

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.