

BÀI 1. TỔNG QUAN NGÔN NGỮ LẬP TRÌNH C#

A. MỤC TIÊU:

- Xử lý các thao tác cơ bản trong ngôn ngữ C#;
- Sử dụng được các cấu trúc điều khiển trong C#;
- Sử dụng các vòng lặp thực hiện các bài toán cơ bản;
- Xử lý các ngoại lệ phát sinh.

B. NỘI DUNG THỰC HÀNH

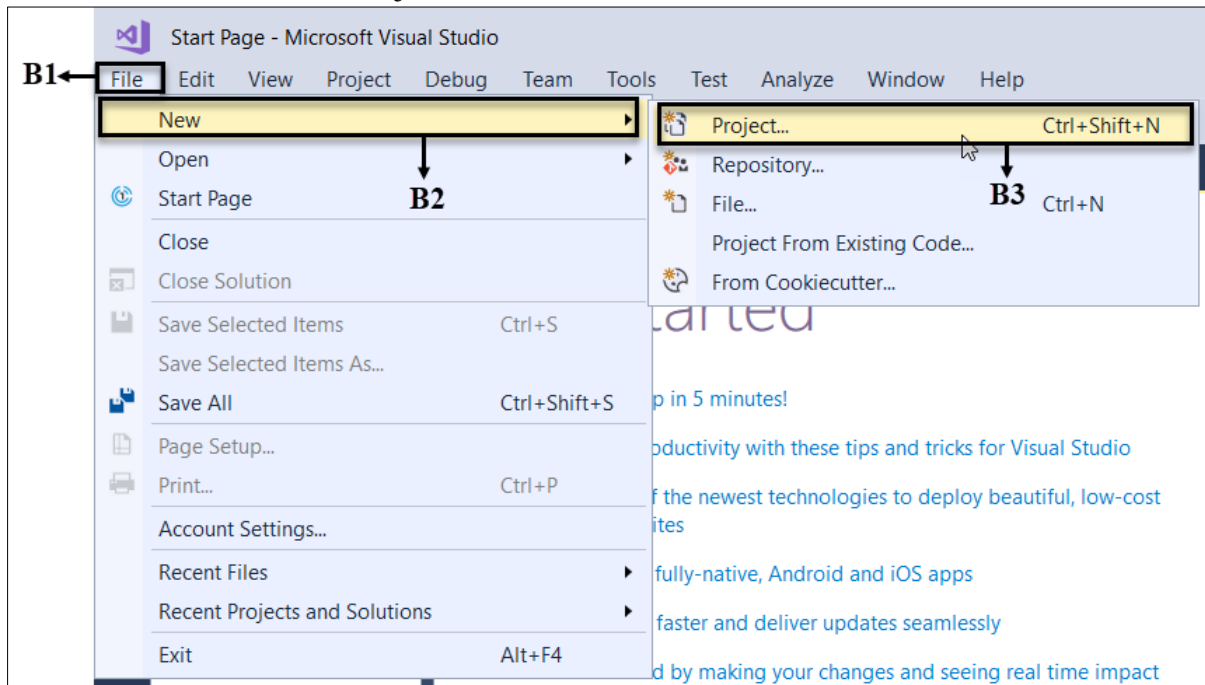
1. Cơ sở lý thuyết

1.1. Kiến thức cần nhớ

- Các cấu trúc điều khiển: if, if ...else, switch...
- Các câu lệnh vòng lặp: for(...;...;...), while, do...while
- Tạo một project mới :

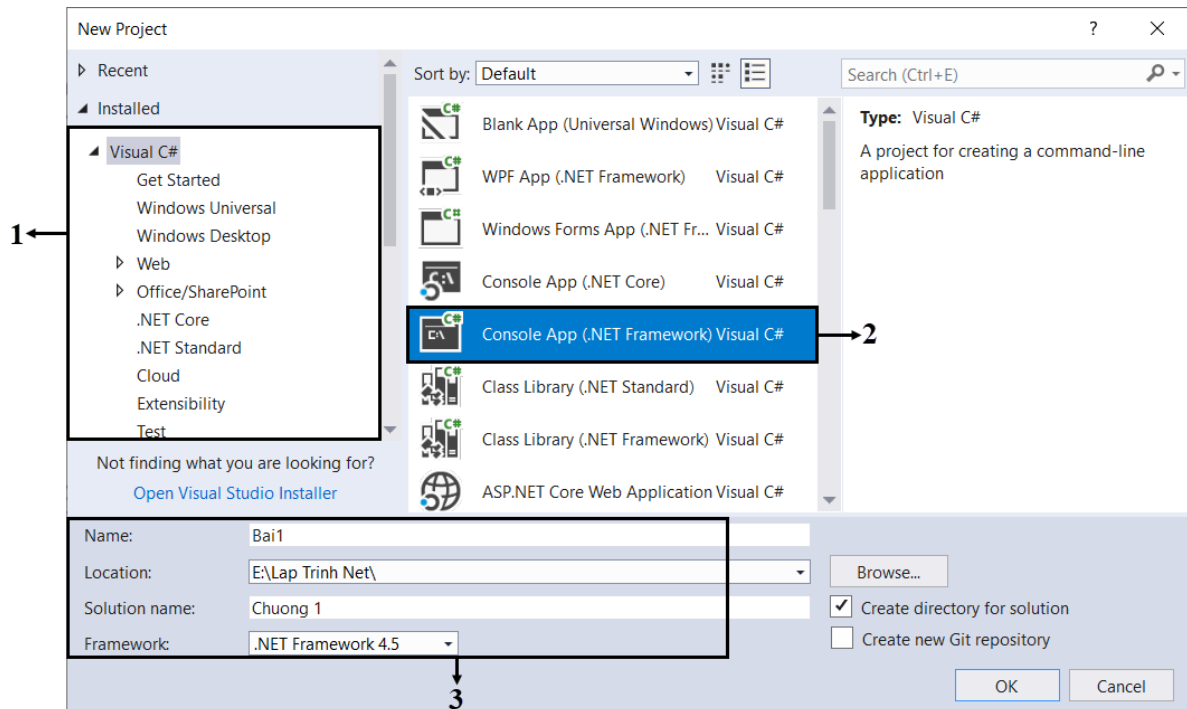
Bước 1: Khởi động Microsoft Visual Studio 2017. Bước

2: Chọn File -> New -> Project:



Hình 1. 1. Tạo project mới

Thiết lập thông tin Project trong hộp thoại như **Hình 1.2**:



Hình 1. 2. Thiết lập thông tin project

Trong đó:

[1]: Nơi chọn ngôn ngữ lập trình (trong hình chọn Visual C#) [2]:

Chọn loại project (trong hình chọn Console App .NET Framework)

[3]: Thông tin project bao gồm: **Name**: Tên Project

Location: Vị trí lưu Project

Solution name: Tên của Solution

Framework: Chọn phiên bản .Net. Tùy vào phiên bản mà ta sẽ có các thư viện hỗ trợ.

Sau khi nhấn OK (đồng ý) sẽ có một hộp thoại hiện ra màn hình như Hình 1.3.

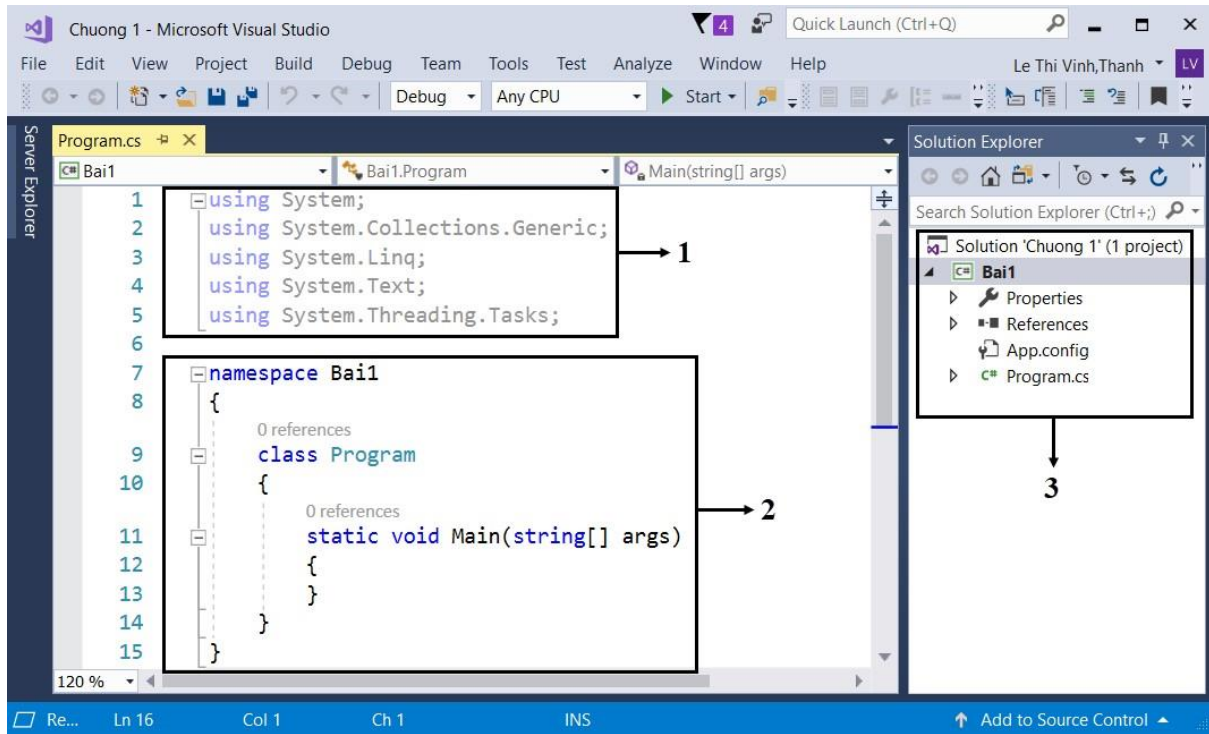
Mặc định Visual Studio sẽ mở file Program.cs.

Trong đó:

[1]: Phần khai báo các thư viện thông qua từ khóa using

[2]: Phần code mà bạn sẽ gõ vào

[3]: Đây là nơi quản lý các file dự án, hình, data,



Hình 1. 3. Giao diện thực hiện chương trình project

Bước 3: Để hiển thị đoạn văn bản lên màn hình chúng ta có đoạn code như sau:

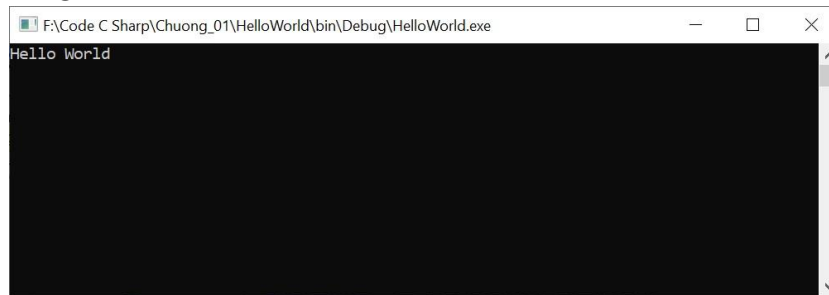
```

using System; using
System.Collections.Generic;
using System.Linq; using
System.Text; using
System.Threading.Tasks; //Đây là
tên của project namespace
HelloWorld
{
    //Đây là tên của một class
class Program
    {
        //Là một hàm
        static void Main(string[] args)
        {
            //Code của bạn tại đây
            //Đây là code hiển thị lên màn hình
            Console.WriteLine("Hello World");
            Console.ReadLine();
        }
    }
}

```

Sau khi nhập nội dung đoạn code trên nhấn **Run (F5)** để chạy Project.

Kết quả chương trình xuất ra đoạn văn bản như hình 1.4:



Hình 1. 4. Kết quả chạy chương trình

1.2. Giới thiệu bài tập mẫu

Bài tập 1. Viết chương trình nhập vào số nguyên, kiểm tra số đã nhập là số âm hay số dương.

Bước 1: Xử lý nhập vào số nguyên trong hàm Main của file Program.cs:

```

using System; using
System.Collections.Generic;
using System.Linq; using
System.Text; using
System.Threading.Tasks;
namespace Demo
{
    class
    Program
    {
        static void
        Main(string[] args)
        {
            //khai báo biến số nguyên n
            int n;
            //khai báo chuỗi s
            string s
            //Thông báo nhập n
            Console.WriteLine("Nhập vào số nguyên n= "); //Đọc
            dữ liệu nhập từ bàn phím vào một chuỗi s s =
            Console.ReadLine();
            //Chuyển chuỗi thành s số nguyên và gán giá trị cho biến n
            n = int.Parse(s);
        }
    }
}

```

Hoặc có thể xử lý ngắn gọn như đoạn code sau:

```

using System; using
System.Collections.Generic;
using System.Linq; using
System.Text; using
System.Threading.Tasks;
namespace Demo

```

```

{
class Program
{
    static void Main(string[] args)
    {
        //Khai báo biến số nguyên n
        int n;
        //Thông báo nhập n
        Console.Write("Nhập vào số nguyên n=");
        //Đọc dữ liệu nhập từ bàn phím vào một chuỗi
        //Chuyển chuỗi thành số nguyên và gán giá trị cho biến n
        n = int.Parse(Console.ReadLine());
    }
}
}

```

Bước 2: Kiểm tra số đã nhập là số âm hay số dương bằng câu lệnh điều khiển **if** như đoạn code sau:

```

static void Main(string[] args)
{
    int n; //Khai báo biến số nguyên n
    //Thông báo nhập n
    Console.Write("Nhập vào số nguyên n=");
    n = int.Parse(Console.ReadLine()); //sử
    dụng cấu trúc điều kiện if
    if (n == 0)
    {
        Console.Write("Số không âm không dương");
    }
    if (n > 0)
    {
        Console.Write("Số dương");
    }
    if (n < 0)
    {
        Console.Write("Số âm");
    }

    Console.ReadLine();
}

```

Ngoài ra, chúng ta có thể sử dụng cấu trúc điều khiển đầy đủ của **if-else** như sau:

```
using System; using
System.Collections.Generic;
using System.Linq; using
System.Text;
```

```
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            int n; //khai báo biến số nguyên n
            //Thông báo nhập n
            Console.Write("Nhập vào số nguyên n =");
            n = int.Parse(Console.ReadLine());
            //Sử dụng cấu trúc điều kiện if-else
            if (n == 0)
            {
                Console.Write("Số không âm không dương");
            }
            else
            {
                if (n > 0)
                {
                    Console.Write("Số dương");
                }
                else
                {
                    Console.Write("Số âm");
                }
            }
            Console.ReadLine();
        }
    }
}
```

Hướng dẫn sử dụng *try...catch*

Khi thực hiện một lệnh hoặc đoạn lệnh, chương trình có khả năng phát sinh lỗi. Vì vậy khi viết code chúng ta nên dùng mệnh đề ***try...catch*** để tránh phát sinh lỗi không mong muốn khi chương trình đang được thực hiện.

Ví dụ, khi chương trình yêu cầu nhập vào số nguyên, người dùng có thể vô tình nhập một ký tự không phải là số khi đó việc thực hiện câu lệnh:

```
n = int.Parse(Console.ReadLine());
```

sẽ xảy ra lỗi. Lúc này chúng ta có thể sử dụng mệnh đề ***try...catch*** với cú pháp như sau:

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



```
using System.Text; using
System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            int n; // khai báo biến số nguyên n
            // Thông báo nhập n
            Console.Write("Nhập vào số nguyên n=");
            try
            {
                // Chuyển chuỗi nhập vào thành số
                n = int.Parse(Console.ReadLine());
            }
            catch
            {
                Console.Write("Bạn nhập n không phải số");
                return;
            }
            // Sử dụng cấu trúc điều kiện if
            if (n == 0)
            {
                Console.Write("Không âm không dương");
            }
            else
            {
                if (n > 0)
                {
                    Console.Write("Số dương");
                }
                else
                {
                    Console.Write("Số âm");
                }
            }
            Console.ReadLine();
        }
    }
}
```

Trong ví dụ này, nếu người dùng vô tình nhập một ký tự không phải là số thì những lệnh trong *catch*{ } sẽ được thực thi. Kết quả ngoài màn hình sẽ hiển thị đoạn văn bản báo lỗi khi người dùng nhập dữ liệu không phải là số.

Bài tập 2. Viết chương trình tính tích $S = 1*2*3*4*5*...*n$, trong đó n là số nguyên nhập từ phím. Bộ giá trị kiểm tra:

$n = 1: S1 = 1;$ $n = 2: S1 = 2;$ $n = 3:$

$S1 = 6$ Bước 1: Xử lý nhập vào số nguyên:

```
using System; using
System.Collections.Generic;
using System.Linq; using
System.Text; using
System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            // Khai báo biến số nguyên n
            int n;
            // Thông báo nhập n
            Console.Write("Nhập vào số nguyên n=");
            // Đọc dữ liệu nhập từ bàn phím vào chuỗi s
            // Chuyển chuỗi nhập vào thành số
            n = int.Parse(Console.ReadLine());
        }
    }
}
```

Bước 2: Xác định biểu thức 1, biểu thức 2, biểu thức 3 cho vòng lặp *for*. Lưu ý có thể sử dụng các vòng lặp khác để thay thế vòng lặp *for*. Biểu thức 1: `int i = 1;` // khởi tạo gán giá trị

Biểu thức 2: `i <= n;` // điều kiện dừng

Biểu thức 3: `i++;` // tăng bước nhảy

Công việc cần làm là tính tích giá trị từ 1 đến n . Khai báo khởi tạo biến lưu giá trị tích.

Bước 3: Code xử lý theo yêu cầu đề bài:

```

using System;
using System.Collections.Generic;
using System.Linq; using
System.Text; using
System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            //Khai báo biến số nguyên n
            int n;
            //Khởi tạo giá trị cho biến tích
            int tich = 1;
            //Thông báo nhập n
            Console.Write("Nhập vào số nguyên n=");
            //Nhập n từ bàn phím
            try
            {
                //Chuyển chuỗi nhập vào thành số
                n = int.Parse(Console.ReadLine());
            }
            catch
            {
                Console.Write("Bạn nhập n không phải số");
                return;
            }
            //Xử lý tính tích
            for (int i = 1; i <= n; i++)
            {
                tich = tich * i;
            }
            //In giá trị ra màn hình theo 1 trong 2 cách:
            //Console.Write("Tích là: {0}",tich);
            Console.Write("Tích là: " + tich);
            Console.ReadLine();
        }
    }
}

```

Chúng ta có thể chuyển ví dụ trên thành dạng phương thức gọi khi cần. Lưu ý, có thể đặt phương thức phía trên hoặc phía dưới `void main()` mà không cần phải khai báo prototype (nguyên mẫu hàm).

```
using System; using
System.Collections.Generic;
using System.Linq; using
System.Text; using
System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            // Gọi phương thức BaiMau()
            BaiMau();
            Console.ReadLine();
        }
    }
}
```

```

        //Phương thức BaiMau() được định nghĩa như sau
static void BaiMau()
{
    //khai báo biến số nguyên n
    int n;
    //Khởi tạo giá trị cho biến tích
    int tich = 1;
    //Thông báo nhập n
    Console.Write("Nhập vào số nguyên n=");
    //Nhập từ bàn phím
    try
    {
        //Chuyển chuỗi nhập vào thành số
        n = int.Parse(Console.ReadLine());
    }
    catch
    {
        Console.Write("Bạn nhập n không phải số");
        return;
    }
    //Xử lý tính tích
    for (int i = 1; i <= n; i++)
    {
        tich = tich * i;
    }
    //In giá trị ra màn hình
    Console.Write("Tích là: {0}", tich);
    //Console.Write("Tích là: "+ tich);
}
}
}

```

Bài tập 3. Viết chương trình tìm ước chung lớn nhất (UCLN) của hai số nguyên dương a và b.

Bước 1: Xử lý nhập vào 2 số nguyên dương a và b:

```

using System; using
System.Collections.Generic;
using System.Linq; using
System.Text;
using System.Threading.Tasks;
namespace Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            //khai báo
            int a, b;
            //Nhập a, b Console.Write("Mời nhập a: ");
            a = int.Parse(Console.ReadLine());
            Console.Write("Mời nhập b: ");
            b = int.Parse(Console.ReadLine());
            Console.ReadLine();
        }
    }
}

```

Bước 2: Xác định các công việc cần làm

Công việc phải làm là tìm ước chung lớn nhất của hai số nguyên dương a và b.

- Sử dụng vòng lặp while kiểm tra a khác b hay không?

Kiểm tra a lớn hơn b hay không?

+ Nếu đúng thực hiện: $a = a - b$;

+ Nếu sai thực hiện: $b = b - a$;

- Nếu a bằng b thoát khỏi vòng lặp và trả về giá trị a chính là ước chung lớn nhất.

Bước 3: Công việc cần làm là tìm UCLN của 2 số a và b, thực hiện như sau:

```

using System; using System.Collections.Generic; using
System.Linq; using System.Text; using
System.Threading.Tasks; namespace Demo
{   class Program
    {
        static void Main(string[] args)
        {
            //khai báo
            int a, b;
            //Nhập a, b
            do
            {
                Console.Write("Mời nhập a: ");
                a = int.Parse(Console.ReadLine());
            }while(a <= 0);
            do
            {
                Console.Write("Mời nhập b: ");
                b = int.Parse(Console.ReadLine());
            }while (b <=0);
            while (a != b)
            {
                if (a > b)
                {
                    a = a - b;
                }
                else
                {
                    b = b - a;
                }
            }
            //In kết quả
            Console.Write("UCLN là: {0}", a);
            Console.ReadLine();
        }
    }
}

```

Viết dạng phương thức bài trên chúng ta làm như sau:

```
static int UCLN(int a, int b)
{
    while (a != b) //bt2
    {
        if (a > b)
        {
            a = a - b; //bt2
        }
        else
        {
            b = b - a; //bt2
        }
    }
    return a;
}
```

2. Bài tập tại lớp

Bài tập 1. Viết chương trình nhập vào đơn giá một mặt hàng và số lượng bán của mặt hàng đó. Tính tiền khách phải trả với thông tin như sau:

Thành tiền = đơn giá * số lượng.

Giảm giá: Nếu thành tiền >100, thì giảm 3% thành tiền, ngược lại không giảm.

Tổng tiền phải trả = thành tiền – giảm giá.

Hướng dẫn:

Bước 1: Tạo Project mới có tên Bai_01.

Bước 2: Trong file Program.cs xây dựng các như yêu cầu đề bài.

Bước 3: Tạo các biến theo yêu cầu đề bài.

```
//Thực hiện khai báo các biến cho chương trình
double donGia; int soLuong;
//Tạo một biến thành tiền và khởi tạo nó là 0
double thanhTien = 0;
```

Bước 4: Thực hiện thông báo người dùng nhập giá trị và thực hiện tính thành tiền tạm thời:


```
//Thông báo người dùng hãy nhập vào đơn giá
Console.WriteLine("Don gia san pham la: ");
//Sử dụng method (hàm-phương thức) Parse để chuyển kiểu dữ liệu từ chuỗi
sang số
//Tại đây chuyển sang kiểu double-số thực
donGia = double.Parse(Console.ReadLine());
Console.WriteLine("So luong mua: ");
//Tại đây chuyển thành kiểu số nguyên
soLuong= int.Parse(Console.ReadLine());
//Thực hiện tính thành tiền tạm thời
thanhTien = donGia * soLuong;
```

Bước 5. Tại bài này ta chỉ cần sử dụng cấu trúc *if* xử lý yêu cầu đề bài:

```
//Nếu thành tiền tạm thời lớn hơn 100 thực hiện cập nhật lại thành tiền;
Ngược lại chỉ in thành tiền
if (thanhTien>100)
{
    thanhTien = thanhTien - thanhTien * 0.03;
}
Console.WriteLine("Thanh tien cua ban se la: " + thanhTien);
//Dừng màn hình xem kết quả
Console.ReadLine();
```

Bài tập 2. Viết chương trình tính tiền điện phải trả trong tháng:

Từ 1 – 100 KW: 5\$

Từ 101 – 150 KW: 7\$

Từ 151 – 200 KW: 10\$

Từ 201 – 300 KW: 15\$

Từ 301 KW trở lên: 20\$

Ví dụ: Nếu điện tiêu thụ 50 KW thì tiền điện phải trả là: $50 \times 5 = 250\$$.

Nếu điện tiêu thụ 101 KW thì tiền điện phải trả là: $100 \times 5 + 1 \times 7 = 507\$$.

Hướng dẫn: Sử dụng cấu trúc *if else* để giải quyết bài toán trên:

Bước 1: Khai báo thuộc tính và cho phép người dùng nhập vào hệ thống số điện tiêu thụ:

```
//Khai báo và thực hiện cho người dùng nhập vào số điện tiêu
thụ int soDienTieuThu; double thanhTien = 0;
Console.Write("Moi nhap so luong dien tieu thu: ");
soDienTieuThu = int.Parse(Console.ReadLine());
```

Bước 2: Thực hiện yêu cầu bằng cấu trúc *if else*:

```

//Sử dụng câu lệnh điều khiển if else
//Tại đây phân tích công thức
//Nếu số điện chỉ dưới 100 chỉ cần lấy số_điện_tiêu_thụ * 5-->Thành tiền
//Nếu lớn hơn 100 thực hiện công thức: 100*5+(số điện tiêu thụ 100)*(Đơn
giá)
//Nếu lớn hơn 150 thực hiện công thức: 100 * 5 + 50 * 7+
(soDientieuThu-150) * (Đơn giá);
if (soDientieuThu <= 100)
{
    thanhTien += soDientieuThu * 5;
}
else if (soDientieuThu <=150)
{
    thanhTien += 100 * 5 + (soDientieuThu- 100) * 7;
}
else if (soDientieuThu <=200)
{
    thanhTien += 100 * 5 + 50 * 7+ (soDientieuThu-150) * 10;
}
else if (soDientieuThu <= 300)
{
    thanhTien += 100 * 5 + 50 * 7 + 50 * 10 + (soDientieuThu-200) * 15;
}
else
{
    //Nếu vượt kế hoạch (300 kwh) thực hiện như sau:
    thanhTien += 100 * 5 + 50 * 7 + 50 * 10 + 100 * 15 +
        (soDientieuThu- 300) * 20;
}
//Thực hiện in kết quả ra
Console.WriteLine("So tien ban phai tra la: " + thanhTien);
//Dừng màn hình xem kết quả
Console.ReadLine();

```

Bài tập 3. Viết chương trình in bảng cửu chương từ 1 đến 9 theo hàng dọc.

Hướng dẫn: Sử dụng cấu trúc lặp *for* hoặc *while* hoặc *do while* để giải quyết bài toán trên.

```

//Cấu trúc bản cửu chương như sau:
// 1 x 1 = 1
// 1 x 2 = 2
//....
// 1 x 10 = 10
Console.WriteLine("Bang cuu chuong 9 theo chieu doc co dang:\n");
//A X B= C
//Trong đó:
//A, B: Thừa số
//C: tích của A X B
//Sử dụng cấu trúc vòng lặp for lồng vào nhau:
//Vì in bảng cửu chương theo chiều dọc từ 1 đến 9
//nên vòng for thứ 1 điều kiện dừng lại là 9
//Thừa số B chạy từ 1-->10 nên vòng for thứ 2 điều kiện dừng lại là 10
for (int i = 1; i <= 9; i++)
{
    for (int j = 1; j <=10; j++)
    {
        //Thực hiện in ra từng dòng A x B = C
        //0,1,2 là vị trí biến phía sau vị trí
        Console.WriteLine("{0} x {1} = {2}", i, j, i * j);
    }
    //Tại đây sau khi kết thúc một bảng cửu chương thì xuống dòng
    // Giúp ngăn cách các bảng cửu chương
    //Khuyến nghị nên thêm vào
    Console.WriteLine();
}
//Dừng màn hình--Nhấn phím bất kì để thoát
Console.ReadKey();

```

Bài tập 4. Viết chương trình in bảng cửu chương từ 1 đến 9 theo hàng ngang.

Bài tập 5. Viết chương trình vẽ hình chữ nhật có kích thước $d \times r$, trong đó d là chiều dài và r là chiều rộng được nhập từ phím.

```

*           * * * *
*           * * * *
*           * * * *

```

Hướng dẫn: Sử dụng vòng lặp *for* để giải quyết bài trên:

```

//Khởi tạo và nhập bàn phím int
dong, cot;
Console.WriteLine("Nhap so dong: ");

```

```

dong = int.Parse(Console.ReadLine());
Console.WriteLine("Nhập số cột: ");
cot = int.Parse(Console.ReadLine());
//Sử dụng vòng for in ra từng dòng
for (int i = 0; i < dong; i++)
{
    for (int j = 0; j < cot; j++)
    {
        //In liên tiếp các kí tự '*' kèm dấu tab (ngăn cách)
        Console.Write("*\t");
    }

    //Sau mỗi dòng thực hiện xuống hàng
    Console.WriteLine();
}
Console.ReadKey();

```

Bài tập 6. Viết chương trình vẽ hình chữ nhật có kích thước $d \times r$, trong đó d là chiều dài và r là chiều rộng được nhập từ phím.

```

*           * * * *
*           *
*           *
*           * * * *

```

Hướng dẫn: Cũng tương tự sử dụng vòng lặp *for* tuy nhiên xét đến điều kiện khi nào nó là khoảng trắng:

Ta xét ví dụ trên có 4 dòng và 5 cột:

+Ta thấy dòng đầu (1) và dòng cuối (4) có đặc điểm là giống nhau và được in ra tất cả;

+Dòng giữa 2 và 3 khi qua cột thứ 2 bắt đầu thay kí tự “*” thành “ ”; Từ việc phân tích trên ta có thể thiết lập thuật toán như sau:

```

int dong, cot;
Console.WriteLine("Nhap so dong: ");
dong = int.Parse(Console.ReadLine());
Console.WriteLine("Nhap so cot: ");
cot = int.Parse(Console.ReadLine());
//Sử dụng vòng lặp for in ra từng dòng
for (int i = 0; i < dong; i++)
{
    for (int j = 0; j < cot; j++)
    {
        //In liên tiếp các kí tự '*' kèm dấu tab (ngăn cách)
        //Theo cách phân tích ta sẽ lọc ra khi nào in dấu *
        //Tại đây giá trị đếm bắt đầu là 0 nên dòng 1 --> dòng 0
        //dòng cuối sẽ trừ đi một đơn vị: dòng-1 -->dòng cuối
        //Nếu dòng(i) 0 hoặc dòng cuối(dong-1)
        //Nếu cột(j) đầu tiên hoặc cột cuối(cot -1)
        //Cho tất cả 2 nếu vừa nêu vào if và thực hiện in '*'
        if (i == 0 || j == 0 || j == cot - 1 || i == dong - 1)
            Console.Write("*\t");
        else
            //Ngược lại với điều kiện trên in ' '-khoảng trắng
            Console.Write(" \t");
    }
    //Sau mỗi dòng thực hiện xuống hàng
    Console.WriteLine();
}
Console.ReadKey();

```

Bài tập 7. Viết chương trình tính tổng: $1^2 + 2^2 + \dots + n^2$ với n là số nguyên dương được nhập từ bàn phím.

Bộ giá trị kiểm tra: n = 1: $S_1 = 1$; n = 2: $S_1 = 5$; n = 3: $S_1 = 14$

Bước 1: Nhập số nguyên dương n từ bàn phím, kiểm tra số đã nhập sử dụng *try...catch*.

```
//Tạo một biến tổng là số thực
double sum = 0; int n;
do
{
Console.WriteLine("Nhập vào số nguyên dương n: ");
}while (n <= 0);
try
{
    n = int.Parse(Console.ReadLine());
}
catch (Exception ex)
{
    Console.WriteLine("Lỗi, bạn vừa nhập kí tự không phải số!!!");
    Console.ReadLine();
    return;
}
}
```

Bước 2: Sử dụng vòng lặp *for* để duyệt từng giá trị từ 1 đến n.

```
for (int i = 1; i <= n; i++)
{
    //Sử dụng method Pow có sẵn của class Math
    //Tham số đầu tiên là giá trị n chạy từ 1 đến n
    //Tham số thứ hai là lũy thừa
    sum += Math.Pow(i, 2);
    //Hoặc sum+= i*i
}
Console.WriteLine("Tổng S= "+sum);
Console.ReadLine();
```

Bài tập 8. Kiểm tra số nguyên n có phải là số nguyên tố hay không?

Hướng dẫn:

Số nguyên tố là số tự nhiên lớn hơn 1 chỉ có hai ước số dương phân biệt là 1 và chính nó. Các số có nhiều hơn 2 ước số dương được gọi là hợp số.

Lưu ý: Do số 1 chỉ có một (1) ước số dương là chính nó, nên số 1 không phải là số nguyên tố và cũng không phải là hợp số.

Từ hướng dẫn trên ta xây dựng một method (hàm - phương thức) kiểm tra ước số trả về *true/false*.

Bước 1: Xây dựng phương thức như sau:

Cách 1:

```
static bool checkSNT(int n)
{
    //Tạo một biến đếm count
    int count = 0;
    for (int i = 1; i <= n; i++)
    {
        //Nếu như n chia hết cho i (% - toán tử chia lấy dư -
        //nếu dư = 0--> chia hết)
        if (n%i==0)
        {
            count++;
        }
    }
    if (count>2)
    {
        return false;
    }
    else
    {
        return true;
    }
}
```

Cách 2:

```
static bool checkSNT(int n)
{
    //Đầu tiên kiểm tra nếu vi phạm điều kiện trong phần lưu ý thực hiện
    //trả về false
    if (n <= 1)
```

```

{
    return false;
}

//Tạo một biến count (đếm) các ước số và khởi tạo cho nó bằng 0
    int count = 0;
//Vì mọi số từ 2 trở lên đều chia hết cho 1
//Giá trị khởi đầu là 2
for (int i = 2; i <= n; i++)
{
    //Nếu như n chia hết (%-toán tử chia lấy dư-dư 0-->hết)
        if (n%i == 0)
        {
            //Tăng biến count lên một đơn vị
            count++;
        }
        if (count == 2)
        {
            //Nếu có 2 ước số thực hiện trả về false và ngừng kiểm tra
            return false;
        }
    }
return true;
}

```

Bước 2: Trong hàm main thực hiện khởi tạo và kiểm tra giá trị:

```

//Viết rõ ràng tường minh
//if (checkSNT(n)==true)
//{

//}

//Viết ngắn gọn
//Nếu kiểm tra trả về true thì in kết quả đây là số nguyên tố
if (checkSNT(n))
{
    Console.WriteLine("Số " + n + " là số nguyên tố");
}
else
{
    Console.WriteLine("Số " + n + " không phải là số nguyên tố");
}
Console.ReadKey();

```


Bài tập 9. Liệt kê các số nguyên tố nhỏ hơn n , với n là một số nguyên dương được nhập từ bàn phím.

Hướng dẫn:

Bước 1: Tạo một hàm kiểm tra số nguyên tố hay không? (Sử dụng lại hàm bài trên).

Bước 2: Cho người dùng nhập một số nguyên dương n từ bàn phím sử dụng bắt lỗi *try catch*.

Bước 3: Trong hàm Main thực hiện thêm vòng *for* để in ra số nguyên tố nhỏ hơn n :

```
//Các số nguyên tố từ 2 đến n là:
Console.WriteLine("Cac so nguyen to < {0} là: \n", n);
for (int i = 2; i < n; i++)
{
    //Kiểm tra các số nhỏ hơn n và là số nguyên tố
    //Thực hiện in ra màn hình
    if (checkSNT(i))
    {
        Console.Write("{0};", i);
    }
}
Console.ReadKey();
```

Bài tập 10. Viết chương trình nhập một dãy số nguyên (nếu nhập số 0 thì dừng).

Tìm tổng dãy số vừa nhập.

Tìm giá trị lớn nhất của dãy số đó.

Tìm giá trị nhỏ nhất của dãy số đó.

Hướng dẫn: Sử dụng vòng lặp *while* vô tận cách xử lý như sau:

Bước 1: Tạo và khởi tạo giá trị ban đầu cho các biến sau:

```
//Tạo ra 4 biến lưu giá
trị int sum = 0; int max
= 0; int min = 0; int
temp;
```

Bước 2: Thực hiện vòng lặp *while* hoặc *do... while.....* Để kiểm tra min, max và tính tổng:

```

do
{
    Console.Write("Nhập số n= ");
    temp = int.Parse(Console.ReadLine());
    if (temp == 0)
    {
        //Thực hiện dừng vòng lặp và trả ra kết quả
        break;
    }
    if (max==0||min==0)
    {
        max = temp;
        min = temp;
    }
    //Thực hiện cộng dồn các số lại để tính tổng dãy số
    sum += temp;
    //Nếu temp đúng gán max = temp
    if (temp >= max)
    {
        max = temp;
    }
    //Nếu temp đúng gán min = temp
    if (temp <= min)
    {
        min = temp;
    }
} while (true);

```

Bước 3: Thực hiện in kết quả ra màn hình:

```

//In kết quả ra màn hình
Console.WriteLine("Tổng là: " + sum);
Console.WriteLine("Max là: " + max);
Console.WriteLine("Min là: " + min); Console.ReadLine();

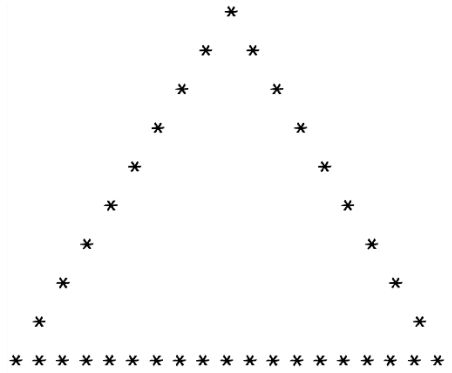
```

3. Bài tập nâng cao

Bài tập 1. Viết chương trình giải phương trình bậc 2 có dạng: $ax^2 + bx + c = 0$, với a, b, c là các hệ số thực, a khác 0.

Bài tập 2. In tam giác cân rỗng.

Tam giác cân là tam giác có 2 cạnh bên bằng nhau. Tam giác cân rỗng như hình bên dưới:



4. Bài tập về nhà

Bài tập 1. Viết chương trình xuất số có 2 chữ số sao cho các chữ số khác nhau đôi một.

Ví dụ: Danh sách các số có 2 chữ số đôi một khác nhau là: 10, 12, 13, ... , 20, 21, 23, ..., 98

Bài tập 2. Viết chương trình in ra tất cả các số lẻ nhỏ hơn n , trong đó n nhập từ bàn phím.

Bài tập 3. Viết chương trình đếm số lượng số chẵn trong dãy từ $[n, m]$, trong đó n, m nhập từ bàn phím.