

ОТГОВОРИ, УПЪТВАНИЯ И ПРИМЕРНИ ИЗГЛЕДИ
НА РЕШЕНИЯТА НА ПРАКТИЧЕСКИТЕ
ЗАДАЧИ ОТ

ТЕМА 1

Въпрос	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Отговор	В	А	В	Б	А	В	Г	Б	В	В	Б	А	Б	Б	А	В

17.

С#

```
for (int i = 0; i < myList.Count; i++)
using System;
using System.Collections.Generic;

namespace Program
{
    class Program
    {
        static void Main(string[] args)
        {
            List<int> myList = new List<int> { 5, 10, 8, 87, 24 };

            for (int i = 0; i < myList.Count; i++)
            {
                if (myList[i] == 5 || myList[i] == 8 || myList[i]
== 24)
                {
                    Console.WriteLine(myList[i]);
                }
            }
        }
    }
}
```

Java

```
for (int i = 0; i < myList.size(); i++)
import java.util.List;
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        List<Integer> myList = new
ArrayList<Integer>();
        myList.add(5);
        myList.add(10);
        myList.add(8);
    }
}
```

```

        myList.add(87);
        myList.add(24);

        for (int i = 0; i < myList.size();
i++) {
            if (myList.get(i) == 5 || myList.get(i) == 8 ||
myList.get(i) == 24) {
                System.out.println(myList.get(i));
            }
        }
    }
}

```

18.

```

CREATE TABLE Courses (
    Cid INT PRIMARY KEY,
    CTitle VARCHAR(100),
    StartDate DATE,
    EndDate DATE,
    Instructor VARCHAR(100),
    NumberParticipants INT
);

```

19.

C#
<pre> double time = 19.14; if (time < 10) { Console.WriteLine("Добро утро"); } else if (time < 20) { Console.WriteLine("Добър ден"); } else { Console.WriteLine("Добър вечер"); } </pre>
Java
<pre> double time = 19.14; if (time < 10) { System.out.println("Добро утро"); } </pre>

```

else if (time < 20)
{
    System.out.println("Добър ден");
} else
{
    System.out.println("Добър вечер");
}

```

20.

C#

или

```

if (min > numbers[i])
    min = numbers[i];
if (max < numbers[i])
    max = numbers[i];

if (numbers[i] < min)
    min = numbers[i];

if (numbers[i] > max)
    max = numbers[i];

```

```

using System;
class Program
{
    static void Main(string[] args)
    {
        int[] numbers = new int[10] { 25, 50, 37, 71, 14, 78, 92,
82, 75, 83 };
        int min = numbers[0], max = numbers[0];

        for (int i = 1; i < numbers.Length; i++)
        {
            if (numbers[i] < min)
                min = numbers[i];

            if (numbers[i] > max)
                max = numbers[i];
        }

        Console.WriteLine("max = " + max);
        Console.WriteLine("min = " + min);
    }
}

```

Java

```
        if (min > numbers[i])
            min = numbers[i];

        if (max < numbers[i])
            max = numbers[i];
или
        if (numbers[i] < min)
            min = numbers[i];

        if (numbers[i] > max)
            max = numbers[i];
public class Main {
public static void main(String[] args) {
    int[] numbers = { 25, 50, 37, 71, 14, 78, 92, 82, 75, 83
};
    int min = numbers[0], max = numbers[0];

    for (int i = 1; i < numbers.length; i++) {
        if (numbers[i] < min)
            min = numbers[i];

        if (numbers[i] > max)
            max = numbers[i];
    }

    System.out.println("max = " + max);
    System.out.println("min = " + min);
}
}
```

21.

```
CREATE TABLE Employees (
    EmployeeId INT PRIMARY KEY,
    FirstName NVARCHAR(50),
    LastName NVARCHAR(50),
    Department NVARCHAR(50),
    Salary DECIMAL(10, 2)
);
```

```
INSERT INTO Employees (EmployeeId, FirstName, LastName, Department,
Salary)
VALUES (1, 'Иван', 'Петров', 'Разработка', 5000);
```

```
INSERT INTO Employees (EmployeeId, FirstName, LastName, Department, Salary)
VALUES (2, 'Мария', 'Иванова', 'Маркетинг', 4500);
```

```
INSERT INTO Employees (EmployeeId, FirstName, LastName, Department, Salary)
VALUES (3, 'Петър', 'Георгиев', 'Продажби', 4800);
```

```
INSERT INTO Employees (EmployeeId, FirstName, LastName, Department, Salary)
VALUES (4, 'Анна', 'Димитрова', 'Администрация', 4200);
```

```
INSERT INTO Employees (EmployeeId, FirstName, LastName, Department, Salary)
VALUES (5, 'Георги', 'Николов', 'Финанси', 5200);
```

Или

```
INSERT INTO Employees (EmployeeId, FirstName, LastName, Department, Salary)
VALUES
    (1, 'Иван', 'Петров', 'Разработка', 5000),
    (2, 'Мария', 'Иванова', 'Маркетинг', 4500),
    (3, 'Петър', 'Георгиев', 'Продажби', 4800),
    (4, 'Анна', 'Димитрова', 'Администрация', 4200),
    (5, 'Георги', 'Николов', 'Финанси', 5200);
```

22. А

23. 1. Добавянето на блока `else if (diffOfDates == 7)` в условната конструкция е да се провери дали разликата между двете дати е точно 7 дни.

2. За подобряване на устойчивостта на програмата при въвеждане на некоректен формат на дата от потребителя може да се използва конструкцията `DateTime.TryParse` вместо `DateTime.Parse` в *C#*.

За подобряване устойчивостта на програмата при въвеждане на некоректен формат на дата от потребителя в *Java* може да се използват блокове на обработка на изключения (**try-catch блокове**).

24. Б

25.

C#

```
using System;
class Program
{
    static bool IsDivisibleByDigits(int number)
    {
        try
        {
            if (number == 0)
                return false;
            number = Math.Abs(number);
            while (number > 0)
            {
                int digit = number % 10;
                if (digit == 0 || number % digit != 0)
                    return false;
                number /= 10;
            }
            return true;
        }
        catch
        {
            return false;
        }
    }
    static void Main()
    {
        try
        {
            Console.Write("Въведете цяло число: ");
            int number = int.Parse(Console.ReadLine());
            if (IsDivisibleByDigits(number))
                Console.WriteLine("Yes");
            else
                Console.WriteLine("No");
        }
        catch
        {
            Console.WriteLine("Something went wrong!");
        }
    }
}
```

Java

```
import java.util.Scanner;
public class Main {
    static boolean isDivisibleByDigits(int number) {
        try {
            if (number == 0)
                return false;
            number = Math.abs(number);
            while (number > 0) {
                int digit = number % 10;
                if (digit == 0 || number % digit != 0)
                    return false;
                number /= 10;
            }
            return true;
        } catch (Exception e) {
            return false;
        }
    }
    public static void main(String[] args) {
        try {
            Scanner scanner = new Scanner(System.in);
            System.out.print("Въведете цяло число: ");
            int number = scanner.nextInt();

            if (isDivisibleByDigits(number))
                System.out.println("Yes");
            else
                System.out.println("No");
        } catch (Exception e) {
            System.out.println("Something went wrong!");
        }
    }
}
```

26.

2.

UPDATE Courses

SET NumberParticipants = 13

WHERE Cid = 3;

3.

SELECT *

FROM Courses

WHERE NumberParticipants > 15;

4.

```
SELECT COUNT(*)
FROM Courses
WHERE EndDate BETWEEN '2023-01-01' AND '2023-12-31';
SELECT CTitle, NumberParticipants
FROM Courses
WHERE EndDate < CURDATE()
ORDER BY CTitle;
```

27.

```
CREATE DATABASE bookstore;
```

```
USE bookstore;
```

```
CREATE TABLE authors (
    author_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    country VARCHAR(50)
);
```

```
INSERT INTO authors (author_id, first_name, last_name, country)
VALUES
```

```
(1, 'John', 'Green', 'USA'),
(2, 'J.K.', 'Rowling', 'UK'),
(3, 'Gabriel', 'Garcia', 'Colombia'),
(4, 'Jane', 'Austen', 'UK');
```

```
CREATE TABLE books (
    book_id INT PRIMARY KEY,
    title VARCHAR(100),
    genre VARCHAR(50),
    author_id INT,
    FOREIGN KEY (author_id) REFERENCES authors(author_id)
);
```

```
INSERT INTO books (book_id, title, genre, author_id)
VALUES
```

```
(1, 'The Fault in Our Stars', 'Young Adult', 1),
(2, 'Looking for Alaska', 'Young Adult', 1),
(3, 'Harry Potter and the Sorcerer's Stone', 'Fantasy', 2),
(4, 'Harry Potter and the Chamber of Secrets', 'Fantasy', 2),
(5, 'Love in the Time of Cholera', 'Romance', 3),
```


(6, 'One Hundred Years of Solitude', 'Magical Realism', 3),

(7, 'Pride and Prejudice', 'Romance', 4);

За да изведете всички книги на определен автор, може да използвате следната заявка, където да замените 'John' с името на автора, за който търсите:

```
SELECT b.title, b.genre
```

```
FROM books b
```

```
JOIN authors a ON b.author_id = a.author_id
```

```
WHERE a.first_name = 'John';
```

28.

C#

```
using System;
using System.Collections.Generic;
class Drink
{
    protected string name;
    protected double price;
    protected string size;
    public Drink(string name, string size, double price)
    {
        this.name = name;
        this.size = size;
        this.price = price;
    }
    public override string ToString()
    {
        return $"Drink: {name}, Size: {size}, Price: {price}";
    }
}
class Coffee : Drink
{
    private int strength;
    public Coffee(string name, string size, double price, int
strength)
        : base(name, size, price)
    {
        this.strength = strength;
    }
    public override string ToString()
    {
        return $"Coffee: {name}, Size: {size}, Strength:
{strength}, Price: {price}";
    }
}
```

```

class Tea : Drink
{
    private string type;
    public Tea(string name, string size, double price, string
type)
        : base(name, size, price)
    {
        this.type = type;
    }
    public override string ToString()
    {
        return $"Tea: {name}, Size: {size}, Type: {type}, Price:
{price}";
    }
}
class Program
{
    static void Main(string[] args)
    {
        List<Drink> drinks = new List<Drink>();
        drinks.Add(new Coffee("Espresso", "small", 3.50, 2));
        drinks.Add(new Tea("Green Tea", "medium", 2.00,
"green"));
        drinks.Add(new Coffee("Cappuccino", "large", 4.00, 3));
        foreach (Drink drink in drinks)
        {
            Console.WriteLine(drink);
        }
    }
}

```

Java

```

import java.util.ArrayList;
import java.util.List;
public class Main {
    public static void main(String[] args) {
        List<Drink> drinks = new ArrayList<>();
        drinks.add(new Coffee("Espresso", "small", 3.50, 2));
        drinks.add(new Tea("Green Tea", "medium", 2.00,
"green"));
        drinks.add(new Coffee("Cappuccino", "large", 4.00, 3));
        for (Drink drink : drinks) {
            System.out.println(drink);
        }
    }
    static class Drink {
        protected String name;

```

```

        protected double price;
        protected String size;
        public Drink(String name, String size, double price) {
            this.name = name;
            this.size = size;
            this.price = price;
        }
        @Override
        public String toString() {
            return "Drink: " + name + ", Size: " + size + ",
Price: " + price;
        }
    }
    static class Coffee extends Drink {
        private int strength;
        public Coffee(String name, String size, double price, int
strength) {
            super(name, size, price);
            this.strength = strength;
        }
        @Override
        public String toString() {
            return "Coffee: " + name + ", Size: " + size + ",
Strength: " + strength + ", Price: " + price;
        }
    }
    static class Tea extends Drink {
        private String type;
        public Tea(String name, String size, double price, String
type) {
            super(name, size, price);
            this.type = type;
        }
        @Override
        public String toString() {
            return "Tea: " + name + ", Size: " + size + ", Type:
" + type + ", Price: " + price;
        }
    }
}

```