//void methods -не връща резултат.

// име - глагол + друга дума

// () - означава че има метод

// { } -блок от код

// задължително в class

// след команда отива жъж воид медода изпълнява го и се връща

метод може да извиква друг метод

// PrinName();

// PrinAge();

//}

//public static void PrinName()

//{

// Console.WriteLine("Ivan");

//}

//public static void PrinAge()

//{

// Console.WriteLine("30");

//}

}

метод не може да извика себе си - Рекурсия.

public class Program

{

public static void Main()

{

PrintFullName();

}

public static void PrintFullName()

{

PrintFullName();

}

}

ако променливата е извън метод но е каласа мога да я използвам във всички методи

public class Program

{

public static int numbers = 100;

public static void Main()

{

Console.WriteLine(numbers);

}

public static void PrintFullName()

{

Console.WriteLine(numbers);

}

}

стремим се един метод да решава само една задача

ако копираме код някоклко пъти по добре да изнесем метод и да го използваме. не повтаряме код

воид метода не връща резултат

параметри - мога да подавам информация с тях

public static void Main()

{

PrintText("Pesho");

PrintText("ivan");

}

public static void PrintText(string text)

{

text живее само в метода

Console.WriteLine(text);

}

public static void Main()

{

PrintNumber(100);

}

public static void PrintNumber(int number)

{

if (number < 1)

{

Console.WriteLine("Invalid Number");

}

for (int i = 0; i < number; i++)

{

Console.WriteLine(i + 1);

}

}

може да има повече от един параметър. приблизително най много 7 параметъра.

когато имаме повече от един параметър, трябва да ги задаваме в същият ред в който сме ги декларирали

public static void Main()

{

Print(100, "the thing", true);

}

public static void Print(int number, string text, bool someThing)

{

Console.WriteLine(number);

Console.WriteLine(text);

Console.WriteLine(someThing);

}

параметрите могат да идват и от променливи. отвън как се казват те няма значение. важен е типът на променливата

няма значение как са наредени методите.важно е да съвпада името и типът на параметрите

public static void Main()

{

int number = int.Parse(Console.ReadLine());

PrintSign(number);

}

public static void PrintSign(int number)

{

if (number > 0)

{

Console.WriteLine(number);

}

else if (number <0)

{

}

}

методите могат да приемат дефолтни стойности

public static void Main()

{

PrintNumbers();

}

public static void PrintNumbers(int start = 0, int end = 1000)

{

for (int i = start; i < end; i++)

{

Console.WriteLine(i);

}

}

има и поименнно извикване

PrintNumbers(start:100, end:300);

връщане на стойност от методите. Return - прекратява метода и връща резултат

ако искаме да използваме резултата от метода а не да го принтираме НЕ използваме Войд метод

public static void Main()

{

int number = sum(1, 7);

Console.WriteLine(number);

}

public static int sum (int first, int second)

{

int result = first + second;

return result;

}

public static void Main()

{

string result = RepeatText("text", 2);

Console.WriteLine(result);

}

public static string RepeatText (string text, int count)

{

string result = string.Empty;

for (int i = 0; i < count; i++)

{

result += text;

}

return result;

}

return приключва целият метод

break приключва само цикъла

събиране на мног на брой тектове. i printiraneto im po byrzo

using System.Text;

string some = "text" + true + 15;

същото е но мого по бързо

StringBuilder result = new StringBuilder();

result.Append("Test");

result.Append(true);

result.Append(15);

string endResult = result.ToString();

да проверим времето за изпълнение на кода

using System.Diagnostics;

Stopwatch sw = new Stopwatch();

sw.Start();

sw.Stop();

sw.Elapsed();

да допиша задачите от последните 30 мин.

Console.WriteLine(endResult);