

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

ТЕМА 10

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

17. Ред 0: 7 2 3

Ред 1: 1 9 6

18.

C#

```
using System;
class Program
{
    static void Main()
    {
        double sum = 0, avg = 0;
        double[] numbers = { 10, 20, 30, 40 };
        for (int i = 0; i < numbers.Length; i++)
        {
            sum += numbers[i];
        }
        avg = sum / numbers.Length;
        Console.WriteLine("Сума: " + sum);
        Console.WriteLine("Средна стойност: " + avg);
    }
}
```

Java

```
public class Main
{
    public static void main(String[] args) {
        double sum = 0, avg = 0;
        double[] numbers = { 10, 20, 30, 40 };
        for (int i = 0; i < numbers.length; i++) {
            sum += numbers[i];
        }
        avg = sum / numbers.length;
        System.out.println("Сума: " + sum);
        System.out.println("Средна стойност: " + avg);
    }
}
```

19. 5 73 48 28 97

20. Г

21. А

22. Отговор:

Exam_Title	Student_Id
Английски език	126-12
Алгоритми	126-12
Програмиране на Java	126-12

23. Г

24. 3

25.

C#
<pre>using System; class Program { static void Main() { try { Console.WriteLine("Въведете ден:"); int day = int.Parse(Console.ReadLine()); Console.WriteLine("Въведете месец:"); int month = int.Parse(Console.ReadLine()); Console.WriteLine("Въведете година:"); int year = int.Parse(Console.ReadLine()); if (IsValidDate(day, month, year)) { Console.WriteLine("Датата е валидна."); } else { Console.WriteLine("Датата не е валидна."); } } catch (Exception) { Console.WriteLine("Something went wrong!"); } } static bool IsValidDate(int day, int month, int year) { if (year < 1 month < 1 month > 12) { return false; } int[] daysInMonth = { 0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 }; if (IsLeapYear(year)) { daysInMonth[2] = 29; // Високосна година, февруари има 29 дни } } }</pre>

```

        return day >= 1 && day <= daysInMonth[month];
    }
    static bool IsLeapYear(int year)
    {
        return (year % 4 == 0 && year % 100 != 0) || (year % 400
== 0);
    }

```

Java

```

import java.util.Scanner;
public class DateValidationProgram {
    public static void main(String[] args) {
        try {
            Scanner scanner = new Scanner(System.in);
            System.out.println("Въведете ден:");
            int day = scanner.nextInt();
            System.out.println("Въведете месец:");
            int month = scanner.nextInt();
            System.out.println("Въведете година:");
            int year = scanner.nextInt();
            if (isValidDate(day, month, year)) {
                System.out.println("Датата е валидна.");
            } else {
                System.out.println("Датата не е валидна.");
            }
        } catch (Exception e) {
            System.out.println("Something went wrong!");
        }
    }
    static boolean isValidDate(int day, int month, int year) {
        if (year < 1 || month < 1 || month > 12) {
            return false;
        }
        int[] daysInMonth = { 0, 31, 28, 31, 30, 31, 30, 31, 31,
30, 31, 30, 31 };
        if (isLeapYear(year)) {
            daysInMonth[2] = 29; // Високосна година, февруари
има 29 дни
        }
        return day >= 1 && day <= daysInMonth[month];
    }
    static boolean isLeapYear(int year) {
        return (year % 4 == 0 && year % 100 != 0) || (year % 400
== 0);
    }
}

```

26.

- Заявка за създаване на таблицата „books“:

CREATE TABLE books (ID AutoNumber PRIMARY KEY, Заглавие Text, Автор Text, „Брой страници“ Number, Цена Currency);

- Пет заявки, които зареждат в таблицата следните кортежи:

ID	Заглавие	Автор	Брой страници	Цена
1	Игра на тронове	Джордж Р. Р. Мартин	694	25.99
2	Престъпление и наказание	Фьодор Достоевски	551	19.99
3	Малкият принц	Антоан дьо Сент-Екзюпери	96	12.50
4	1984	Джордж Оруел	328	15.99
5	Хобитът	Дж. Р. Р. Толкин	310	22.75

INSERT INTO books (ID, Заглавие, Автор, „Брой страници“, Цена) VALUES (1, 'Игра на тронове', 'Джордж Р. Р. Мартин', 694, 25.99), (2, 'Престъпление и наказание', 'Фьодор Достоевски', 551, 19.99), (3, 'Малкият принц', 'Антоан дьо Сент-Екзюпери', 96, 12.50), (4, '1984', 'Джордж Оруел', 328, 15.99), (5, 'Хобитът', 'Дж. Р. Р. Толкин', 310, 22.75);

- Заявка, която изтрива данните за книгата „Малкият принц“:

DELETE FROM books WHERE Заглавие = 'Малкият принц';

- Заявка, която извежда общата сума (Брой страници * Цена), с добавен ДДС (20%) към цената, за всички книги:

SELECT Заглавие, „Брой страници“ * Цена * 1.2 AS „Обща сума с ДДС“ FROM books;

- Заявка, която намира броят на всички страници за книгите на автора „Джордж Р. Р. Мартин“:

SELECT SUM(„Брой страници“) AS „Общ брой страници“ FROM books WHERE Автор = 'Джордж Р. Р. Мартин';

27.

C#
<pre>using System; using System.Collections.Generic; class Movie { public string Title { get; set; } public string Director { get; set; } public int Year { get; set; } }</pre>

```

    public Movie(string title, string director, int year)
    {
        Title = title;
        Director = director;
        Year = year;
    }
    public override string ToString()
    {
        return $"Movie: {Title}, Director: {Director}, Year: {Year}";
    }
}
class FilmLibrary
{
    public string Name { get; set; }
    public List<Movie> Movies { get; set; }
    public FilmLibrary(string name)
    {
        Name = name;
        Movies = new List<Movie>();
    }
    public void AddMovie(string title, string director, int year)
    {
        Movie newMovie = new Movie(title, director, year);
        Movies.Add(newMovie);
    }
    public void DisplayInfo()
    {
        Console.WriteLine($"Film library: {Name}");
        foreach (var movie in Movies)
        {
            Console.WriteLine(movie);
        }
    }
}
class Program
{
    static void Main()
    {
        Console.WriteLine("Enter a name for the library: ");
        string libraryName = Console.ReadLine();
        FilmLibrary filmLibrary = new FilmLibrary(libraryName);
        char choice;
        do
        {
            Console.WriteLine("Select an option:");
            Console.WriteLine("a - adding a movie");
            Console.WriteLine("v - view information about the film library");
            Console.WriteLine("q - exit");
            choice = Console.ReadKey().KeyChar;
            switch (choice)
            {
                case 'a':

```

```

        Console.WriteLine("\nEnter a movie title: ");
        string title = Console.ReadLine();
        Console.WriteLine("Enter the director of the film:");

        string director = Console.ReadLine();
        int year;
        do
        {
            Console.WriteLine("Enter year of issue: ");
        } while (!int.TryParse(Console.ReadLine(),
out year));

        filmLibrary.AddMovie(title, director, year);
        Console.WriteLine("The movie was successfully
added!\n");

        break;
    case 'v':
        Console.WriteLine();
        filmLibrary.DisplayInfo();
        Console.WriteLine();
        break;
    }
} while (choice != 'q');
}
}

```

Java

```

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
abstract class Perfume {
    protected String name;
    protected double price;
    protected String brand;
    protected int volume;
    public Perfume(String name, double price, String brand, int
volume) {
        this.name = name;
        this.price = price;
        this.brand = brand;
        this.volume = volume;
    }
    // getters and setters
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public double getPrice() {

```

```

        return price;
    }
    public void setPrice(double price) {
        this.price = price;
    }
    public String getBrand() {
        return brand;
    }
    public void setBrand(String brand) {
        this.brand = brand;
    }
    public int getVolume() {
        return volume;
    }
    public void setVolume(int volume) {
        this.volume = volume;
    }
    public abstract double calculateDiscountedPrice();
}
class EauDeToilette extends Perfume {
    public EauDeToilette(String name, double price, String brand,
int volume) {
        super(name, price, brand, volume);
    }
    public double calculateDiscountedPrice() {
        // 10% discount for EauDeToilette
        return getPrice() * 0.9;
    }
}
class EauDeParfum extends Perfume {
    public EauDeParfum(String name, double price, String brand,
int volume) {
        super(name, price, brand, volume);
    }
    @Override
    public double calculateDiscountedPrice() {
        // 15% discount for EauDeParfum
        return getPrice() * 0.85;
    }
}
class PerfumeShopMain {
    public static void main(String[] args) {
        List<Perfume> perfumes = new ArrayList<>();
        System.out.println("Enter perfumes information (type ,END"
to finish):");
        Scanner scanner = new Scanner(System.in);
        while (true) {
            String input = scanner.nextLine();
            if (input.equals("END"))
                break;

```

```

        String[] tokens = input.split(" ");
        if (tokens.length >= 2) {
            String typeProduct = tokens[0];
            double productionPrice;
            try {
                productionPrice = Double.
parseDouble(tokens[1]);
            } catch (NumberFormatException e) {
                System.out.println("Invalid production price.
Skipping...");
                continue;
            }
            Perfume perfume;
            if (typeProduct.equals("EauDeToilette")) {
                perfume = new EauDeToilette("EauDeToilette",
productionPrice, "Brand1", 100);
            } else if (typeProduct.equals("EauDeParfum")) {
                perfume = new EauDeParfum("EauDeParfum",
productionPrice, "Brand2", 50);
            } else {
                System.out.println("Invalid perfume type.
Skipping...");
                continue;
            }
            perfumes.add(perfume);
        } else {
            System.out.println("Invalid input. Skipping...");
        }
    }
    System.out.println("\nAll perfumes:");
    for (Perfume perfume : perfumes) {
        System.out.printf("%s by %s - Original Price: $%.2f,
Discounted Price: $%.2f\n",
            perfume.getName(), perfume.getBrand(),
            perfume.getPrice(), perfume.calculateDiscountedPrice());
    }
}
}

```

28.

```

C#
using System;
using System.Collections.Generic;
abstract class Perfume
{
    protected string name;
    protected double price;
    protected string brand;
    protected int volume;
}

```



```

    public Perfume(string name, double price, string brand, int
volume)
    {
        this.name = name;
        this.price = price;
        this.brand = brand;
        this.volume = volume;
    }
    // getters and setters
    public string Name
    {
        get { return name; }
        set { name = value; }
    }
    public double Price
    {
        get { return price; }
        set { price = value; }
    }
    public string Brand
    {
        get { return brand; }
        set { brand = value; }
    }
    public int Volume
    {
        get { return volume; }
        set { volume = value; }
    }
    public abstract double CalculateDiscountedPrice();
}
class EauDeToilette : Perfume
{
    public EauDeToilette(string name, double price, string brand,
int volume) : base(name, price, brand, volume)
    {
    }
    public override double CalculateDiscountedPrice()
    {
        // 10% discount for EauDeToilette
        return Price * 0.9;
    }
}
class EauDeParfum : Perfume
{
    public EauDeParfum(string name, double price, string brand,
int volume) : base(name, price, brand, volume)
    {
    }
    public override double CalculateDiscountedPrice()

```

```

    {
        // 15% discount for EauDeParfum
        return Price * 0.85;
    }
}
class PerfumeShopMain
{
    static void Main()
    {
        List<Perfume> perfumes = new List<Perfume>();
        // Въвеждане на парфюми
        Console.WriteLine("Enter perfumes information (type ,END"
to finish):");
        while (true)
        {
            string input = Console.ReadLine();
            if (input == "END")
                break;
            string[] tokens = input.Split(, ,);
            if (tokens.Length >= 2)
            {
                string typeProduct = tokens[0];
                double productionPrice;
                if (double.TryParse(tokens[1], out
productionPrice))
                {
                    Perfume perfume;
                    if (typeProduct == "EauDeToilette")
                    {
                        perfume = new
EauDeToilette("EauDeToilette", productionPrice, "Brand1", 100);
                    }
                    else if (typeProduct == "EauDeParfum")
                    {
                        perfume = new EauDeParfum("EauDeParfum",
productionPrice, "Brand2", 50);
                    }
                    else
                    {
                        Console.WriteLine("Invalid perfume type.
Skipping...");
                        continue;
                    }
                    perfumes.Add(perfume);
                }
                else
                {
                    Console.WriteLine("Invalid production price.
Skipping...");
                }
            }
        }
    }
}

```

```

        else
        {
            Console.WriteLine("Invalid input. Skipping...");
        }
    }
    Console.WriteLine("\nAll perfumes:");
    foreach (Perfume perfume in perfumes)
    {
        Console.WriteLine($"{perfume.Name} by {perfume.Brand}
- Original Price: ${perfume.Price}, Discounted Price: ${perfume.
CalculateDiscountedPrice():F2}");
    }
}
}
}

```

Java

```

import java.util.ArrayList;
abstract class Perfume {
    protected String name;
    protected double price;
    protected String brand;
    protected int volume;
    public Perfume(String name, double price, String brand, int
volume) {
        this.name = name;
        this.price = price;
        this.brand = brand;
        this.volume = volume;
    }
    // getters and setters
    public abstract double calculateDiscountedPrice();
}
class EauDeToilette extends Perfume {
    public EauDeToilette(String name, double price, String brand,
int volume) {
        super(name, price, brand, volume);
    }
    public double calculateDiscountedPrice() {
        // 10% discount for EauDeToilette
        return price * 0.9;
    }
}
class EauDeParfum extends Perfume {
    public EauDeParfum(String name, double price, String brand,
int volume) {
        super(name, price, brand, volume);
    }
    public double calculateDiscountedPrice() {
        // 15% discount for EauDeParfum

```

```

        return price * 0.85;
    }
}
class PerfumeShopMain {
    public static void main(String[] args) {
        ArrayList<Perfume> perfumes = new ArrayList<>();

        perfumes.add(new EauDeToilette("Cool Breeze", 50.0,
"FreshScents", 100));
        perfumes.add(new EauDeParfum("Elegant Rose", 80.0,
"LuxuryFragrance", 50));
        // Display information for each perfume
        for (Perfume perfume : perfumes) {
            System.out.println(perfume.name + " by " + perfume.
brand + " - Original Price: $" + perfume.price +
            ", Discounted Price: $" + perfume.
calculateDiscountedPrice());
        }
    }
}

```