

<pre>C# using System; class Program { static void Main(string[] args) { Console.WriteLine("Въведете две цели числа"); try { int a = Convert.ToInt32(Console.ReadLine()); int b = Convert.ToInt32(Console.ReadLine()); if (a < b) Console.WriteLine("Първото число е по-малко"); else Console.WriteLine("Първото число НЕ е по-малко"); } catch (FormatException) { Console.WriteLine("Некоректно въведено число"); } } }</pre>
<pre>Java import java.util.Scanner; class Program { public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.println("Въведете две цели числа"); try { int a = Integer.parseInt(scanner.nextLine()); int b = Integer.parseInt(scanner.nextLine()); if (a < b) System.out.println("Първото число е по-малко"); else System.out.println("Първото число НЕ е по-малко"); } catch (NumberFormatException e) { System.out.println("Некоректно въведено число"); } } }</pre>

23. В дадената по-долу UML диаграма са дадени класовете Person и Student.

- 1) Кои са полетата на всеки един от класовете?
- 2) Каква е връзката между класовете?

(В листите за срещу (1) и (2) запишете съответните думи или изрази.)



24. Разгледайте дадения по-долу фрагмент. Три места в него са оставени празни. Те са обозначени и номерирани с коментар, съдържащ число (например /* 1 */). Какви конструкции трябва да се добавят на тяхно място, за да може кодът да реализира валидно двоично търсене в масив, който е сортиран възходящо?

(В листите за отговори срещу (1), (2) и (3) запишете съответните думи или изрази.)

```
C#
int BinSearch(int[] arr, int key){
    int left = 0, right = arr.Length;
    while (/* 1 */){
        int mid = /* 2 */;
        if (arr[mid] == key)
            return mid;
        if (/* 3 */)
            left = mid + 1;
        else
            right = mid;
    }
    return -1;
}
```

Java

```
int binSearch(int[] arr, int key){
    int left = 0, right = arr.length;
    while (/* 1 */){
        int mid = /* 2 */;
        if (arr[mid] == key)
            return mid;
        if (/* 3 */)
            left = mid + 1;
        else
            right = mid;
    }
    return -1;
}
```

МИНИСТЕРСТВО НА ОБРАЗОВАНИЕТО И НАУКАТА

ДЪРЖАВЕН ЗРЕЛОСТЕН ИЗПИТ ПО

ИНФОРМАТИКА

20 май 2022 г.

ПРОФИЛИРАНА ПОДГОТОВКА

ВАРИАНТ 1

ЧАСТ 2 (Време за работа: 150 минути)

Файловете с отговорите на задачите от 25. до 28. включително запишете 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!

25. Да се реализира програма, която въвежда от клавиатурата брой елементи в редица и непразна редица от толкова на брой цели положителни числа. На стандартния изход да се извежда информация колко пъти се среща всяко от числата в редицата. В изведената информация за броя на срещания НЕ трябва има повторения. Редът на извеждане на числата няма значение. Първото въведено число е брой на числата в редицата. В програмата да бъде направена необходимата валидация и прихващане на изключения.

Пример:

Примерен вход	Примерен изход
6	число: 1, брой: 3
1	число: 5, брой: 2
5	число: 2, брой: 1
5	
1	
2	
1	

*Запазете решението във файл с име **ВАШИЯТ ВХОДЯЩ НОМЕР_zad25./съответното разширение/** и го прикачете в изпитната система.*

26. Реализирайте описаните по-долу класове.

А) Напишете клас `Item`, представящ стока в магазин. Той има характеристики `description` (описание; символен низ) и `price` (цена; реално число). Обектите на класа да са `immutable` (характеристиките им не могат да се променят след конструирането на обекта). Класът да има конструктор, който задава стойности на характеристиките. Ако му се подадат некоректни данни, да се хвърли изключение. Да се гарантира, че цената е положително число, а описанието е непразен низ.

Б) Стоките трябва да може да се сравняват. Една стока А е по-напред в наредбата от друга стока В, ако описанието на А е лексикографски по-малко от това на В или ако описанията съвпадат, но цената на А е по-малка от тази на В. Имплементирайте подходящ интерфейс (`Comparable` в Java, `IComparable` в C#) в `Item` и предефинирайте съответния метод за сравнение.

В) Стоките трябва да може да се конвертират до символен низ във формат „описание (цена)“ (например “Тетрадка (3,50)”). За целта предефинирайте метода `toString` (Java) / `ToString` (C#) за `Item`.

Г) Напишете клас `ItemList`, който пази списък от стоки. Вътрешно класът да използва `ArrayList` (Java) или `List` (C#). Стоките в списъка трябва да се поддържат сортирани в нарастващ ред и да не се допускат повторения. Класът да има следните методи:

* `size` – връща броя на елементите в списъка. В C# може вместо това да напишете свойство (property) `Count`.

* `get` – връща елемент съхранен на даден индекс. Ако се подаде невалиден индекс, да се хвърля изключение.

* `add` – добавя стока в списъка, вмъква я на подходяща позиция, така че списъкът да остане сортиран. Ако стоката вече се среща в списъка, да се хвърли изключение.

Д) Напишете текстови клас `ItemList`, в който се въвежда от потребителя цяло положително число N. След това да се въвежда информация за N на брой стоки и ги запазва в `ItemList` обект. Накрая да се извежда съдържанието на списъка на екрана. Програмата да прихваща възможните изключения, да уведомява потребителя за тях и да предприема подходящи действия.

Прикачете в изпитната система архив с име ВАШИЯТ ВХОДЯЩ НОМЕР_zad26.zip (например 123456789_zad26.zip), съдържащ файловете с Вашите решения.

27. Създайте База данни на онлайн магазин за техника, съдържаща таблица laptops със следните атрибути:

ID – AutoNumber, PK;

Марка – Text;

Модел – Text;

Наличност – Number;

Цена – Currency;

Напишете:

1. Заявка за създаване на таблицата laptops.
2. Пет заявки, които зареждат в таблицата следните кортежи:

ID	Марка	Модел	Наличност	Цена
1	Laptop1	L29KAS	10	1100
2	Laptop2	15FDR7	14	1350
3	Laptop1	L29GTA	12	1500
4	Laptop1	L29DFT	8	1499
5	Laptop2	15FDM5	11	1600

3. Заявка, която да изтрива данните за модел лаптоп – 15FDR7.
4. Заявка, която извежда общата сума (наличност*цена), с добавена ДДС стойност (20%) към цената, за всички модели.
5. Заявка, която намира броят на всички наличности, от марката лаптопи – Laptop1.

Прикачете в изпитната система архив **ВАШИЯТ ВХОДЯЩ НОМЕР_zad27.zip**, (например 123456789_zad27.zip).

28. А) Напишете метод, който получава като аргументи обект, представляващ списък (обект от клас List или LinkedList в C# / ArrayList или LinkedList в Java) с елементи цели числа, и цяло число K. Методът трябва да премахне от списъка всички елементи, за които сумата от цифрите им се дели без остатък на K. Резултатът да се получава в същия списък.

Б) Напишете метод, който получава като аргумент обект, представляващ списък (обект от клас List или LinkedList в C# / ArrayList или LinkedList в Java) с елементи цели числа, и го подрежда в нарастващ ред на сумата от цифрите на тези числа.

В) Напишете метод, който получава като аргумент символен низ - име на текстов файл. Методът трябва да прочете от файла с това име цели числа, разделени с нови редове, и да ги запише в списък, който да върне като резултат. Погрижете се да прихванете и обработите възможните изключения при работа с файла.

Г) Напишете програма, която чрез използване на описаните по-горе методи, прочита от стандартния вход име на файл и цяло число K . След това прочита от текстов файл с това име цели числа в списък (`List` или `LinkedList` в `C#` / `ArrayList` или `LinkedList` в `Java`) и премахва от списъка всички елементи, чиято сума от цифрите се дели на K . Да се изведе на стандартен изход полученият като резултат списък, подреден в нарастващ ред на сумата от цифрите на числата в него.

Примерен вход от файла
21 32 44 42 11 100 101 12
Изход на екрана при $K = 3$
100 11 101 32 44
Примерен вход от файла
21 32 44 42 11 100 101 12
Изход на екрана при $K = 4$
100 11 101 21 12 32 42

Прикачете в изпитната система архив с име **ВАШИЯТ ВХОДЯЩ НОМЕР_zad28.zip** (например **123456789_zad28.zip**), съдържащ файловете с Вашите решения от проекта.

МИНИСТЕРСТВО НА ОБРАЗОВАНИЕТО И НАУКАТА

ДЪРЖАВЕН ЗРЕЛОСТЕН ИЗПИТ ПО

ИНФОРМАТИКА

20 май 2022 г.

ПРОФИЛИРАНА ПОДГОТОВКА

ВАРИАНТ 1

Задача от 1. до 16. Ключ с верните отговори

Въпрос №	Верен отговор	Брой точки
1.	Б	1
2.	Б	1
3.	Г	1
4.	А	1
5.	Г	1
6.	А	1
7.	Б	1
8.	Б	1
9.	А	1
10.	А	1
11.	Г	1
12.	Г	1
13.	Б	1
14.	В	1
15.	В	1
16.	В	1

Задача 17. – 3 точки

Верен отговор	<p>C#</p> <p>Вариант 1: Hashtable hash = new Hashtable();</p> <p>Вариант 2: Dictionary<int, string> hash = new Dictionary<int, string>();</p> <p>Java</p> <p>Вариант 1: HashMap<Integer, String> hash = new HashMap<>();</p> <p>Вариант 2:</p>
---------------	--

	<p>Остаряла, но валидна алтернатива: <code>Hashtable<Integer, String> hash = new Hashtable<>();</code></p> <p>Вариант 3: <code>Map<Integer, String> hash = new HashMap<>();</code></p> <p>Вариант 4: <code>Map<Integer, String> hash = new HashMap<Integer, String>();</code></p> <p>Вариант 5: <code>HashMap<Integer, String> hash = new HashMap<Integer, String>();</code></p>
--	--

Задача 18. – 3 точки

Верен отговор	91
---------------	----

Задача 19. – 3 точки

Верен отговор	5
---------------	---

Задача 20. – 3 точки

Верен отговор	3 реда, 9 колони
---------------	------------------

Задача 21. – 3 точки

Верен отговор	1;4;4;
---------------	--------

Задача 22. – 3 точки

Верен отговор	Некоректно въведено число
---------------	---------------------------

Задача 23. – 3 точки

Верен отговор	<p>Класът Person е с полета: Name и Address</p> <p>Класът Student е с поле: School</p> <p>Класът Student наследява класа Person</p>
---------------	---

Задача 24. – 3 точки

Верен отговор	<pre>/* 1 */ - left < right или right > left /* 2 */ - left + (right - left) / 2 или (left + right) / 2 /* 3 */ - arr[mid] < key или key > arr[mid]</pre>
---------------	---

Забележка: Признават се и всички други верни предложени решения.

Задача 25. – 10 точки

Примерно решение

C#

```
using System;

class Problem25{
    static int[] readArray() {
        Console.WriteLine("Въведете брой елементи");
        int n;
        do {
            n = Convert.ToInt32(Console.ReadLine());
            if (n < 1)
                Console.WriteLine("Моля, въведете положително число!");
        } while (n < 1);
        int[] result = new int[n];
        for(int i = 0; i < n; i++) {
            do {
                Console.WriteLine("Моля, въведете елемент " + (i +
1));
                result[i] = Convert.ToInt32(Console.ReadLine());
                if (result[i] < 1)
                    Console.WriteLine("Моля, въведете положително
число!");
            }
        }
    }
}
```

```

        } while (result[i] < 1);
    }
    return result;
}
static void Main(string[] args) {
    try {
        int[] array = readArray();
        for(int i = 0; i < array.Length; i++) {
            // това първо срещане ли е?
            int j = 0;
            while (j < array.Length && array[i] != array[j])
                j++;
            if (j == i) {
                int count = 1;
                for(j++; j < array.Length; j++)
                    if (array[i] == array[j])
                        count++;
                Console.WriteLine("Число: " + array[i] + ", брой
срещания: " + count);
            }
        }

    } catch (FormatException) {
        Console.WriteLine("Некоректно въведено число");
    }
}
}

```

Java

```

import java.util.Scanner;
public class Problem25 {
    static int[] readArray(Scanner scanner) {

```

```

System.out.println("Въведете брой елементи");
int n;
do {
    n = Integer.parseInt(scanner.nextLine());
    if (n < 1)
        System.out.println("Моля, въведете положително
число!");
} while (n < 1);
int[] result = new int[n];
for(int i = 0; i < n; i++) {
    do {
        System.out.println("Моля, въведете елемент " +
(i + 1));
        result[i] = Integer.parseInt(scanner.nextLine());
        if (result[i] < 1)
            System.out.println("Моля, въведете положително
число!");
    } while (result[i] < 1);
}
return result;
}
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    try {
        int[] array = readArray(scanner);
        for(int i = 0; i < array.length; i++) {
            // това първо срещане ли е?
            int j = 0;
            while (j < array.length && array[i] != array[j])
                j++;
            if (j == i) {
                int count = 1;
                for(j++; j < array.length; j++)
                    if (array[i] == array[j])
                        count++;
                System.out.println("Число: " + array[i] + ",
брой срещания: " + count);
            }
        }
    } catch (NumberFormatException e) {
        System.out.println("Некоректно въведено число");
    }
}
}

```

Задача 26 – 15 точки

Примерно решение

C#

```
using System;
using System.Collections.Generic;

class Item : IComparable<Item> {
    private string description;
    private double price;

    public Item(string description, double price)
    {
        if (!IsValid(description, price))
            throw new ArgumentException("Incorrect argument");

        this.description = description;
        this.price = price;
    }

    public string getDescription() { return description; }
    public double getPrice() { return price; }

    bool IsValid(string description, double price)
    {
        return description != null &&
            description.Length > 0 &&
            price > 0;
    }

    public int CompareTo(Item o)
    {
        int compare = description.CompareTo(o.description);
        return (compare == 0) ? price.CompareTo(o.price) :
compare;
    }

    public override string ToString()
    {
        return description + " (" + price + ")";
    }
}

class ItemList
{
    private List<Item> contents = new List<Item>();

    public void Add(Item item)
    {
        int i = 0;
```



```

    int comp = -1;

    for(; i < contents.Count; ++i) {
        comp = contents[i].CompareTo(item);
        if(comp >= 0) {
            break;
        }
    }

    if (comp == 0) {
        throw new Exception("Item already exists in list");
    }
    else {
        contents.Insert(i, item);
    }
}

public int Count
{
    get
    {
        return contents.Count;
    }
}

public Item get(int index)
{
    return contents[index];
}
}

class ItemTest
{
    static void Main(string[] args)
    {
        try {

            ItemList items = new ItemList();

            Console.Write("How many items do you want to enter:
");

            int size = Convert.ToInt32(Console.ReadLine());

            for (; size > 0; --size)
            {
                Console.Write("Enter description: ");
                string description = Console.ReadLine();
                Console.Write("Enter price: ");
                double price =
Convert.ToDouble(Console.ReadLine());

                items.Add(new Item(description, price));
            }
        }
    }
}

```

```

    }

    for (int i = 0; i < items.Count; ++i)
    {
        Console.WriteLine(items.get(i).ToString());
    }
}
catch (Exception e)
{
    Console.WriteLine("An error has occurred: " +
e.Message);
}
}
}

```

Java

```

public class Item implements Comparable<Item> {
    private String description;
    private double price;

    public Item(String description, double price) {
        if(!isValid(description, price))
            throw new IllegalArgumentException("Incorrect
argument");

        this.description = description;
        this.price = price;
    }

    public String getNumber() { return description; }
    public double getPrice() { return price; }

    boolean isValid(String description, double price)
    {
        return description != null &&
            description.length() > 0 &&
            price > 0;
    }

    @Override
    public int compareTo(Item o) {
        int compare = description.compareTo(o.description);
        return (compare == 0) ? Double.compare(price,o.price) :
compare;
    }

    @Override
    public String toString() {
        return description + " (" + price + ")";
    }
}

```

```

    }
}

import java.util.ArrayList;

public class ItemList {
    private ArrayList<Item> contents = new ArrayList<>();

    public void add(Item item) throws Exception{
        int i = 0;
        int comp = -1;

        for(; i < contents.size(); ++i) {
            comp = contents.get(i).compareTo(item);
            if(comp >= 0) {
                break;
            }
        }

        if(comp == 0) {
            throw new Exception("Item already exists in list");
        }
        else {
            contents.add(i,item);
        }
    }

    public int size() {
        return contents.size();
    }

    public Item get(int index) {
        return contents.get(index);
    }
}

import java.util.ArrayList;
import java.util.Scanner;

public class ItemTest {
    public static void main(String[] args) {

        try (Scanner input = new Scanner(System.in)) {

            ItemList items = new ItemList();

            System.out.print("How many items do you want to enter:
");

            int size = Integer.parseInt(input.nextLine());

            for (; size > 0; --size) {

```

```

        System.out.print("Enter description: ");
        String description = input.nextLine();
        System.out.print("Enter price: ");
        double price =
Double.parseDouble(input.nextLine());

        items.add(new Item(description, price));
    }

    for (int i = 0; i < items.size(); ++i) {
        System.out.println(items.get(i).toString());
    }
} catch (Exception e) {
    System.out.println("An error has occurred: " +
e.getMessage());
}
}
}

```

Задача 27 – 15 точки

Примерно решение в ACCESS:

1. CREATE TABLE laptops (
 ID AUTOINCREMENT PRIMARY KEY,
 Марка varchar(255),
 Модел varchar(255),
 Наличност int,
 Цена currency
);

2. Заявки за въвеждане на кортежите:
 - a) INSERT INTO laptops (Марка, Модел, Наличност, Цена)
 VALUES ('Laptop 1', 'L29KAS', 10, 1100);
 - b) INSERT INTO laptops (Марка, Модел, Наличност, Цена)
 VALUES ('Laptop 2', '15FDR7', 14, 1350);
 - c) INSERT INTO laptops (Марка, Модел, Наличност, Цена)
 VALUES ('Laptop 1', 'L29GTA', 12, 1500);
 - d) INSERT INTO laptops (Марка, Модел, Наличност, Цена)
 VALUES ('Laptop 1', 'L29DFT', 8, 1499);
 - e) INSERT INTO laptops (Марка, Модел, Наличност, Цена)
 VALUES ('Laptop 2', '15FDM5', 11, 1600);

```
3.DELETE FROM laptops
WHERE laptops.Модел='15FDR7';
```

```
4.SELECT laptops.Модел, laptops.Наличност*laptops.Цена*1.2 AS Общо
FROM laptops;
```

5. Текстът SumOfНаличност е потребителски. Скобите при командата WHERE може да се пропуснат.

```
SELECT laptops.Марка, Sum(laptops.Наличност) AS SumOfНаличност
FROM laptops
WHERE (((laptops.Марка)= 'Laptop 1'))
GROUP BY laptops.Марка;
```

Примерно решение в SQL Server

```
1. CREATE TABLE laptops (
  ID int IDENTITY PRIMARY KEY,
  Марка nvarchar(255),
  Модел nvarchar(255),
  Наличност int,
  Цена smallmoney
);
```

Останалите заявки са аналогични!

Задача 28 – 20 точки

Примерно решение

C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.IO;

namespace _28
{
    class Problem28
    {
        static LinkedList<int> ReadList(string fileName)
        {
            StreamReader file;
            try
            {
                file = File.OpenText(fileName);
            }
            catch (IOException ex)
            {
                Console.WriteLine("Проблем с отваряне на {0} :
{1}.", fileName, ex.Message);
                return null;
            }

            LinkedList<int> list = new LinkedList<int>();
            try
            {
```

```
        for (string line = file.ReadLine(); line != null;
line = file.ReadLine())
    {
        if (line.Length > 0)
        {
            int number = int.Parse(line);
            list.AddLast(number);
        }
    }
}
catch (FormatException)
{
    Console.WriteLine("Невалиден формат на входа.");
}
catch (OverflowException)
{
    Console.WriteLine("Невалидно число.");
}

file.Close();
return list;
}

static int SumDigits(int num)
{
    num = Math.Abs(num);
    int sum = 0;
    while (num != 0)
    {
        sum += num % 10;
        num /= 10;
    }
}
```

```

    }
    return sum;
}

static void FilterList(LinkedList<int> list, int k)
{
    if (list == null || list.Count == 0 || k == 0)
        return;
    k = Math.Abs(k);

    while (list.Count > 0 && SumDigits(list.First.Value) %
k == 0)
        list.RemoveFirst();
    if (list.Count == 0)
        return;

    LinkedListNode<int> node = list.First;
    while(node.Next != null)
    {
        if (SumDigits(node.Next.Value) % k == 0)
            list.Remove(node.Next);
        else
            node = node.Next;
    }
}

class Comparer : IComparer<int>
{
    public int Compare(int x, int y)
    {
        return SumDigits(x) - SumDigits(y);
    }
}

```



```

    }
}

    static void Insert(LinkedList<int> list,
LinkedListNode<int> node)
    {
        if (list.Count == 0 || SumDigits(list.First.Value) >
SumDigits(node.Value))
            list.AddFirst(node);
        else {
            LinkedListNode<int> after = list.First;
            while (after.Next != null &&
SumDigits(after.Next.Value) < SumDigits(node.Value))
                after = after.Next;
            list.AddAfter(after, node);
        }
    }

static void SortListOwn(LinkedList<int> list)
{
    if (list.Count < 2) return;

    LinkedList<int> result = new LinkedList<int>();
    while (list.Count > 0)
    {
        LinkedListNode<int> node = list.First;
        list.RemoveFirst();
        Insert(result, node);
    }

    foreach (int elem in result)
        list.AddLast(elem);
}

```

```

    }

    static void SortList(LinkedList<int> list)
    {
        IComparer<int> digitComparer = new Comparer();

//        1)
        List<int> ll = new List<int>(list);
        ll.Sort(digitComparer);

//        2)
//        LinkedList<int> ll = new
LinkedList<int>(list.OrderBy(i => i, digitComparer));

        list.Clear();
        foreach (int elem in ll)
            list.AddLast(elem);
    }

    static void Main()
    {
        Console.WriteLine("Въведете име на файл:");
        String fileName = Console.ReadLine();
        LinkedList<int> list = ReadList(fileName);
        if (list == null)
            return;

        Console.WriteLine("Въведете число:");
        int k = int.Parse(Console.ReadLine());
        if (k <= 0)
            {

```

```

        Console.WriteLine("Невалидна стойност на
числото.");

        return;
    }

    FilterList(list, k);
    SortList(list);
    foreach (int e in list)
        Console.Write(e + " ");
    Console.WriteLine();
}
}
}

```

Java

```

import java.io.File;
import java.io.IOException;
import java.util.*;

public class Problem28 {

    private static ArrayList<Integer> readList(String fileName) {
        Scanner scanner;
        try {
            scanner = new Scanner(new File(fileName));
        } catch (IOException ex) {
            System.err.println("Проблем с отварянето на " + fileName);
            System.err.println(ex.getMessage());
            return null;
        }

        ArrayList<Integer> list = new ArrayList<>();
        try {
            while (scanner.hasNext()) {
                list.add(scanner.nextInt());
            }
        } catch (InputMismatchException ex) {
            System.err.println("Невалидни входни данни: " +
ex.getMessage());
        }
        scanner.close();
        return list;
    }
}

```

```

private static int sumDigits(int num) {
    num = Math.abs(num);
    int sum = 0;
    while (num != 0) {
        sum += num % 10;
        num /= 10;
    }
    return sum;
}

private static void filterList(ArrayList<Integer> list, int k) {
    if (list == null || list.isEmpty() || k == 0) {
        return;
    }
    k = Math.abs(k);

    ArrayList<Integer> result = new ArrayList<>();
    for (Integer i : list) {
        if (sumDigits(i) % k != 0) {
            result.add(i);
        }
    }
    list.clear();
    list.addAll(result);
}

public static void sortList(ArrayList<Integer> list) {
    Collections.sort(list, new Comparator<Integer>() {
        @Override
        public int compare(Integer i1, Integer i2) {
            return sumDigits(i1) - sumDigits(i2);
        }
    });

    /*
    Collections.sort(list, (i1, i2) -> {
        return sumDigits(i1) - sumDigits(i2);
    });

    list.sort((i1, i2) -> {
        return sumDigits(i1) - sumDigits(i2);
    });
    */
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Въведете име на файл:");
    String fileName = sc.nextLine();

    ArrayList<Integer> list = readList(fileName);
}

```

```
if (list == null) {
    System.out.println("Не са прочетени данни");
    return;
}

System.out.println("Въведете число:");
int number = sc.nextInt();
if (number <= 0) {
    System.out.println("Числото трябва да е строго
положително");
    return;
}

filterList(list, number);
sortList(list);
for (int i : list) {
    System.out.print(i + " ");
}
System.out.println();
}
}
```

Забележка: Признават се и всички други верни предложени решения.