

Създаване на конзолни програми - първи стъпки — 8б, Въведение в програмирането

1. Какво е конзолна програма?

Конзолна програма е програма, която работи в текстов режим - използва текстовия интерфейс за комуникация с потребителя. Всички входове и изходи са текстове, които се показват в прозореца на конзолата (Command Prompt в Windows или Terminal в Linux/Mac). ★

Основни характеристики на конзолните програми:

- **Текстов интерфейс** - без графични елементи ★
- **Бързо изпълнение** - минимални ресурси ★
- **Прости за писане** - идеални за начинаещи ★
- **Крос-платформени** - работят на различни операционни системи ★

2. Първа конзолна програма

Класическият "Hello World":

```
using System;

class Program {
    static void Main() {
        Console.WriteLine("Здравей, свят!");
        Console.WriteLine("Това е моята първа конзолна програма!");

        Console.WriteLine("\nНатисни произволен клавиш за изход...");
        Console.ReadKey();
    }
}
```

```
>
Здравей, свят!
Това е моята първа конзолна програма!
Натисни произволен клавиш за изход...
```

3. Основни команди за конзолен изход ★

Console класът - основни методи:

- **Console.WriteLine()** - извежда текст и преминава на нов ред ★
- **Console.Write()** - извежда текст без да преминава на нов ред ★
- **Console.ReadLine()** - чете ред от конзолата ★
- **Console.ReadKey()** - чете един символ и чака натискане на клавиш ★
- **Console.Clear()** - изчиства екрана на конзолата ★

Примери с различни команди:

```
using System;

class Program {
    static void Main() {
        // Извеждане на текст
        Console.WriteLine("Първи ред");
        Console.WriteLine("Втори ред");

        // Извеждане без нов ред
        Console.Write("Текст на ");
        Console.Write("същия ред");
        Console.WriteLine(); // Премахване на нов ред

        // Изчистване на екрана
        Console.WriteLine("Натисни клавиш за изчистване на екрана...");
        Console.ReadKey();
        Console.Clear();
        Console.WriteLine("Екранът е изчистен!");

        Console.WriteLine("\nНатисни клавиш за изход...");
        Console.ReadKey();
    }
}
```

4. Четене на вход от потребителя

Как да четем данни от потребителя:

1. Използвай `Console.ReadLine()` за четене на текст ★
2. Запази резултата в променлива от тип `string` ★
3. Ако трябва число, конвертирай с `int.Parse()` или `Convert.ToInt32()` ★
4. Винаги добавяй съобщения, които обясняват какво се очаква ★

Програма за въвеждане на име и възраст:

```
using System;

class Program {
    static void Main() {
        Console.WriteLine("=== Добре дошли в програмата! ===");

        // Четене на име
        Console.Write("Въведете вашето име: ");
        string name = Console.ReadLine();

        // Четене на възраст
        Console.Write("Въведете вашата възраст: ");
        string ageText = Console.ReadLine();
        int age = int.Parse(ageText);

        // Извеждане на резултата
        Console.WriteLine($"Здравейте, {name}!");
        Console.WriteLine($"Вие сте на {age} години.");

        Console.WriteLine("\nНатисни клавиш за изход...");
        Console.ReadKey();
    }
}
```

```
>
=== Добре дошли в програмата! ===
Въведете вашето име: Иван
Въведете вашата възраст: 15
Здравейте, Иван!
Вие сте на 15 години.
Натисни клавиш за изход...
```

5. Основни типове данни ★

Най-често използвани типове данни:

- **string** - текст (например: "Здравей") ★
- **int** - цели числа (например: 25, -10, 0) ★
- **double** - десетични числа (например: 3.14, 2.5) ★
- **bool** - логически тип (true или false) ★
- **char** - един символ (например: 'A', '5') ★

Работа с различни типове данни:

```
using System;

class Program {
    static void Main() {
        // Деклариране на променливи
        string name = "Мария";
        int age = 14;
        double height = 1.65;
        bool isStudent = true;
        char grade = 'A';

        // Извеждане на информация
        Console.WriteLine("=== Информация за ученика ===");
        Console.WriteLine($"Име: {name}");
        Console.WriteLine($"Възраст: {age} години");
        Console.WriteLine($"Височина: {height} метра");
        Console.WriteLine($"Студент: {(isStudent ? "Да" : "Не")}");
        Console.WriteLine($"Оценка: {grade}");

        Console.WriteLine("\nНатисни клавиш за изход...");
        Console.ReadKey();
    }
}
```

6. Аритметични операции

Основни математически операции:

- $+$ - събиране ★
- $-$ - изваждане ★
- $*$ - умножение ★
- $/$ - деление ★
- $\%$ - остатък от деление (модуло) ★

Калкулатор програма:

```
using System;

class Program {
    static void Main() {
        Console.WriteLine("=== Прост калкулатор ===");

        // Четене на числата
        Console.Write("Въведете първото число: ");
        double num1 = double.Parse(Console.ReadLine());

        Console.Write("Въведете второто число: ");
        double num2 = double.Parse(Console.ReadLine());

        // Извършване на операциите
        double sum = num1 + num2;
        double difference = num1 - num2;
        double product = num1 * num2;
        double quotient = num1 / num2;

        // Извеждане на резултатите
        Console.WriteLine($"\\nРезултати:");
        Console.WriteLine($"{num1} + {num2} = {sum}");
        Console.WriteLine($"{num1} - {num2} = {difference}");
        Console.WriteLine($"{num1} * {num2} = {product}");
        Console.WriteLine($"{num1} / {num2} = {quotient:F2}");

        Console.WriteLine("\\nНатисни клавиш за изход...");
        Console.ReadKey();
    }
}
```

7. Прости условия (if-else) ★

Синтаксис за условия:

- **if** - ако условието е вярно ★
- **else if** - ако предишното условие е невярно, но това е вярно ★
- **else** - ако никое условие не е вярно ★
- **Оператори за сравнение:** == (равно), != (различно), > (по-голямо), < (по-малко), >= (по-голямо или равно), <= (по-малко или равно) ★

Програма за определяне на успех:

```
using System;

class Program {
    static void Main() {
        Console.WriteLine("=== Калкулатор на успех ===");

        Console.Write("Въведете вашата оценка: ");
        double grade = double.Parse(Console.ReadLine());

        // Проверка на оценката
        if (grade >= 5.5) {
            Console.WriteLine("Отличен успех! 🏆");
            Console.WriteLine("Вие сте много добър ученик!");
        }
        else if (grade >= 4.5) {
            Console.WriteLine("Много добър успех! 👍");
            Console.WriteLine("Продължавайте така!");
        }
        else if (grade >= 3.5) {
            Console.WriteLine("Добър успех! ✅");
            Console.WriteLine("Можете да се подобрите още!");
        }
        else if (grade >= 3.0) {
            Console.WriteLine("Среден успех. 📚");
            Console.WriteLine("Трябва да учите повече!");
        }
        else {
            Console.WriteLine("Слаб успех. ⚠️");
            Console.WriteLine("Необходимо е сериозно подобрение!");
        }

        Console.WriteLine("\nНатисни клавиш за изход...");
        Console.ReadKey();
    }
}
```

8. Прости цикли (for, while)

Видове цикли:

- **for цикъл** - когато знаем колко пъти да повторим ★
- **while цикъл** - когато не знаем колко пъти, но имаме условие ★
- **do-while цикъл** - изпълнява се поне веднъж ★

Примери с цикли:

```
using System;

class Program {
    static void Main() {
        Console.WriteLine("=== Примери с цикли ===");

        // For цикъл - броене от 1 до 5
        Console.WriteLine("For цикъл - числа от 1 до 5:");
        for (int i = 1; i <= 5; i++) {
            Console.WriteLine($"Число: {i}");
        }

        Console.WriteLine();

        // While цикъл - таблица за умножение
        Console.WriteLine("Таблица за умножение с 3:");
        int number = 1;
        while (number <= 10) {
            int result = 3 * number;
            Console.WriteLine($"3 × {number} = {result}");
            number++;
        }

        Console.WriteLine();

        // Do-while цикъл - четене до въвеждане на 0
        Console.WriteLine("Въведете числа (0 за изход):");
        int input;
        do {
            Console.Write("Въведете число: ");
            input = int.Parse(Console.ReadLine());
            Console.WriteLine($"Въведеното число: {input}");
        } while (input != 0);

        Console.WriteLine("Програмата приключи!");
        Console.WriteLine("\nНатисни клавиш за изход...");
        Console.ReadKey();
    }
}
```

9. Пълна програма - Игра "Познай числото"

Комбиниране на всички изучени концепции:

```
using System;

class Program {
    static void Main() {
        Console.WriteLine("=== Игра 'Познай числото' ===");
        Console.WriteLine("Компютърът ще мисли за число между 1 и 100.");
        Console.WriteLine("Опитайте се да го познаете!");
        Console.WriteLine();

        // Генериране на случайно число
        Random random = new Random();
        int secretNumber = random.Next(1, 101);

        int attempts = 0;
        int guess;

        Console.WriteLine("Числото е генерирано! Започнете да познавате...");

        // Основен цикъл на играта
        do {
            Console.Write($"Опит {attempts + 1}: Въведете вашето предположение: ");
            guess = int.Parse(Console.ReadLine());
            attempts++;

            if (guess < secretNumber) {
                Console.WriteLine("Числото е по-голямо! ⬆️");
            }
            else if (guess > secretNumber) {
                Console.WriteLine("Числото е по-малко! ⬆️");
            }
            else {
                Console.WriteLine($"🎉 Поздравления! Познахте числото {secretNumber}!");
                Console.WriteLine($"Направихте {attempts} опита.");
            }

        } while (guess != secretNumber);

        // Оценка на резултата
        if (attempts <= 3) {
            Console.WriteLine("Отлично! Много бързо познахте!");
        }
        else if (attempts <= 7) {
```

```
        Console.WriteLine("Добре! Добър резултат!");  
    }  
    else {  
        Console.WriteLine("Можете да се подобрите! Практикувайте  
повече!");  
    }  
  
    Console.WriteLine("\nБлагодарим за играта!");  
    Console.WriteLine("Натисни клавиш за изход...");  
    Console.ReadKey();  
}  
}
```

10. Структура на конзолна програма ★

Стандартна структура:

1. **using System;** - включване на необходимите библиотеки ★
2. **namespace** - име на проекта (опционално за начинаещи) ★
3. **class Program** - основен клас ★
4. **static void Main()** - начална точка на програмата ★
5. **Код на програмата** - основните команди ★
6. **Console.ReadKey()** - спиране на програмата ★

Шаблон за нова конзолна програма:

```
using System;

class Program {
    static void Main() {
        // Вашият код тук

        Console.WriteLine("\nНатисни клавиш за изход...");
        Console.ReadKey();
    }
}
```

11. Добри практики

Съвети за писане на добри конзолни програми:

- **Ясни съобщения** - обяснявайте на потребителя какво се очаква ★
- **Проверка на входните данни** - винаги проверявайте дали входът е валиден ★
- **Четлив код** - използвайте смислени имена на променливите ★
- **Коментари** - обяснявайте сложните части от кода ★
- **Тестване** - винаги тествайте програмата с различни входове ★
- **Обработка на грешки** - предвиждайте възможни проблеми ★

12. Практически задачи

Задачи за упражнение:

1. **Калкулатор за площ** - изчисляване на площта на правоъгълник ★
2. **Конвертор на валута** - преобразуване на лева в евро ★
3. **Програма за възраст** - определяне на дали човек е пълнолетен ★
4. **Брояч на думи** - броене на думите в изречение ★
5. **Генериране на парола** - създаване на случайна парола ★
6. **Игра "Камък-Ножица-Хартия"** - игра срещу компютъра ★

Примерна задача - Калкулатор за площ:

```
using System;

class Program {
    static void Main() {
        Console.WriteLine("=== Калкулатор за площ на правоъгълник ===");

        Console.Write("Въведете дължината (см): ");
        double length = double.Parse(Console.ReadLine());

        Console.Write("Въведете ширината (см): ");
        double width = double.Parse(Console.ReadLine());

        double area = length * width;

        Console.WriteLine($"Площта на правоъгълника е: {area} кв. см");

        if (area > 100) {
            Console.WriteLine("Това е голям правоъгълник!");
        } else if (area > 50) {
            Console.WriteLine("Това е среден правоъгълник.");
        } else {
            Console.WriteLine("Това е малък правоъгълник.");
        }

        Console.WriteLine("\nНатисни клавиш за изход...");
        Console.ReadKey();
    }
}
```

13. Заключение

Конзолните програми са отлична отправна точка за изучаване на програмирането. Те са прости за писане и разбиране, но позволяват да се изучат всички основни концепции на програмирането.

Ключови концепции, които научихте:

- **Структура на програмата** - как се организира C# код ★
- **Вход и изход** - четене и извеждане на данни ★
- **Типове данни** - работа с различни видове информация ★
- **Аритметични операции** - математически изчисления ★
- **Условия** - правене на решения в програмата ★
- **Цикли** - повтаряне на действия ★
- **Добри практики** - писане на качествен код ★

Следващи стъпки: Продължете с изучаването на по-сложни структури данни, методи, класове и обекти. Практикувайте редовно с различни задачи!

