

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

ТЕМА 2

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

17.

C#

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

public class Program
{
    public static void Main()
    {
        List<string> myList = new List<string> { "котка", "куче",
"заек", "хамстер" };
        Console.WriteLine("Напишете животно:");
        string animal = Convert.ToString(Console.ReadLine());

        foreach (string item in myList)
        {
            if (item.Contains(animal))
            {
                Console.WriteLine("Вашето животно е намерено в
списъка");
                break;
            }
        }
    }
}
```

Java

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        List<String> myList = new ArrayList<>();
        myList.add("котка");
        myList.add("куче");
        myList.add("заек");
        myList.add("хамстер");

        Scanner scanner = new Scanner(System.in);
        System.out.println("Напишете животно:");
        String animal = scanner.nextLine();
    }
}
```

```

        for (String item : myList) {
            if (item.contains(animal)) {
                System.out.println("Вашето животно е намерено в
списъка");
                break;
            }
        }
    }
}

```

18.

C#

```

string[] names = { "Иван", "Ива", "Георги", "георги",
"Петър", "Кирил" };
var stringToFind = "иван";

string[] result = Array.FindAll(names, element =>
element.ToLower() == stringToFind); // return Иван, иван

```

Java

```

String[] names = { "Иван", "Ива", "Георги", "георги", "Петър",
"Кирил" };
String stringToFind = "иван";

List<String> result = new ArrayList<>();
for (String name : names) {
    if (name.equalsIgnoreCase(stringToFind)) {
        result.add(name);
    }
}

```

19.

C#

```

using System;
public class Program
{
    public static void Main()
    {
        string word;
        int number;
        Console.Write("Въведете дума: ");
        word = Console.ReadLine();
        number = word.Length;
        Console.WriteLine("Броят на буквите в думата = " + number);
    }
}

```

Java

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        String word;
        int number;
        Scanner scanner = new Scanner(System.in);

        System.out.print("Въведете дума: ");
        word = scanner.nextLine();
        number = word.length();

        System.out.println("Броят на буквите в думата = " +
number);
    }
}
```

20. В**21. Б****22.**

name	type_of_blood
куче	топлокръвни
лъв	топлокръвни
гуша	студенокръвни
орел	студенокръвни
щука	студенокръвни
пеликан	студенокръвни
сом	студенокръвни

23. Въведеното число 21 е нечетно**24.**

(1) След последната операция стекът остава със следните елементи:

10, -7, 5

(2) В края на операциите в опашката има 7 елемента.

(3) В края на операциите елементите на опашката са:

-2, 6, -4, 8, -1, 2, -3

25.**C#**

```
using System;
class Program
{
    static void Main()
    {
        Console.Write("Въведете цяло число N: ");
        int N = int.Parse(Console.ReadLine());

        int sum = 0;

        for (int i = 1; i <= N; i++)
        {
            sum += i;
        }

        Console.WriteLine($"Сумата на числата от 1 до {N} е {sum}.");
    }
}
```

Java

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Въведете цяло число N: ");
        int N = scanner.nextInt();

        int sum = 0;

        for (int i = 1; i <= N; i++) {
            sum += i;
        }

        System.out.println("Сумата на числата от 1 до " + N + " е " + sum + ".");
    }
}
```

26.

C#

```
using System;
public class Person
{
    public string firstName { get; set; }
    public string lastName { get; set; }
    public int age { get; set; }

    public Person()
    {
        this.firstName = "Иван";
        this.lastName = "Петров";
        this.age = 18;
    }

    public void GetInfo()
    {
        Console.WriteLine(this.firstName + " " + this.lastName +
            ", " + this.age);
    }
}

public class Program
{
    public static void Main(string[] args)
    {
        Person[] persons = new Person[3];

        for (int i = 0; i < 3; i++)
        {
            persons[i] = new Person()
            {
                firstName = Console.ReadLine(),
                lastName = Console.ReadLine()
            };
        }

        for (int i = 0; i < 3; i++)
        {
            persons[i].GetInfo();
        }
    }
}
```

Java

```
import java.util.Scanner;
public class Person {
    private String firstName;
    private String lastName;
    private int age;

    public Person() {
        // Конструктор по подразбиране
        firstName = "Име";
        lastName = "Фамилия";
        age = 0;
    }

    public void GetInfo() {
        System.out.println("Име: " + firstName);
        System.out.println("Фамилия: " + lastName);
        System.out.println("Възраст: " + age);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Person[] people = new Person[3];

        // Въвеждаме данни за всеки елемент от масива
        for (int i = 0; i < 3; i++) {
            System.out.println("Въведете данни за човек " + (i +
1) + ":");
            Person person = new Person();
            System.out.print("Име: ");
            person.firstName = scanner.nextLine();
            System.out.print("Фамилия: ");
            person.lastName = scanner.nextLine();
            System.out.print("Възраст: ");
            person.age = Integer.parseInt(scanner.nextLine());

            people[i] = person;
        }

        // Извикваме метода GetInfo() за всеки елемент от масива
        for (int i = 0; i < 3; i++) {
            System.out.println("Данни за човек " + (i + 1) + ":");
            people[i].GetInfo();
            System.out.println();
        }
    }
}
```

27.

1.

```
CREATE TABLE Courses(  
ID INT PRIMARY KEY,  
Course_Title varchar(100) NOT NULL,  
Course_Teacher varchar(100),  
Hours INT,  
Course_date DATE);
```

2.

```
INSERT INTO Courses (ID, Course_Title, Course_Teacher, Hours, Course_date)  
VALUES(1, 'Английски език', 'Иван Атанасов', 30, '2022-12-20');  
INSERT INTO Courses (ID, Course_Title, Course_Teacher, Hours, Course_date)  
VALUES(2, 'Английски език', 'Светлана Стефанова', 60, '2022-12-27');  
INSERT INTO Courses (ID, Course_Title, Course_Teacher, Hours, Course_date)  
VALUES(3, 'Програмиране на Java', 'Петър Колев', 60, '2023-01-11');  
INSERT INTO Courses (ID, Course_Title, Course_Teacher, Hours, Course_date)  
VALUES(4, 'Алгоритми', 'Светлин Киров', 30, '2023-01-11');  
INSERT INTO Courses (ID, Course_Title, Course_Teacher, Hours, Course_date)  
VALUES(5, 'Структури от данни', 'Светлин Киров', 60, '2023-01-20');
```

3.

```
SELECT Course_Title, Hours  
FROM Courses  
WHERE Course_Teacher LIKE '%Киров';
```

4.

```
SELECT COUNT(ID)  
FROM Courses  
WHERE YEAR(Course_date)=2023;
```

5.

```
SELECT Course_Title, Hours  
FROM Courses  
WHERE Hours>(SELECT AVG(Hours) FROM Courses);
```

6.

```
DELETE FROM Courses  
WHERE Hours<60;
```

28.

C#

```

using System;
using System.Data;

class Program
{
    static void Main()
    {
        while (true)
        {
            Console.Write("Въведете математическа формула или
,quit' за изход: ");
            string input = Console.ReadLine();

            if (input.ToLower() == "quit")
            {
                Console.WriteLine("Приложението се изключва.");
                break;
            }

            try
            {
                DataTable table = new DataTable();
                table.Columns.Add("expression", typeof(string),
input);

                DataRow row = table.NewRow();
                table.Rows.Add(row);
                double result = double.Parse((string)
row["expression"]);
                Console.WriteLine($"Резултатът е {result}.");
            }
            catch (Exception ex)
            {
                Console.WriteLine($"Грешка: {ex.Message}");
            }
        }
    }
}

```

Java

```

import java.util.Scanner;
public class Calculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
    }
}

```



```

        while (true) {
            System.out.print("Въведете математическа формула или
'quit' за изход: ");
            String input = scanner.nextLine();

            if (input.equalsIgnoreCase("quit")) {
                System.out.println("Приложението се изключва.");
                break;
            }

            try {
                double result = evaluateExpression(input);
                System.out.println("Резултатът е " + result +
".");
            } catch (Exception ex) {
                System.out.println("Грешка: " + ex.getMessage());
            }
        }

        private static double evaluateExpression(String expression) {
            return new Object() {
                int pos = -1, ch;

                void nextChar() {
                    ch = (++pos < expression.length()) ? expression.
charAt(pos) : -1;
                }

                boolean isDigit(int c) {
                    return c >= '0' && c <= '9';
                }

                double parse() {
                    nextChar();
                    double x = parseExpression();
                    if (ch != -1)
                        throw new RuntimeException("Невалидна формула");
                    return x;
                }

                double parseExpression() {
                    double x = parseTerm();
                    for (; ; ) {
                        if (ch == '+') {
                            nextChar();
                            x += parseTerm();
                        } else if (ch == '-') {

```

```

        nextChar();
        x -= parseTerm();
    } else
        return x;
    }
}

double parseTerm() {
    double x = parseFactor();
    for (; ; ) {
        if (ch == '*') {
            nextChar();
            x *= parseFactor();
        } else if (ch == '/') {
            nextChar();
            x /= parseFactor();
        } else
            return x;
    }
}

double parseFactor() {
    if (ch == '(') {
        nextChar();
        double x = parseExpression();
        if (ch != ')')
            throw new RuntimeException("Очаква се
затваряща скоба");
        nextChar();
        return x;
    }
    if (ch == 'x') {
        nextChar();
        return 42; // Пример за използване на
променлива ,x‘
    }
    int startPos = pos;
    while ((ch >= '0' && ch <= '9') || ch == '.')
        nextChar();
    if (startPos == pos)
        throw new RuntimeException("Очаква се
число");
    return Double.parseDouble(expression.
substring(startPos, pos));
    }
}.parse();
}
}

```