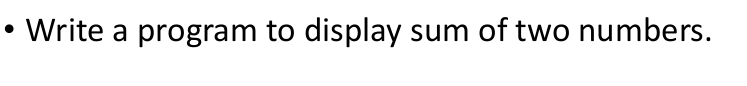
**Name: Nguyễn Thị Hà Giang**

**MSV: 22IT072**

**CÁC VÍ DỤ TRONG CHƯƠNG 2**

**src:** https://github.com/zannie-hg/Bai-tap-Web-nang-cao/tree/main/baitap\_tuan3/exampleC2****

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SumOfTwoNumbers

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Nhap so thu nhat: ");

int num1 = int.Parse(Console.ReadLine());

Console.WriteLine("Nhap so thu hai: ");

int num2 = int.Parse(Console.ReadLine());

int sum = num1 + num2;

Console.WriteLine($"Tong cua {num1} va {num2} la {sum}");

}

}

}

**A screenshot of a computer

AI-generated content may be incorrect.**

**For Loop**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Loop\_For

{

internal class Program

{

static void Main(string[] args)

{

for (int i = 0; i < 100; i++) {

Console.WriteLine("Hello World!");

}

}

}

}

**A screen shot of a computer

AI-generated content may be incorrect.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Loop\_For\_2

{

internal class Program

{

static void Main(string[] args)

{

for (int i = 0; i < 10; i=i+2) {

Console.WriteLine("Value of i: {0}", i);

}

}

}

}

**A screenshot of a computer

AI-generated content may be incorrect.**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace SumOfSeries

{

internal class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.UTF8;

int n, tong = 0;

Console.Write("Nhập vào một số nguyên lớn hơn 0: ");

n = Int32.Parse(Console.ReadLine());

for (int i = 1; i <= n; i++)

{

tong += i;

}

Console.WriteLine("Tổng của dãy số là: {0}", tong);

}

}

}

**A black screen with white text

AI-generated content may be incorrect.**

**Boxing and Unboxing**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace BoxingAndUnboxing

{

internal class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.UTF8;

int num = 10; // Value type

object obj = num; // Boxing: Ép kiểu từ value type sang object

int unboxedNum = (int)obj; // Unboxing: Ép kiểu từ object về int

Console.WriteLine($"Giá trị sau khi unboxing: {unboxedNum}");

}

}

}

**A screenshot of a computer

AI-generated content may be incorrect.**

**Var/Dynamic**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Var\_Dynamic

{

internal class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.UTF8;

var name = "John";

var age = 25;

dynamic data = "Hello";

Console.WriteLine($"Giá trị dynamic: {data}");

data = 100; // dynamic có thể thay đổi kiểu dữ liệu tại runtime

Console.WriteLine($"Giá trị dynamic sau khi thay đổi: {data}");

}

}

}

**A black screen with white text

AI-generated content may be incorrect.**

**String Interpolation**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace String\_Interpolation

{

internal class Program

{

static void Main(string[] args)

{

string name = "Soren";

double salary = 200.234;

// Dùng string.Format()

string str1 = string.Format("Name{0,6}, Salary{1,7:N2}", name, salary);

// Dùng string interpolation

string str2 = $"Name{name,7},Salary{salary,8:N2}";

Console.WriteLine(str1);

Console.WriteLine(str2);

}

}

}

A computer screen with white text

AI-generated content may be incorrect.

**Numeric Literal Syntax**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Numberic\_Literal\_Syntax

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("\*\*\*\*\* Use Digit Separators \*\*\*\*\*");

Console.Write("Integer:");

Console.WriteLine(123\_456); // Tương đương với 123456

Console.Write("Double:");

Console.WriteLine(123\_456.12); // Tương đương với 123456.12

Console.Write("Hex:");

Console.WriteLine(0x\_00\_00\_FF); // Tương đương với 255 trong hệ thập phân

Console.WriteLine("\*\*\*\*\* Use Binary Literals \*\*\*\*\*");

Console.WriteLine("Sixteen: {0}", 0b\_0001\_0000); // Nhị phân 0001 0000 = 16

Console.WriteLine("Thirty Two: {0}", 0b\_0010\_0000); // Nhị phân 0010 0000 = 32

Console.WriteLine("Sixty Four: {0}", 0b\_0100\_0000); // Nhị phân 0100 0000 = 64

Console.ReadLine();

}

}

}

**A black screen with white text

AI-generated content may be incorrect.**

**Ref/ out**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Ref\_Out

{

internal class Program

{

static void Main(string[] args)

{

int a = 5;

Console.WriteLine($"Giá trị ban đầu của a: {a}");

Square(ref a);

Console.WriteLine($"Giá trị của a sau khi gọi hàm Square: {a}");

int result;

GetDoubleValue(out result);

Console.WriteLine($"Giá trị của result sau khi gọi GetDoubleValue: {result}");

}

static void Square(ref int num)

{

num \*= num;

}

static void GetDoubleValue(out int value)

{

value = 10 \* 2; // Bắt buộc phải gán giá trị trước khi return

}

}

}

**A computer screen with white text

AI-generated content may be incorrect.**

**Local Function**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Local\_Function

{

internal class Program

{

static void Main(string[] args)

{

int x = 1;

int y = 2;

// Định nghĩa Local Function (Hàm cục bộ)

void AddValue(int a, int b)

{

Console.WriteLine("Value of a is: " + a);

Console.WriteLine("Value of b is: " + b);

Console.WriteLine("Value of x is: " + x);

Console.WriteLine("Value of y is: " + y);

Console.WriteLine("Sum: {0}", a + b + x + y);

Console.WriteLine();

}

// Gọi hàm cục bộ với các giá trị cụ thể

AddValue(3, 4);

AddValue(5, 6);

Console.ReadLine();

}

}

}

**A screen shot of a computer

AI-generated content may be incorrect.**

**Static Local Function**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Static\_Local\_Function

{

internal class Program

{

static void Main(string[] args)

{

// Hàm cục bộ tính diện tích hình tròn

void AreaOfCircle(double radius)

{

double area = 3.14 \* radius \* radius;

Console.WriteLine("Radius of the circle: " + radius);

Console.WriteLine("Area of circle: " + area);

// Gọi hàm cục bộ tĩnh để tính chu vi

Circumference(radius);

}

// Static Local Function tính chu vi hình tròn

void Circumference(double radius)

{

double circumference = 2 \* 3.14 \* radius;

Console.WriteLine($"Circumference of the circle is: {circumference:N2}");

}

// Gọi hàm tính diện tích với bán kính = 10

AreaOfCircle(10);

Console.ReadLine();

}

}

}

**Tuples**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Tuples

{

internal class Program

{

static (int sum, int count) MyMethod(int[] values)

{

// Khai báo một tuple

var result = (sum: 0, count: 0);

foreach (var value in values)

{

if (IsEvenNumber(value))

{

result.sum += value;

result.count++;

}

}

return result;

}

static bool IsEvenNumber(int n)

{

return n % 2 == 0;

}

static void Main()

{

int[] numbers = { 2, 1, 5, 6, 3, 4, 7, 8, 10, 9 };

var (sum, count) = MyMethod(numbers);

Console.WriteLine($"Sum: {sum}, Count: {count}");

Console.ReadLine();

}

}

}

**A black screen with white text

AI-generated content may be incorrect.**

**Discards with out parameter**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Discard\_With\_Out

{

internal class Program

{

static void Main(string[] args)

{

string[] stringArray = { "12", "Hello", "3.14", "20" };

for (int i = 0; i < stringArray.Length; i++)

{

if (int.TryParse(stringArray[i], out \_))

Console.WriteLine($"{stringArray[i]}: valid");

else

Console.WriteLine($"{stringArray[i]}: invalid");

}

Console.ReadLine();

}

}

}

**A black rectangular object with white dots

AI-generated content may be incorrect.**

**discard with Tuples**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Discard\_With\_Tuples

{

internal class Program

{

static (string first, string middle, string last) SplitNames(string fullName)

{

string[] strArray = fullName.Split(' ');

return (strArray[0], strArray[1], strArray[2]);

}

static void Main(string[] args)

{

var (first, \_, last) = SplitNames("Philip F Japikse");

Console.WriteLine($"{first}:{last}");

Console.ReadLine();

}

}

}

**A black screen with white text

AI-generated content may be incorrect.**

**Pattern Matching with If**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Pattern\_Matching\_If

{

internal class Program

{

static void Main(string[] args)

{

Console.Write("Input data: ");

int.TryParse(Console.ReadLine(), out int n);

if (n is int count && count > 0)

{

Console.WriteLine(new string('\*', count));

}

else

{

Console.WriteLine("Data invalid.");

}

Console.ReadLine();

}

}

}

**A black screen with white text

AI-generated content may be incorrect.**

**Pattern Matching with Switch**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Pattetn\_Matching\_Switch

{

internal class Program

{

static void Main(string[] args)

{

Console.Write("Input data: ");

int.TryParse(Console.ReadLine(), out int n);

switch (n)

{

case int count when count > 0:

Console.WriteLine(new string('\*', count));

break;

default:

Console.WriteLine("Data invalid.");

break;

}

Console.ReadLine();

}

}

}

**A screen shot of a computer

AI-generated content may be incorrect.**