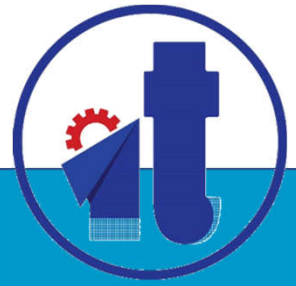
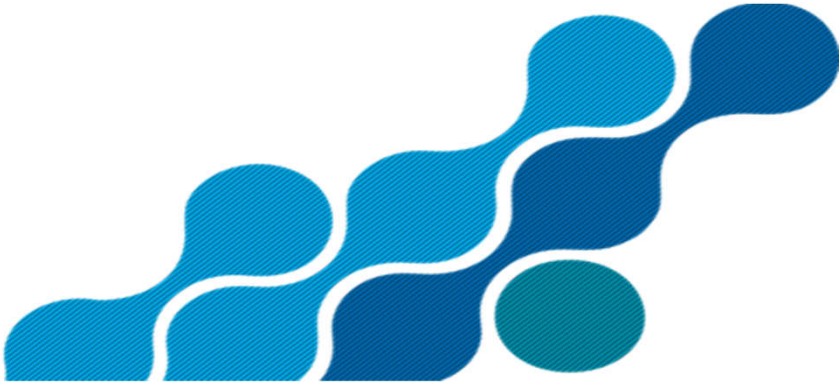
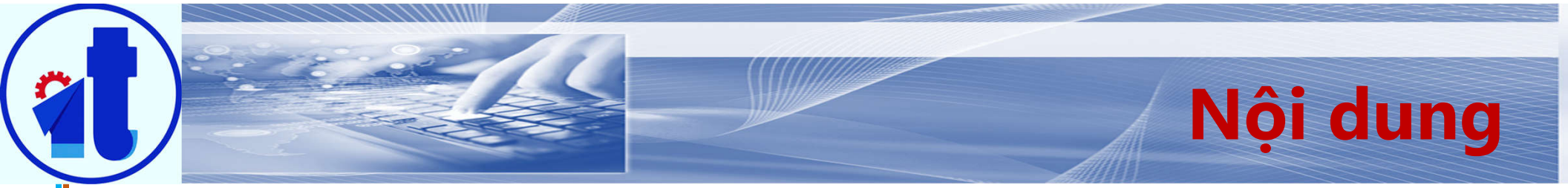

CAO ĐẲNG KỸ THUẬT
CAO THẮNG
LẬP TRÌNH TRÊN MÔI TRƯỜNG WINDOWS



LVHN

Chương 2: Giới Thiệu Ngôn Ngữ Lập Trình C#





Nội dung

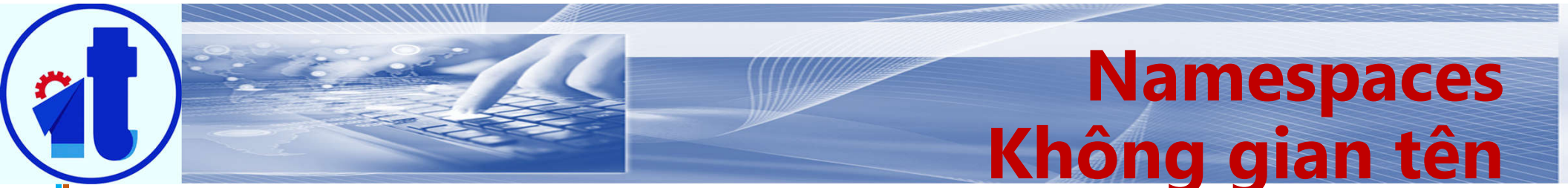
- Kiểu dữ liệu
- Các lệnh và phép toán cơ bản
- Các cấu trúc điều khiển
- Mảng
- Phương thức - hàm

Cấu trúc chương trình cơ bản

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace KhongGianTen
8   {
        0 references
9       class Program
10      {
            0 references
11          static void Main(string[] args)
12          {
13
14              //Chương trình
15          }
16      }
17  }
```

Không gian tên

Chương trình thực thi

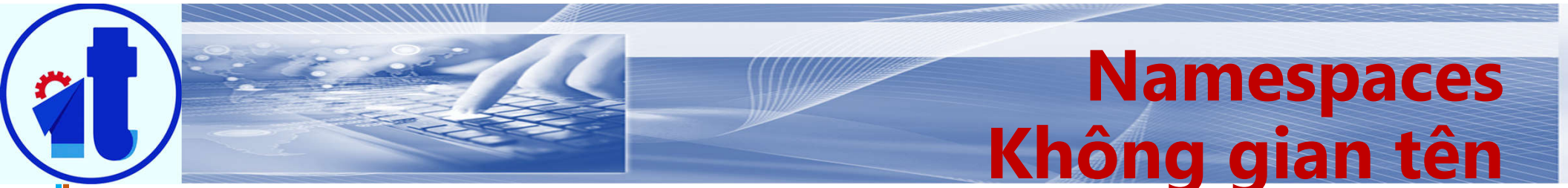


Namespaces
Không gian tên

- Namespaces là một nhóm các lớp/ thư viện có liên quan
- Một namespace có thể chứa lớp/thư viện và namespace con khác
  - Ví dụ: System.Data chứa Sql, OleDb,…
- Hai cách sử dụng namespace
  - Sử dụng chức năng/ hàm/ phương thức trực tiếp
    - ✓ Ví dụ: System.Console.WriteLine("Hello");
  - Sử dụng toán tử using (giống include trong C/C++)
    - ✓ Ví dụ: using System;

```
.....
Console.WriteLine("Hello");
```

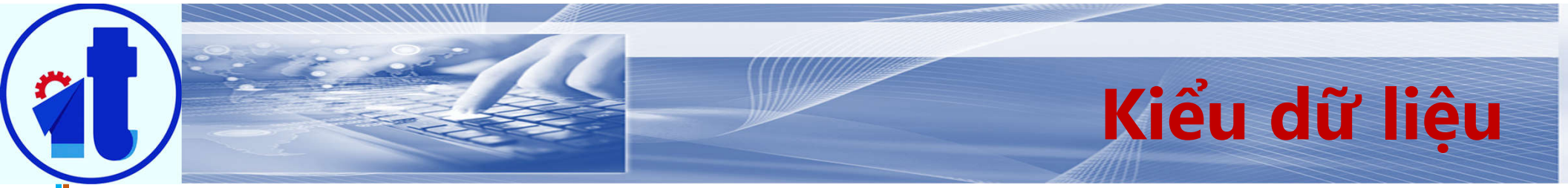
- Mỗi project nên khai báo namespace để thuận tiện quản lý tầm vực của các lớp/ phương thức – dự án lớn



Namespaces
Không gian tên

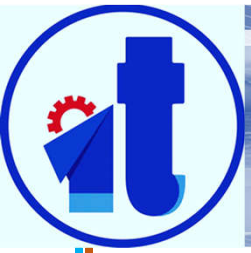
LVHN

- Tính chất:
  - Tổ chức code dự án lớn
  - Phân định bởi toán tử "."
    - ✓ Chị định lớp/namespace khác: System.Data, System.Console,...
    - ✓ Truy xuất dữ liệu phương thức, thuộc tính: System.Convert.ToDouble("12.3") ....
  - Dùng toán tử using để khai báo sử dụng namespace
  - Không gian tên System là không gian tên toàn cục (global::System) chứa nhiều lớp/thư viện giao tiếp với hệ thống và cung cấp các hàm/chức năng thông dụng



Kiểu dữ liệu

- Hai kiểu dữ liệu
  - Kiểu dữ liệu được xây dựng sẵn (build-in): do ngôn ngữ lập trình cung cấp cho người lập trình
  - Kiểu dữ liệu do người dùng định nghĩa (user-defined): người lập trình định nghĩa tùy theo yêu cầu bài toán

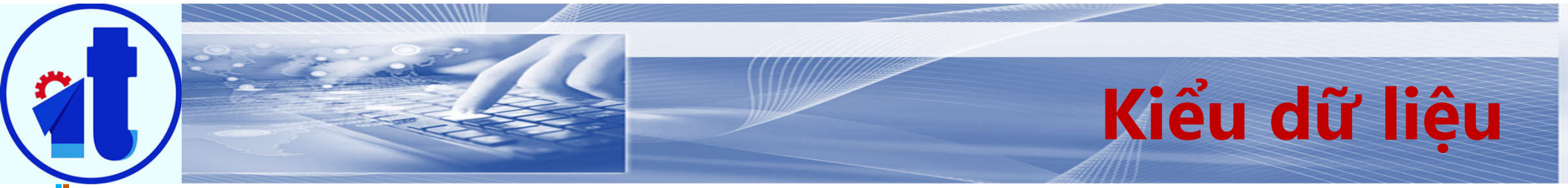


Kiểu dữ liệu

- Một số kiểu dữ liệu được xây dựng sẵn

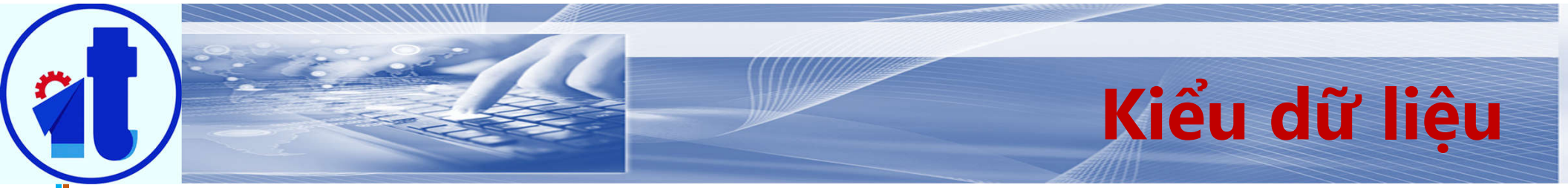
KIỂU DỮ LIỆU	KÍCH THƯỚC (byte)	MIỀN GIÁ TRỊ	GIÁ TRỊ MẶC ĐỊNH
bool	1	true hoặc false	false
byte	1	0..255	0
char	2	'\u0000' (null)	Mã ký tự unicode
float	4	$\pm 1.5 \times 10^{-45}$.. $\pm 3.4 \times 10^{38}$	0.0f
double	8	$\pm 5.0 \times 10^{-324}$.. $\pm 1.7 \times 10^{308}$	0.0d
int	4	-2147483648.. 2147483647	0
…			

Để xem kích thước của một kiểu dữ liệu: **sizeof**(<kiểu dữ liệu>)



Kiểu dữ liệu

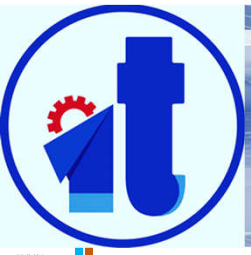
- Phân loại kiểu dữ liệu theo loại giá trị
  - Kiểu số nguyên: sbyte, byte, short, ushort, int, uint, long, ulong, char
  - Kiểu số thực chấm động: float, double
  - Kiểu số thập phân: decimal
  - Kiểu luận lý: bool (true/false)
  - Kiểu dữ liệu khác: Nullable



Kiểu dữ liệu

- Kiểu dữ liệu chuỗi – string
  - Cho phép gán giá trị chuỗi cho biến
  - Kế thừa từ kiểu object
  - Ví dụ:

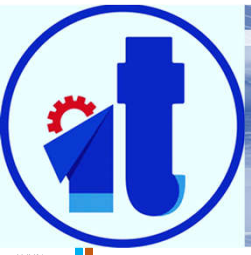
```
string strChuoi = “Xin chao”;
```



Chuyển đổi kiểu dữ liệu

- Chuyển sang chuỗi
  - <biến>.ToString();
  - <biểu thức>.ToString();
- Chuyển từ chuỗi sang kiểu dữ liệu khác
  - Cách 1: <kiểu dữ liệu>.Parse(<chuỗi>);
  - Cách 2: Convert.To<kiểu dữ liệu>(<chuỗi>);

    ToBoolean, ToByte, ToChar, ToDateTime, ToDecimal, ToDouble, ToInt16, ToInt32, ToInt64,...



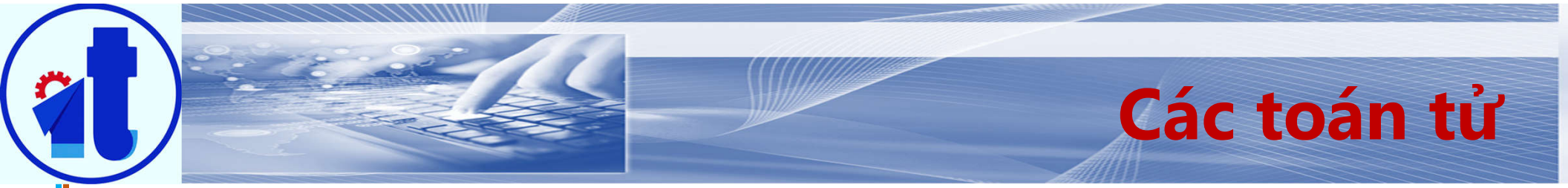
Định nghĩa – khai báo biến

- Cú pháp

  <Kiểu dữ liệu> <danh sách biến>;

  - Kiểu dữ liệu: các kiểu dữ liệu C# định nghĩa sẵn hoặc do người dùng định nghĩa
  - Danh sách biến: danh sách tên biến đặt bởi lập trình viên, danh sách biến này cùng kiểu dữ liệu, phân cách nhau bằng dấu "," (phẩy)

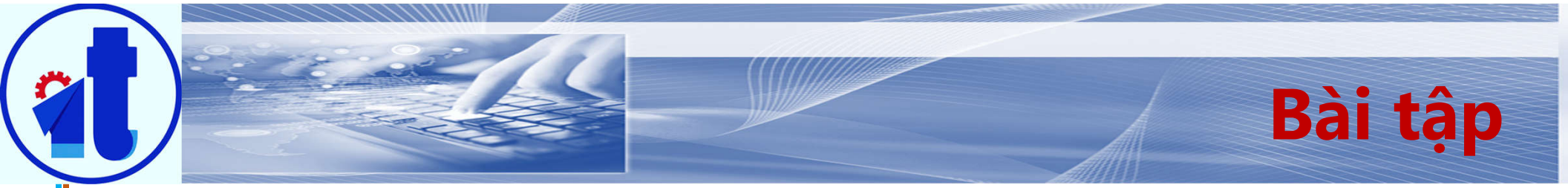
- Khởi tạo biến
  - <tên biến> = <giá trị>;
  - <Kiểu dữ liệu> <tên biến> = <giá trị>;



Các toán tử

- Toán tử số học: +, -, *, /, %, ++, --
- Toán tử so sánh: ==, !=, >, <, >=, <=
- Toán tử luận lý: &&, | |, !
- Toán tử gán: =, +=, -=, *=, /=, %=, ...
- Toán tử hỗn hợp

Toán tử	Mô tả	Ví dụ
&	Trả về địa chỉ của một biến	&x; Trả về địa chỉ của biến x
*	Trỏ đến một biến	*x; Tạo con trỏ với tên là x tới một biến
?:	<Biểu thức điều kiện > ? <Biểu thức thứ 1> : <Biểu thức thứ 2>	x==2?x=3:x=4; nếu x bằng 2 thì gán trị mới x=3 ngược lại thì x=4
is	xác định đối tượng là một kiểu cụ thể hay không	if(iphone is apple) kiểm tra nếu iphone thuộc lớp apple thì..
as	Ép kiểu mà không tạo exception nếu việc ép kiểu thất bại	Object obj = new StringReader("freetuts.net"); StringReader r = obj as StringReader;



Bài tập

1. Viết chương trình nhập vào điểm chuyên cần, trung bình kiểm tra và điểm thi. In ra màn hình tương tự ví dụ sau:

```
Nhap diem chuyen can: 10 (ENTER)
Nhap diem trung binh kiem tra: 8.5 (ENTER)
Nhap diem thi: 9 (ENTER)
----------------------------------------------------------------
DIEM MON HOC LAP TRINH WIN
Ho ten: Lu Cao Tien
MSSV: 0468171234
Lop: CDN QTM 17B
----------------------------------------------------------------
Chuyen can: 10 TBKT: 8.5 Thi: 9
```

Bài tập

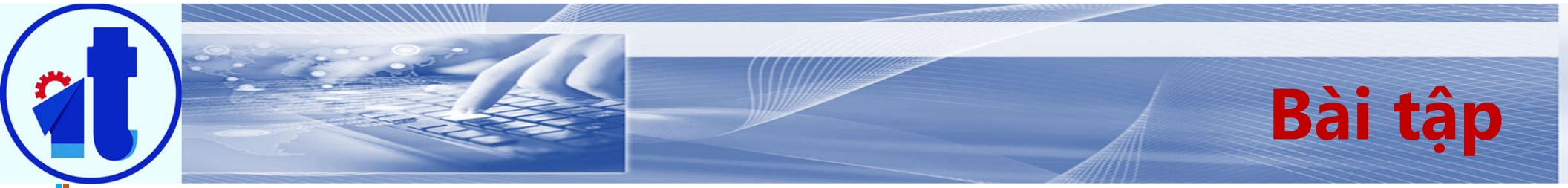
2. Viết chương trình nhập vào tổng số giây. Hãy đổi thành phút – giây từ tổng số giây vừa nhập. In kết quả ra màn hình.

```
Nhap tong so giay: 130 (ENTER)
Quy doi: 2 phut – 10 giay
```

3. Viết chương trình yêu cầu người dùng nhập vào độ C, chuyển giá trị vừa nhập sang độ F. In kết quả ra màn hình.

Cho biết: độ $F = \frac{9}{5} \times$ độ $C + 32$

```
Nhap do C: 43 (ENTER)
Do F: 109.4
```

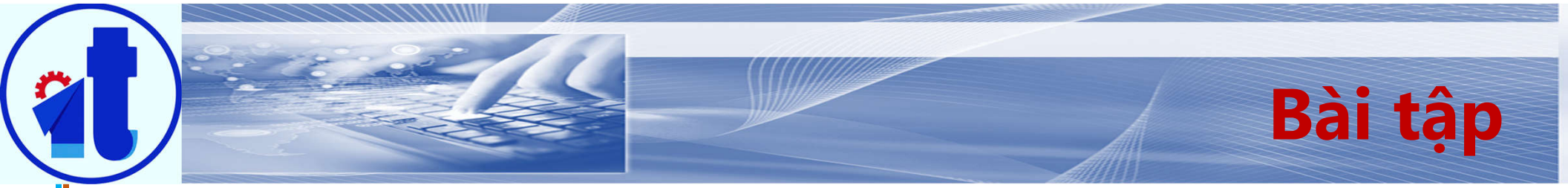


Bài tập

4. Viết chương trình yêu cầu người dùng nhập vào 2 điểm (x_1, y_1), (x_2, y_2) trong mặt phẳng Oxy. Tính khoảng cách giữa 2 điểm và hiển thị kết quả lên màn hình.

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

```
Nhap toa do diem thu nhat: 1.5   -3.4 (ENTER)
Nhap toa do diem thu hai: 4    5 (ENTER)
Khoang cach giua hai diem: 8.764131445842194
```



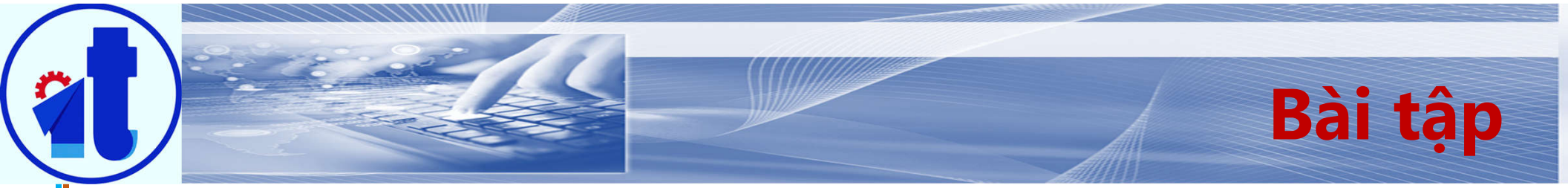
Bài tập

5. Viết chương trình nhập vào hai số nguyên a và b. Hãy hoán vị hai số đó. In kết quả ra màn hình

```
Nhap hai so nguyen: 18 30 (ENTER)
Sau khi hoan vi: 30 18
```

6. Viết chương trình yêu cầu người dùng nhập vào số nguyên có 3 chữ số. Hãy tính tổng 3 chữ số và in kết quả ra màn hình.

```
Nhap so nguyen co 3 chu so: 437 (ENTER)
Tong 3 chu so: 14
```



Bài tập

7. Viết chương trình yêu cầu người dùng nhập một bảng số xe máy gồm 4 chữ số. Hãy cho biết bảng số xe đó được mấy nút?

```
Nhap bang so xe: 5358 (ENTER)
So nut: 1
```

8. Viết chương trình yêu cầu người dùng nhập vào số tiền N. Hãy đổi N ra các loại tiền mệnh giá 10đ, 5đ, 2đ, 1đ (Ưu tiên giá trị tiền từ lớn đến nhỏ). In kết quả lên màn hình.

```
Nhap so tien: 148 (ENTER)
Quy doi: 14 to 10d, 1 to 5d, 1 to 2d, 1 to 1d
```

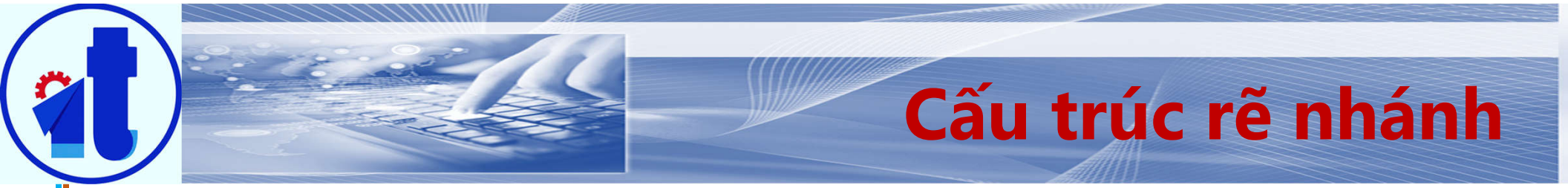
Bài tập

9. Viết chương trình yêu cầu người dùng nhập vào tọa độ 3 đỉnh (x1, y1), (x2, y2) và (x3, y3) của một tam giác. Hãy tính và hiển thị diện tích của tam giác lên màn hình. Công thức tính diện tích tam giác:

$$p = \frac{side1 + side2 + side3}{2}$$

$$s = \sqrt{p(p - side1)(p - side2)(p - side3)}$$

```
Nhap toa do diem thu 1: 1.5 -3.4 (ENTER)
Nhap toa do diem thu 2: 4.6 5 (ENTER)
Nhap toa do diem thu 3: 9.5 -3.4 (ENTER)
Dien tich cua tam giac: 33.6
```

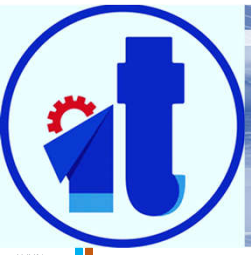


Cấu trúc rẽ nhánh

- Cấu trúc điều kiện if
  - Cú pháp:

```
if (<biểu_thức_điều_kiện>)
{
    /* khối lệnh thực thi khi biểu_thức_điều_kiện là true */
}
```

  - Biểu thức điều kiện: biểu thức luận lý có giá trị true hoặc false
  - Ý nghĩa:
    - ✓ Nếu biểu thức điều kiện có giá trị là true, thì khối lệnh bên trong lệnh if sẽ được thực thi.
    - ✓ Nếu biểu thức điều kiện có giá trị là false, khối lệnh bên trong lệnh if **không** được thực thi mà thực hiện các lệnh sau lệnh if

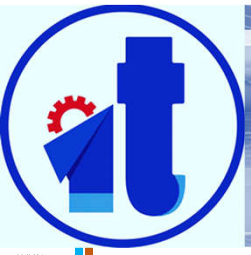


Cấu trúc rẽ nhánh

- Cấu trúc điều kiện if...else
  - Cú pháp:

```
if(<biểu_thức_điều_kiện>)
{
        /* khối lệnh thực thi nếu biểu thức điều kiện là true */
}
else
{
        /* khối lệnh thực thi nếu biểu thức điều kiện là false */
}
```

  - Ý nghĩa: Nếu biểu thức điều kiện có giá trị là true, thì khi đó khối if sẽ được thực thi, ngược lại thì khối else sẽ được thực thi.

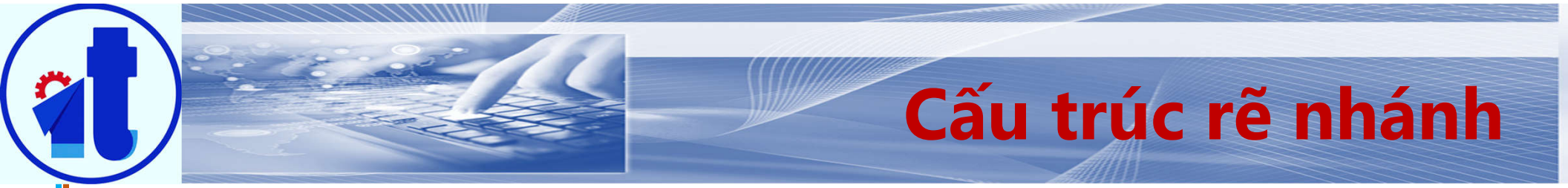


Cấu trúc rẽ nhánh

- if lồng nhau

```
if (điều_kiện)
{
        ...
        [if…else]
        ...
}
else
{
        ...
        [if…else]
        ...
}
```

```
if (điều_kiện_1)
{
        ...
}
else if (điều_kiện_2)
{
        ...
}
...
else if (điều_kiện_n)
{
        ...
}
else
{
}
```

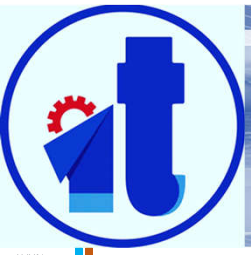


Cấu trúc rẽ nhánh

- Cấu trúc switch...case
  - Cú pháp:

```
switch(biểu_thức) {
case biểu_thức_hằng_1 :
      //khối lệnh cần thực thi;
      break;
case biểu_thức_hằng_2 :
      //khối lệnh cần thực thi;
      break;
...
default : /* tùy ý */
      //khối lệnh cần thực thi;
      break;
}
```

- **Biểu thức**: ở dạng kiểu nguyên hoặc biến... có thể trình bày dạng liệt kê
- **Biểu thức hằng**: cùng kiểu dữ liệu với biểu thức trong switch, giá trị hằng, là một giá trị trong dãy liệt kê
- Khi gặp lệnh break: thoát khỏi cấu trúc lệnh switch...case
- Khi giá trị của biểu thức không bằng giá trị nào trong liệt kê ở case, dòng lệnh thực hiện khối default

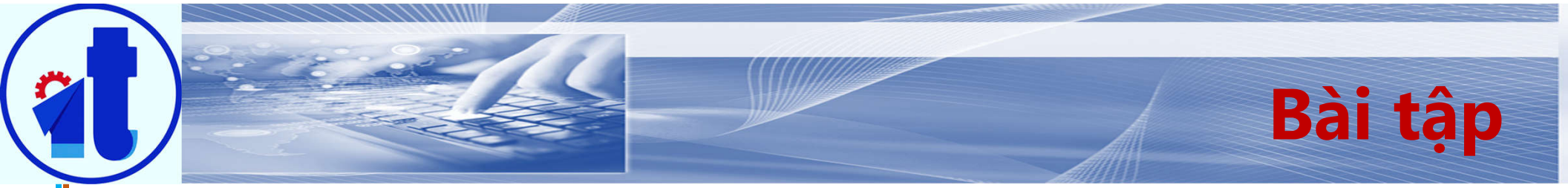


Cấu trúc rẽ nhánh

- Toán tử điều kiện ?
  - Cú pháp

```
(<biểu_thức_điều_kiện>)? Biểu_thức_1 : Biểu_thức_2
```

  - Ý nghĩa: nếu biểu thức điều kiện true, thực thi biểu thức 1; ngược lại thực hiện biểu thức 2.



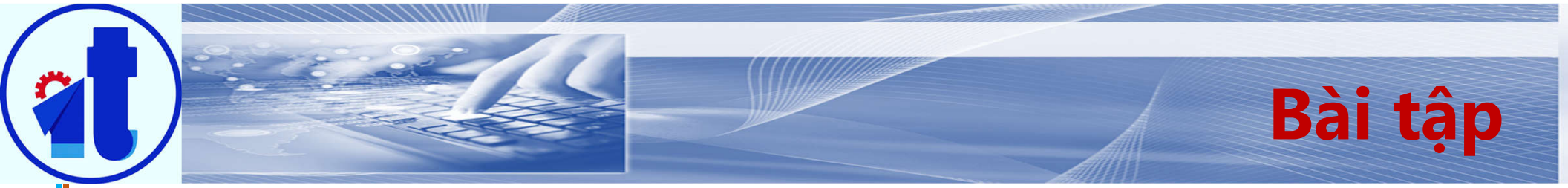
Bài tập

10. Viết chương trình nhập vào 4 số. Xuất ra giá trị lớn nhất của 4 số đó. Không dùng câu lệnh if

```
Nhap so thu 1: 1.5 (ENTER)
Nhap so thu 2:  5 (ENTER)
Nhap so thu 3: 3.4 (ENTER)
Nhap so thu 4: 3 (ENTER)
Gia tri lon nhat: 5
```

11. Viết chương trình nhập vào một số thực, tính giá trị tuyệt đối của nó (không dùng hàm trị tuyệt đối)

```
Nhap vao mot so: -1.5
Gia tri tuyet doi cua no la: 1.5
```



Bài tập

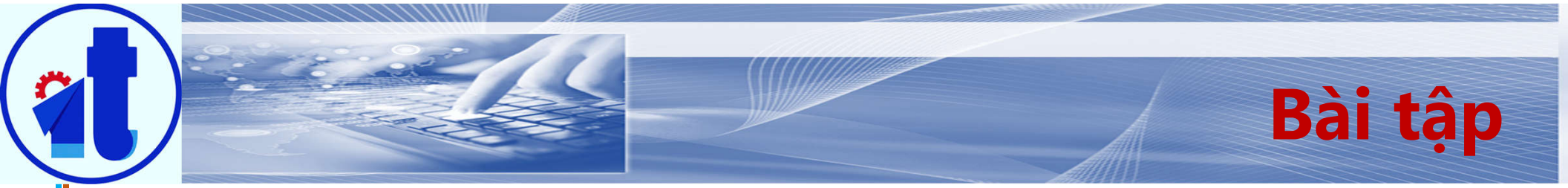
12. Viết chương trình nhập vào năm. Kiểm tra xem năm đó có nhuận không

```
Nhap vao mot nam: 2016
Day la nam nhuan
```

```
Nhap vao mot nam: 2017
Khong phai nam nhuan
```

13. Viết chương trình giải phương trình bậc nhất ax + b = 0. Với a, b là hai số nhập từ bàn phím.

```
Nhap 2 so: 2 -1 (ENTER)
Nghiem x = 0.5
```



Bài tập

14. Viết chương trình nhập vào số tự nhiên từ 0 đến 9, xuất ra cách đọc theo số đó bằng tiếng Việt

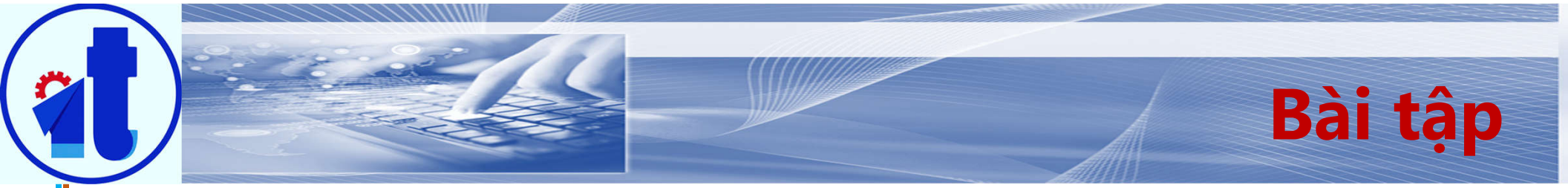
```
Nhap vao mot so tu nhien tu 0 den 9: 6
Sau
```

```
Nhap vao mot so tu nhien tu 0 den 9: -1
Nhap so khong hop le
```

15. Viết chương trình nhập vào khối lượng m (kg) hàng hóa (0 < m ≤ 100). Hãy tính phí vận chuyển, biết rằng phí vận chuyển được tính như sau:

- 0 ≤ m ≤ 10: 5.000đ
- 10 ≤ m ≤ 20: 9.000đ
- 20 ≤ m ≤ 50: 15.000đ
- m ≥ 50: 20.000đ

```
Nhap khoi luong: 0 (ENTER)
Khoi luong phai lon hon 0
----------
Nhap khoi luong: 15 (ENTER)
Phi van chuyen: 9000 dong
```

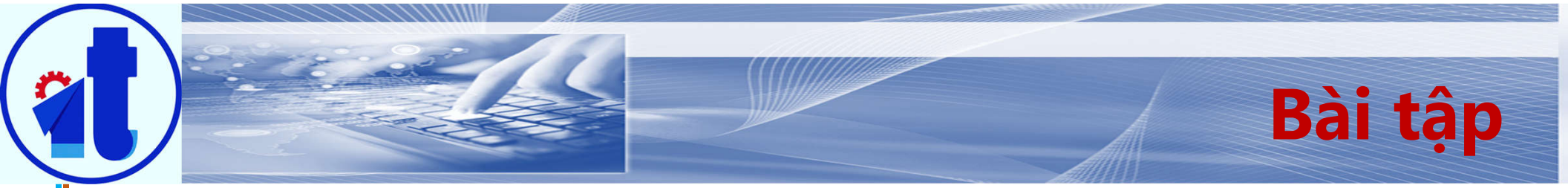


Bài tập

16. Viết chương trình nhập vào thứ (2 → 8) trong tuần. Hãy cho biết tên tiếng Anh của thứ đó?

Thứ	2	3	4	5	6	7	8
Tên	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday

```
Nhap thu: 1 (ENTER)
Thu khong hop le
-----------
Nhap thu: 4 (ENTER)
Thu 4: Wednesday
```

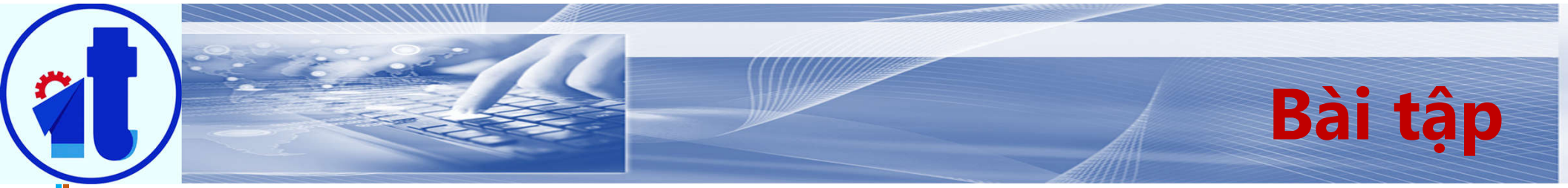


Bài tập

17. Viết chương trình nhập vào một năm dương lịch, cho biết con giáp tương ứng với năm đó

<u>Gợi ý:</u> có 12 con giáp thứ tự: thân, dậu, tuất, hợi, tí, sửu, dần, mão, thìn, tỵ, ngọ, mùi

```
Nhap vao mot Nam: 2017
Dau - Con ga
```



Bài tập

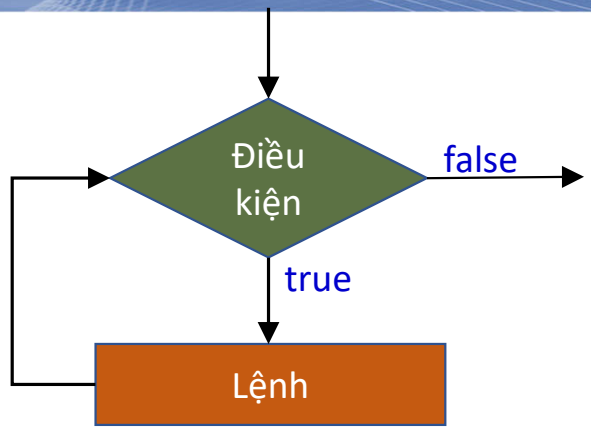
18. Viết chương trình nhập vào chỉ số điện của tháng trước và tháng sau.
    - Tính số điện sử dụng
    - Tính số tiền thanh toán biết đơn giá của mỗi kwh khác nhau theo từng định mức
        - ✓ 100 kwh đầu tiên: 1400 đ/kwh
        - ✓ 50 kwh tiếp theo: 2000 đ/kwh
        - ✓ Từ kwh thứ 151: 3000 đ/kwh

```
Nhap so dien thang truoc: 15 (ENTER)
Nhap so dien thang sau: 128 (ENTER)
----------
So dien: 113
Thanh tien: 166000
```

Cấu trúc vòng lặp

- Vòng lặp while
  - Cú pháp:

```
while(<biểu_thức_điều_kiện>)
{
        /* khối lệnh thực thi */
}
```



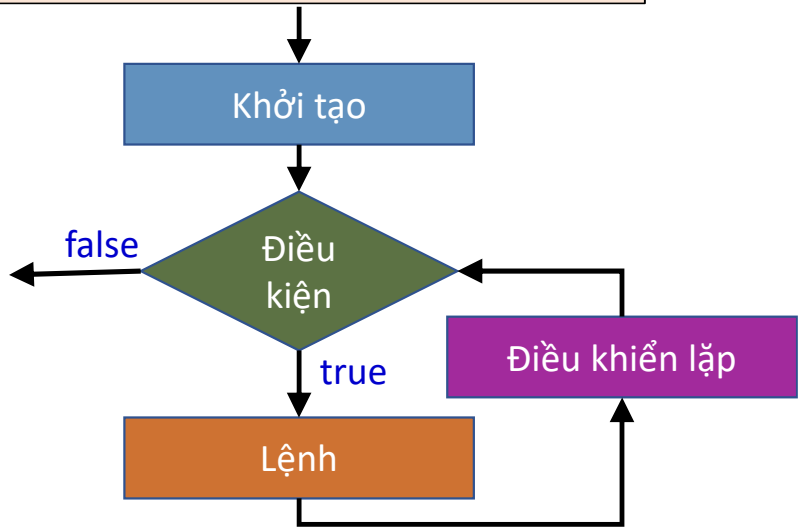
  - Biểu thức điều kiện: biểu thức logic có giá trị true hoặc false
  - Khối lệnh thực thi: lệnh đơn hoặc khối lệnh thực thi khi biểu thức điều kiện có giá trị true
  - Các khối lệnh thực thi sẽ thực hiện lặp đi lặp lại cho điến khi biểu thức điều kiện có giá trị false.

Cấu trúc vòng lặp

- Vòng lặp for
  - Cú pháp:

```
for(<khởi_tạo>;<điều_kiện>;<điều_khiển_lặp>)
{
        /* khối lệnh*/
}
```

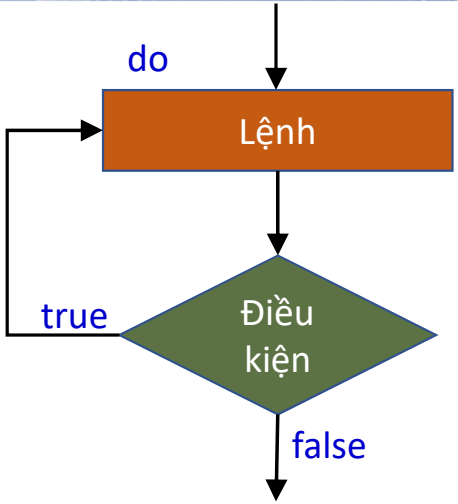
  - Khởi tạo: câu lệnh khởi tạo biến điều khiển lặp, được thực hiện đầu tiên và duy nhất 1 lần trong cấu trúc lặp.
  - Điều kiện: biểu thức điều kiện, nếu có giá trị true thì khối lệnh bên trong sẽ được thực thi.
  - Sau khi các khối lệnh bên trong được thực thi xong, thực hiện lệnh điều khiển lặp rồi kiểm tra điều kiện xem có thực hiện khối lệnh bên trong.
  - Quá trình lặp đi lặp lại cho đến khi biểu thức điều kiện có giá trị false



Cấu trúc vòng lặp

- Vòng lặp do ... while
  - Cú pháp:

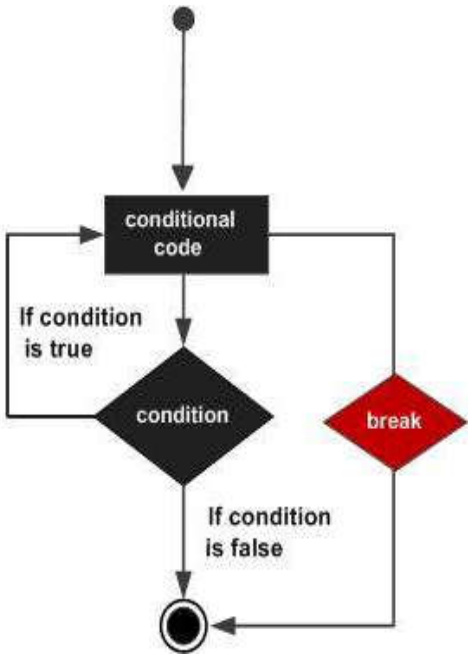
```
do
{
        /* khối lệnh */
}
while(<điều_kiện>);
```

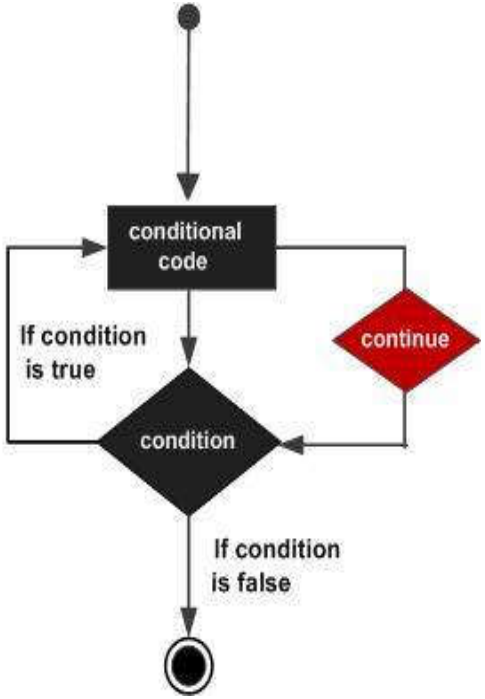


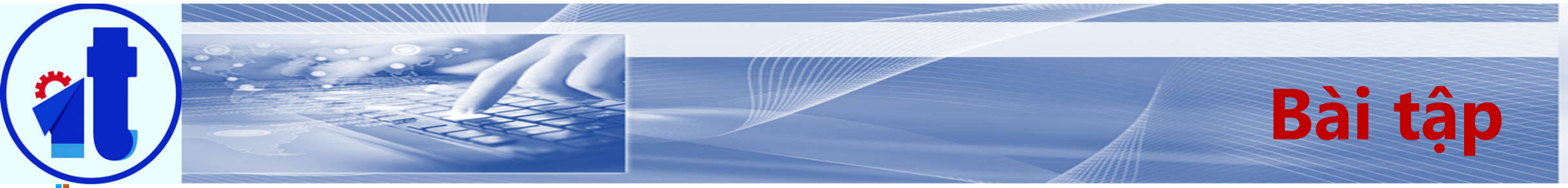
  - Biểu thức điều kiện xuất hiện ở cuối cùng của vòng lặp, vì thế các lệnh trong vòng lặp thực hiện một lần trước khi điều kiện được kiểm tra.
  - Nếu điều kiện là true, dòng điều khiển vòng lặp quay trở lại, và các lệnh trong vòng lặp được thực hiện lần nữa. Tiến trình này lặp đi lặp lại tới khi nào điều kiện đã cho trở thành false.

Cấu trúc vòng lặp

- Lệnh break
  - Khi lệnh break được sử dụng trong vòng lặp / switch, khi gặp lệnh này ngay lập tức kết thúc vòng lặp / switch và điều khiển chương trình bắt đầu lệnh tiếp theo sau vòng lặp / switch.
- Lệnh continue
  - Lệnh continue làm việc hơi giống với lệnh break. Thay vì bắt buộc kết thúc, nó bắt buộc vòng lặp tiếp theo diễn ra: bỏ qua bất kỳ đoạn code nào ở sau continue và bắt đầu lần lặp kế tiếp.







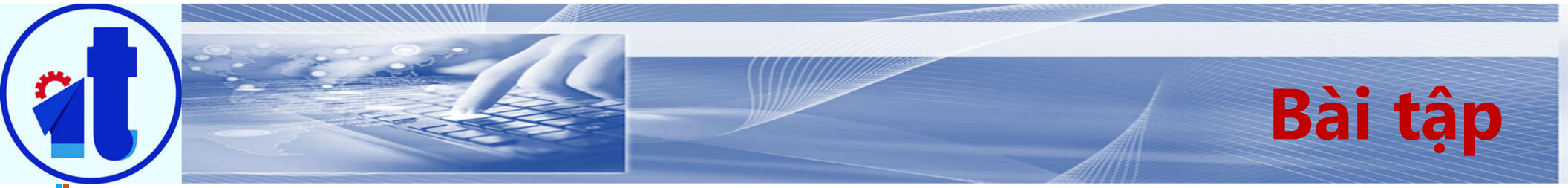
Bài tập

19. Viết chương trình nhập vào một số nguyên dương N (kiểm tra nhập đến khi là số nguyên dương), tính tổng các chữ số của N.

```
Nhap vao mot so nguyen duong: 54903
Tong cac chu so: 21
```

20. Viết chương trình yêu cầu người dùng nhập vào một số nguyên dương N (N > 0). In ra màn hình bảng cửu chương của N.

```
Nhap so nguyen duong N: -1 (ENTER)
Nhap so nguyen duong N: 0 (ENTER)
Nhap so nguyen duong N: 5 (ENTER)
5 x 1 = 5
5 x 2 = 10
5 x 3 =15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x  9 = 45
5 x 10 = 50
```

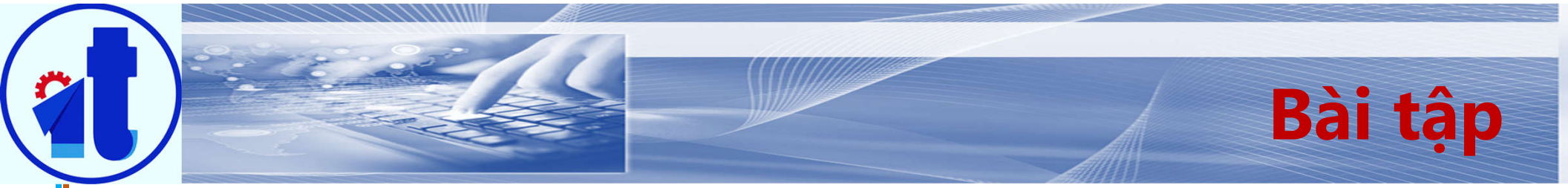


Bài tập

21. Viết chương trình nhập vào tính điểm trung bình kiểm tra của môn học. Yêu cầu nhập số lượng bài kiểm tra, nhập điểm số và hệ số điểm của mỗi bài kiểm tra, tính điểm trung bình và xuất ra kết quả.

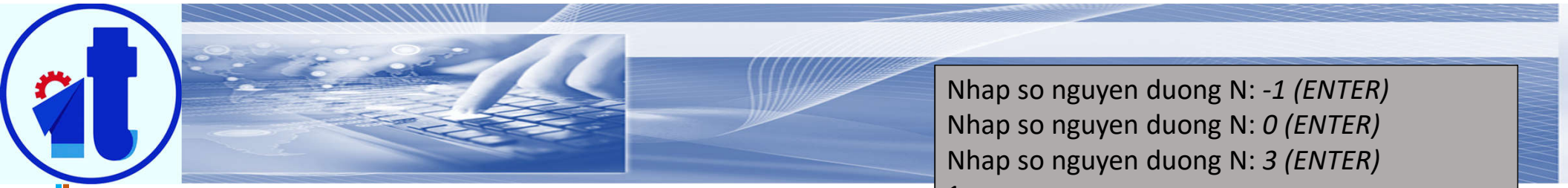
```
Nhap so luong bai kiem tra: 2 (ENTER)
Diem bai kiem tra: 6 (ENTER)
He so: 1 (ENTER)
Diem bai kiem tra: 9 (ENTER)
He so: 2 (ENTER)
--------------------------------------------------
Diem trung binh: 8.0
```

22. Viết chương trình nhập vào các số nguyên cho đến khi người dùng nhập vào số 0, chương trình xuất ra giá trị nguyên âm chẵn lớn nhất trong các số vừa nhập đó.



Bài tập

23. Viết chương trình nhập vào các số nguyên cho đến khi người dùng nhập vào số 0, chương trình xuất ra giá trị nguyên âm chẵn lớn nhất trong các số vừa nhập đó.
24. Viết chương trình tính điểm tổng kết môn học cho danh sách N sinh viên. Giáo viên sẽ nhập điểm trung bình kiểm tra, điểm chuyên cần và điểm thi của mỗi sinh viên. Yêu cầu mỗi khi nhập xong, xuất ra điểm tổng kết (tbkt*0.4+ccan*0.1+thi*0.5) của sinh viên đó ra màn hình, và xếp loại học lực của sinh viên cho môn học (<5: yếu; từ 5 – dưới 6: trung bình; 6 – dưới 7: TB khá; 7- dưới 8: khá; 8 dưới 9: giỏi; từ 9 trở lên: xuất sắc). Sau khi nhập điểm hết danh sách, chương trình sẽ thống kê tỉ lệ phần trăm bao nhiêu sinh viên đạt môn học.



25. Viết chương trình nhập vào số nguyên dương N. Xuất ra các số từ 1→N mà chia hết cho 3 hoặc 7
26. Viết chương trình yêu cầu người dùng nhập vào một số nguyên dương N (N > 0). In ra màn hình các tam giác như sau.

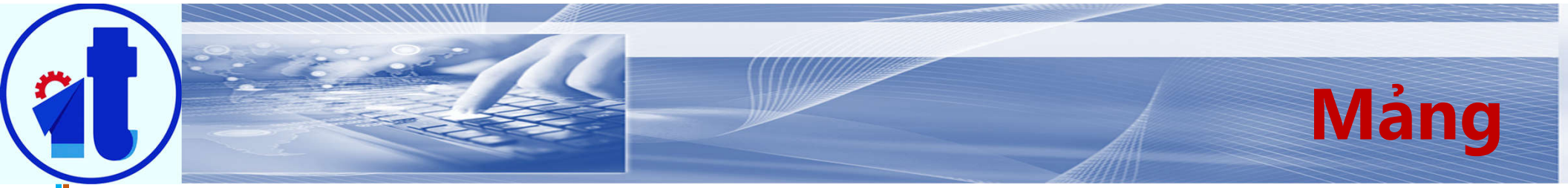
```
Nhap so nguyen duong N: -1 (ENTER)
Nhap so nguyen duong N: 0 (ENTER)
Nhap so nguyen duong N: 3 (ENTER)
1
1 2
1 2 3

*
* *
* * *

1 2 3
1 2
1

* * *
* *
*

3 2 1
2 1
1
```

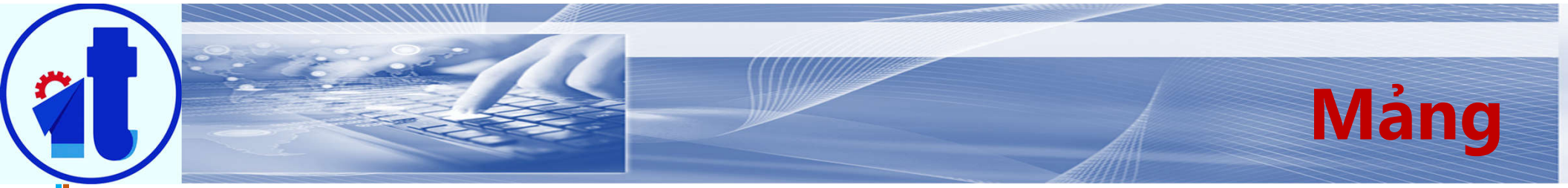


Mảng

- Mảng là một tập hợp có thứ tự của những đối tượng, tất cả các đối tượng này cùng một kiểu.
- Mảng trong ngôn ngữ C# là những đối tượng, do đó, trong mảng sẽ có các phương thức và thuộc tính.
- Ngôn ngữ C# cung cấp cú pháp chuẩn cho việc khai báo đối tượng System.Array.
- Mảng thuộc kiểu tham chiếu

Thuộc tính và phương thức

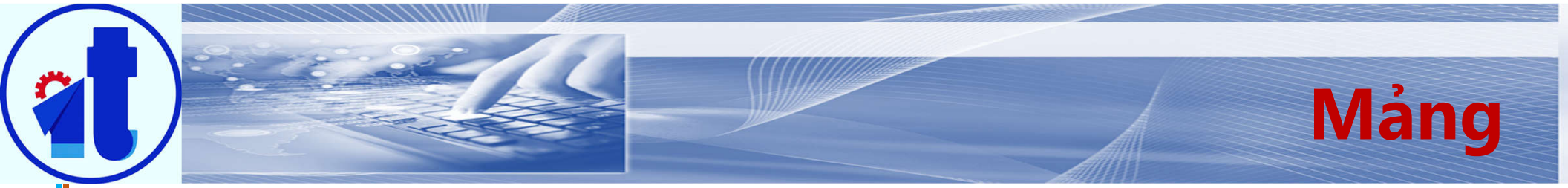
- **Thuộc tính** (property): những **đặc điểm** của đối tượng; có thể được sử dụng tương tự như biến
- **Phương thức** (method): là hành động của đối tượng; sử dụng tương tự như hàm.



LVHN

❑ Thuộc tính và phương thức

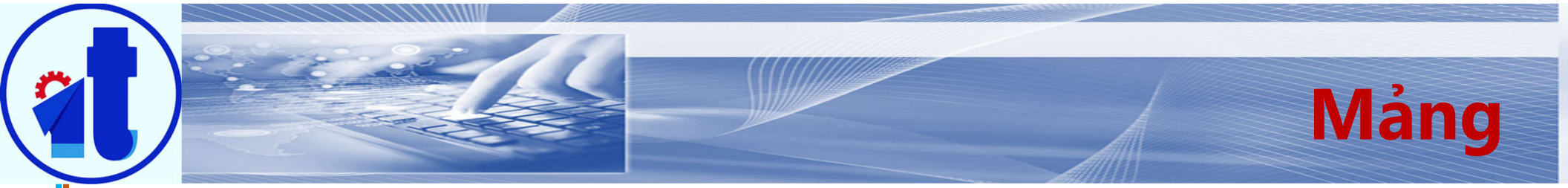
THÀNH VIÊN	MÔ TẢ
BinarySearch()	Phương thức static tìm kiếm trên mảng một chiều đã sắp thứ tự.
Clear()	Phương thức static thiết lập các phần tử của mảng về 0 hay null.
Copy()	Phương thức static đã nạp chồng (overload) thực hiện sao chép một vùng của mảng vào mảng khác.
CreateInstance()	Phương thức static đã nạp chồng (overload) tạo một thể hiện (instance) mới cho mảng
IndexOf()	Phương thức static trả về vị trí đầu tiên của phần tử trong mảng một chiều
LastIndexOf()	Phương thức static trả về vị trí cuối cùng của phần tử trong mảng một chiều
Reverse()	Phương thức static đảo thứ tự của các phần tử trong mảng một chiều



LVHN

❑ Thuộc tính và phương thức

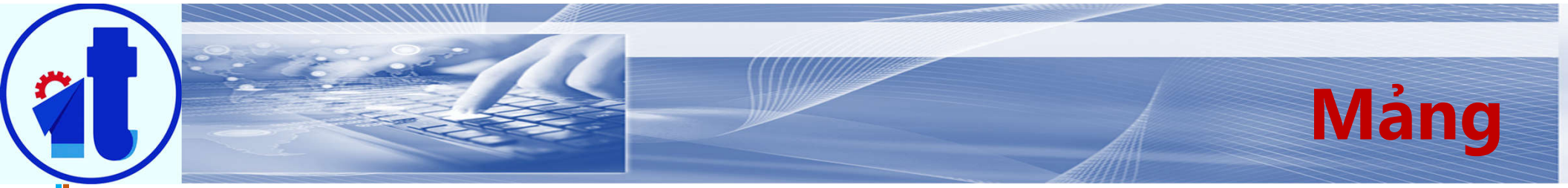
THÀNH VIÊN	MÔ TẢ
Sort()	Phương thức static sắp xếp thứ tự các phần từ trong mảng một chiều
IsFixedSize	Thuộc tính kiểu bool cho biết mảng có kích thước cố định hay thay đổi
IsReadOnly	Thuộc tính kiểu bool cho biết mảng chỉ cho phép đọc hay đọc/ghi
IsSynchronized	Thuộc tính kiểu bool cho biết mảng có hổ trợ an toàn thread (thread-safe)
Length	Thuộc tính cho biết mảng có bao nhiêu phần tử
Rank	Thuộc tính cho biết số chiều của mảng
SyncRoot	Thuộc tính chứa đối tượng dùng để đồng bộ truy cập trong mảng
GetEnumerator()	Phương thức thành viên trả về đối tượng IEnumerator dùng để duyệt các phần tử của mảng



LVHN

❑ Thuộc tính và phương thức

THÀNH VIÊN	MÔ TẢ
GetLength()	Phương thức thành viên trả về kích thước của một chiều trong mảng
GetLowerBound()	Phương thức thành viên trả về cận dưới của một chiều trong mảng
GetUpperBound()	Phương thức thành viên trả về cận trên của một chiều trong mảng
Initialize()	Khởi tạo các phần tử trong mảng bằng cách gọi default contructor cho từng phần tử
SetValue()	Phương thức thành viên thiết lập giá trị cho một phần tử xác định trong mảng



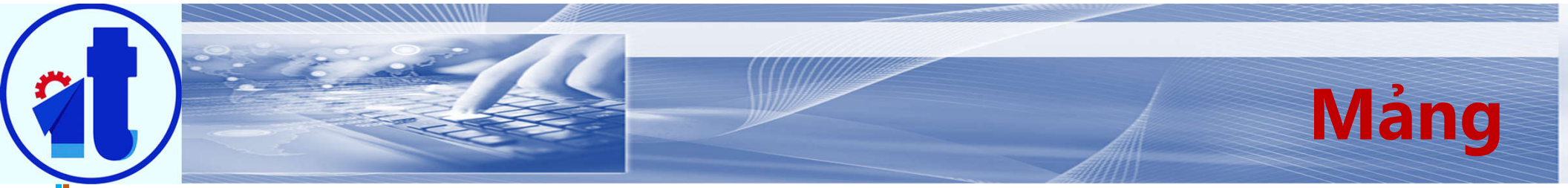
Mảng

- Khai báo mảng một chiều
  - Cú pháp: `Kiểu_dữ_liệu[] tên_mảng;`
  - Cặp đấu ngoặc vuông "`[]`" báo cho trình biên dịch biết rằng chúng ta đang khai báo một mảng.
  - Kiểu dữ liệu sẽ là kiểu của các phần tử trong mảng.
  - Ví dụ:

```
int[] arrayInt;
arrayInt = new int[10];
```

- Khởi tạo giá trị các phần tử: đặt những giá trị bên trong dấu ngoặc ({}).
  - Ví dụ:

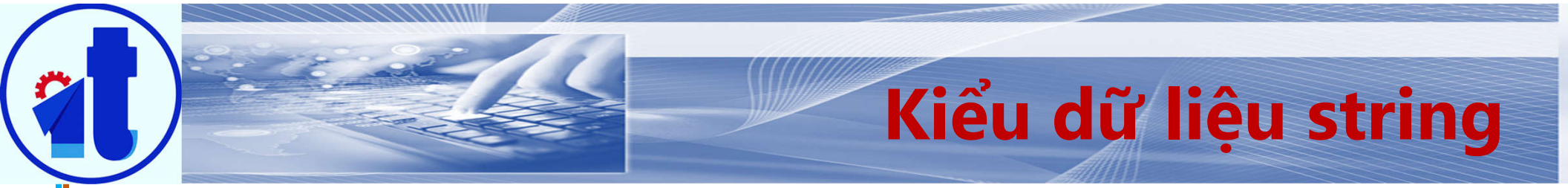
```
int[] arrayInt = new int[4] { 2, 4, 6, 8 };
int[] arrayInt = { 2, 4, 6, 8 };
```



Mảng

- Truy xuất các phần tử:
  - Để truy cập vào phần tử trong mảng ta có thể sử dụng toán tử chỉ mục ([]).
  - Một mảng có thể được đánh chỉ số từ 0 đến length – 1.
  - Ví dụ:

```
int[] arrayInt = { 2, 4, 6, 8 };
System.Console.WriteLine("phan tu thu 2: {0)", arrayInt[1]);
```



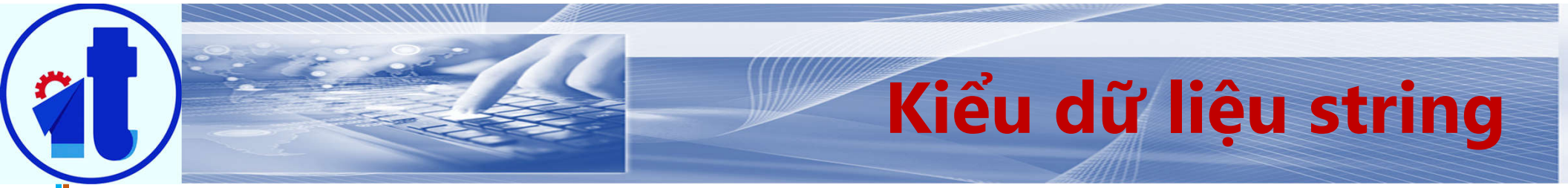
Kiểu dữ liệu string

- Sử dụng các chuỗi (string) như là mảng các ký tự.
- Kiểu dữ liệu string thuộc lớp đối tượng System.String
- Khai báo: string tên_biến;
- Tạo đối tượng string:
  - Gán giá trị cho chuỗi: string strHoTen = "Nguyen Van An";
  - Sử dụng constructor (hàm khởi tạo):

```
char[] arrChao = {'H', 'e', 'l', 'l', 'o'};
string strChao = new string (arrChao);
```

  - Sử dụng toán tử nối chuỗi: string strLoiChao = strChao + " " + strHoTen;
  - Sử dụng các phương thức, biến đổi kiểu dữ liệu sang chuỗi

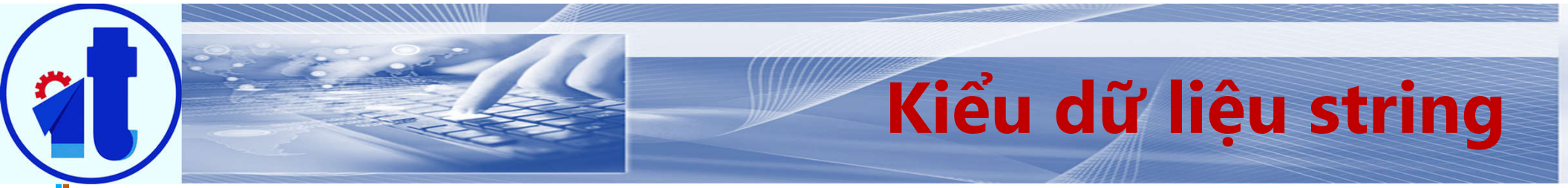
```
int iSo = 75;
string strSo = iSo.ToString();
```



Kiểu dữ liệu string

❑ Thuộc tính và phương thức trong lớp đối tượng String

THÀNH VIÊN	MÔ TẢ
Chars	chứa mảng ký tự từ chuỗi. Ví dụ: String a = "Hello" char a2 = a[2];
Length	Thuộc tính cho biết độ dài (số ký tự)
public static int Compare(string strA, string strB)	Phương thức so sánh chuỗi strA và strB, theo thứ tự từ điển; kết quả: 0 nếu 2 chuỗi bằng nhau, 1 nếu chuỗi A > B, -1 nếu B > A
public static int Compare(string strA, string strB, bool ignoreCase)	So sánh hai đối tượng String cụ thể và trả về một integer, kiểm tra hoa thường
public static string Concat(string str0, string str1, …)	Nối chuỗi đối tượng

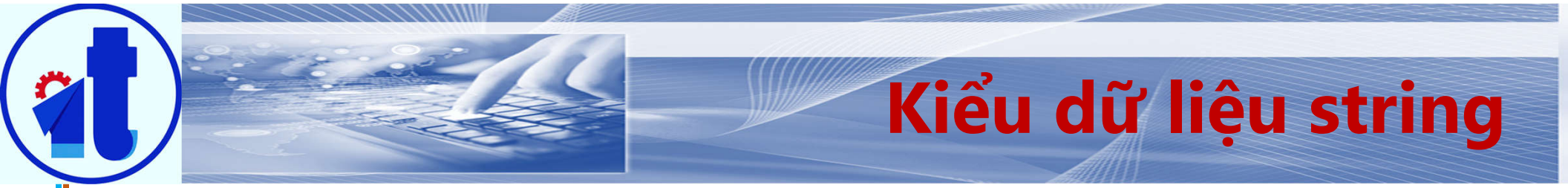


Kiểu dữ liệu string

LVHN

❑ Thuộc tính và phương thức trong lớp đối tượng String

THÀNH VIÊN	MÔ TẢ
public bool Contains(string value)	Kiểm tra chuỗi có chứa chuỗi value không
public static string Copy(string str)	Tạo một đối tượng String sao chép chuỗi đã cho
public void CopyTo(int sourceIndex, char[] destination, int destinationIndex, int count)	Sao chép một số ký tự cụ thể từ một vị trí đã cho của đối tượng String tới một vị trí đã xác định trong một mảng các ký tự Unicode
public bool EndsWith(string value)	Kiểm tra đối tượng String có kết thúc bằng chuỗi value không
public static bool Equals(string a, string b)	Kiểm tra a và b có giá trị bằng nhau không
public static string Format(string format, Object arg0)	Định dạng chuỗi có đối tượng truyền vào
public int IndexOf(<kiểu dữ liệu> value) <Kiểu dữ liệu>: char, string, char[]	Vị trí đầu tiên xuất hiện của value trong chuỗi

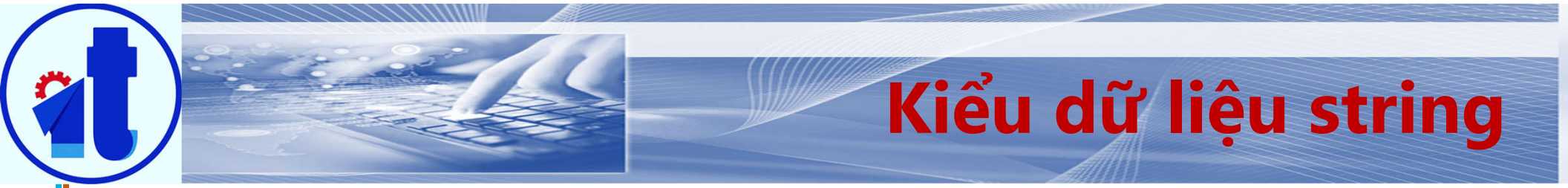


Kiểu dữ liệu string

LVHN

❑ Thuộc tính và phương thức trong lớp đối tượng String

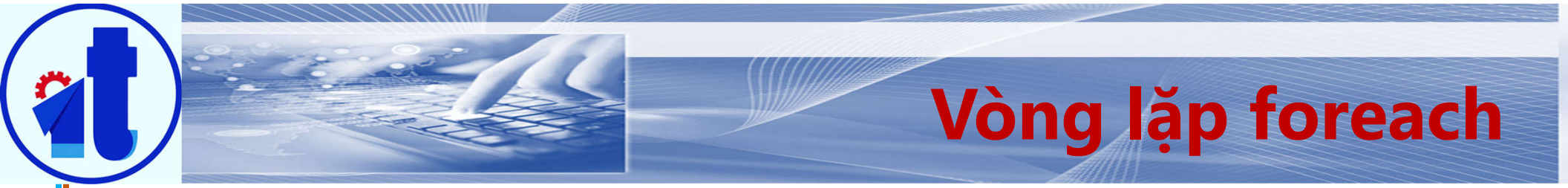
THÀNH VIÊN	MÔ TẢ
public int IndexOf(<kiểu dữ liệu> value, int startIndex) <Kiểu dữ liệu>: char, string, char[]	Vị trí đầu tiên xuất hiện của value trong chuỗi (bắt đầu tìm tại vị trí startIndex)
public string Insert(int startIndex, string value)	Chèn chuỗi value vào chuỗi đã cho tại vị trí startIndex
public static bool IsNullOrEmpty(string value)	Kiểm tra chuỗi có null hoặc rỗng không
public static string Join(string separator, params string[] value)	Nối các choỗi trong mảng value cách nhau bởi separator
public int LastIndexOf(<Kiểu dữ liệu> value) <Kiểu dữ liệu>: char, string	Vị trí cuối cùng value xuất hiện trong chuỗi đã cho
public string Remove(int startIndex)	Bỏ các giá trị từ vị trí startIndex đến hết chuỗi
public string Remove(int startIndex, int count)	Bỏ count ký tự liên tục bắt đầu từ vị trí startIndex



Kiểu dữ liệu string

❑ Thuộc tính và phương thức trong lớp đối tượng String

THÀNH VIÊN	MÔ TẢ
public string Replace(char oldChar, char newChar) public string Replace(string oldChar, string newChar)	Thay thế oldChar trong chuỗi đã cho bằng newChar
public string[] Split(params char[] separator)	Cắt chuỗi đã cho phân cách nhau bởi separator
public bool StartsWith(string value)	Kiểm tra chuỗi đã cho có bắt đầu bằng value không
public char[] ToCharArray()	Chuyển chuỗi thành mảng ký tự
public string ToLower()	Chuyển sang chuỗi thường
public string ToUpper()	Chuyển sang chuỗi in hoa
public string Trim()	Bỏ những khoảng trống thừa
…	

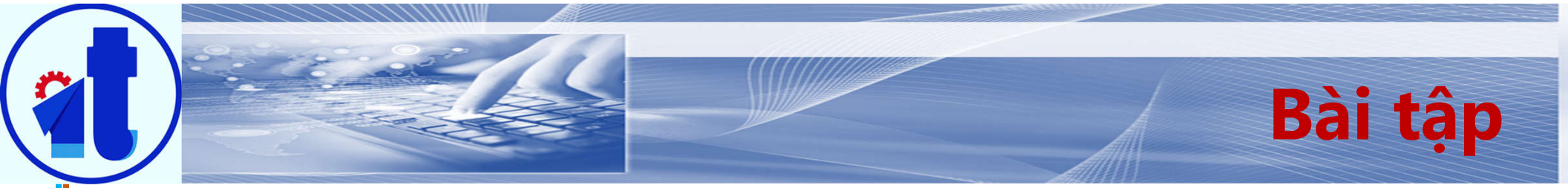


Vòng lặp foreach

- Cú pháp

```
foreach(kiểu_dữ_liệu tên_biến_tạm in danh_sách)
{
      /* khối lệnh thực thi */
}
```

- `danh_sách:` danh sách các phần tử
- `kiểu_dữ_liệu:` kiểu dữ liệu các phần tử trong danh sách
- `tên_biến_tạm:` đại diện phần tử được duyệt trong danh sách

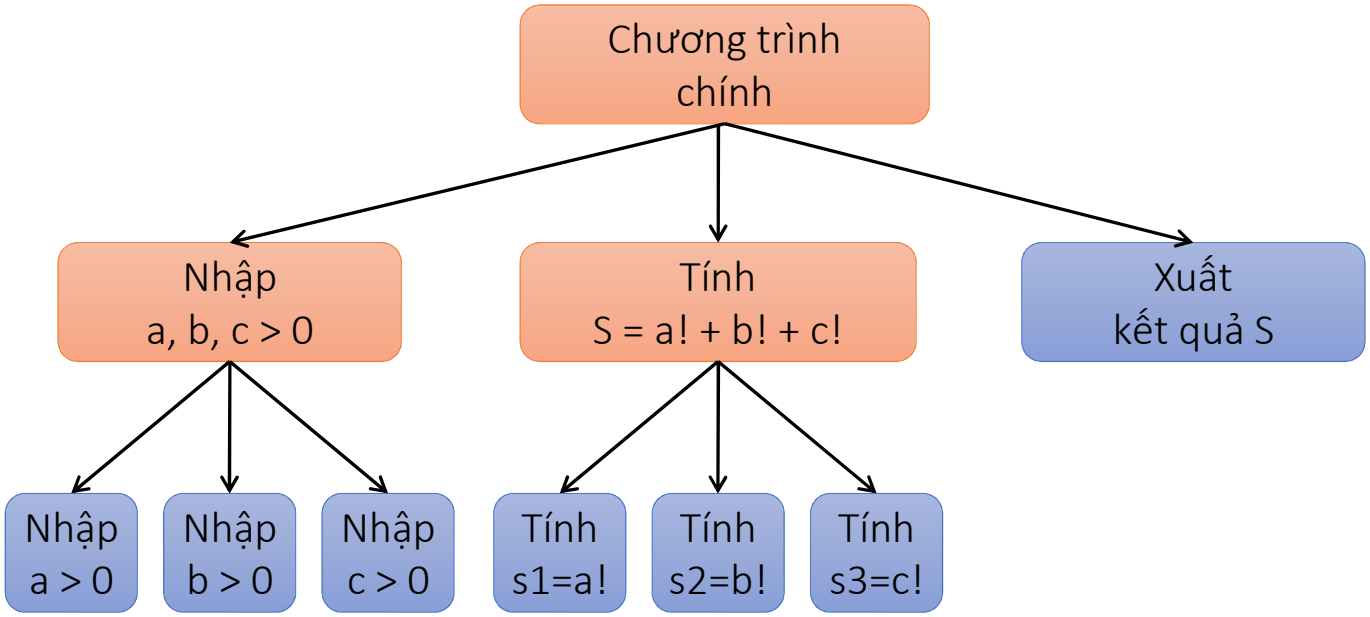


Bài tập

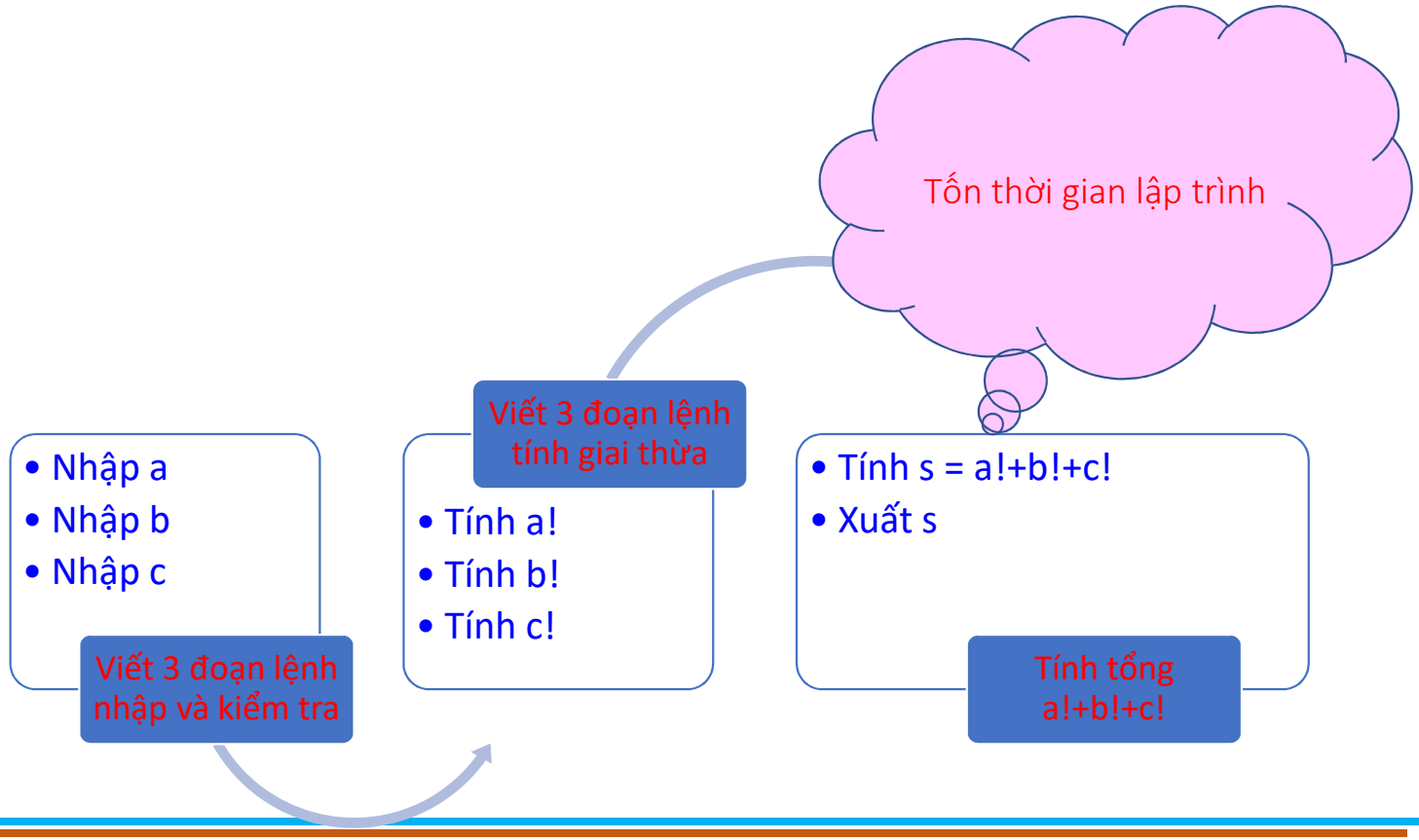
27. Viết chương trình nhập vào một chuỗi, cho biết số ký tự không là khoảng trắng của chuỗi
28. Viết chương trình nhập vào họ và tên của một người (Viết theo tiếng Việt không dấu). Cho biết họ của người đó

Hàm – Tham số

- Viết chương trình tính S = a! + b! + c! với a, b, c là 3 số nguyên dương nhập từ bàn phím.



Hàm – Tham số


Tốn thời gian lập trình
Viết 3 đoạn lệnh tính giai thừa
• Nhập a
• Nhập b
• Nhập c
Viết 3 đoạn lệnh nhập và kiểm tra
• Tính a!
• Tính b!
• Tính c!
• Tính s = a!+b!+c!
• Xuất s
Tính tổng a!+b!+c!

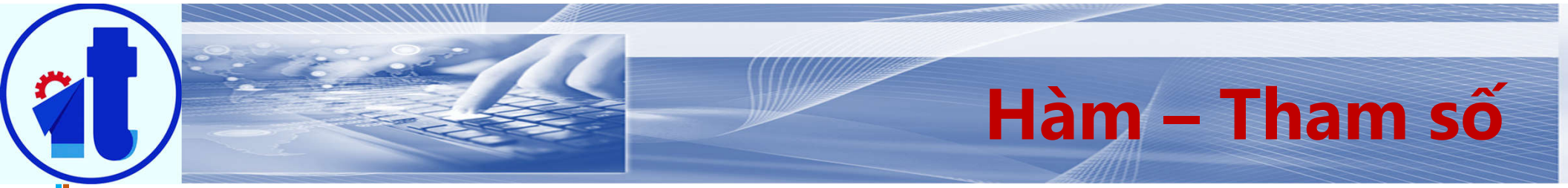
Hàm – Tham số

- Viết 1 lần và sử dụng nhiều lần
  - Đoạn lệnh nhập tổng quát, với n = a, b, c

```
do {
        Console.WriteLine("Nhap mot so nguyen duong: ");
        int n = int.Parse(Console.ReadLine());
} while (n <= 0);
```

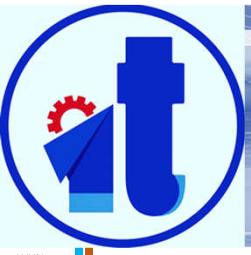
  - Đoạn lệnh tính giai thừa tổng quát, n = a, b, c

```
// Tính s = n! = 1 * 2 * … * n
s = 1;
for (i = 2; i <= n ; i++)
        s = s * i;
```



Hàm – Tham số

- Khái niệm:
  - Một đoạn chương trình có tên, đầu vào và đầu ra.
  - Có chức năng giải quyết một số vấn đề chuyên biệt cho chương trình chính.
  - Được gọi nhiều lần với các tham số khác nhau.
  - Được sử dụng khi có nhu cầu:
    - ✓ Tái sử dụng.
    - ✓ Sửa lỗi và cải tiến.



Hàm – Tham số

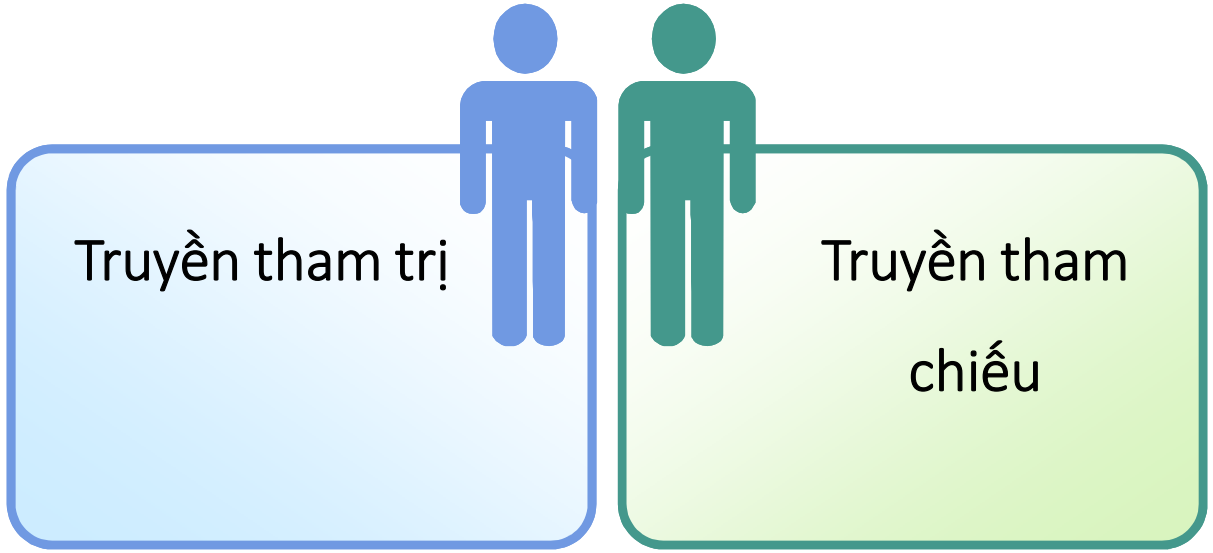
- Cú pháp

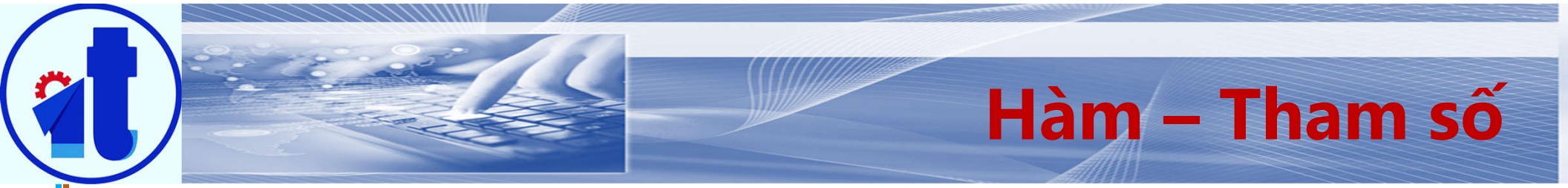
```
<kiểu trả về> <tên hàm>([danh sách tham số])
{
      <các câu lệnh>
      [return <giá trị>;]
}
```

- Trong đó
  - <kiểu trả về>: kiểu bất kỳ của C (int, float,...). Nếu không trả về thì là void.
  - <tên hàm>: theo quy tắc đặt tên định danh.
  - <danh sách tham số>: tham số hình thức đầu vào giống khai báo biến, cách nhau bằng dấu ,
  - <giá trị>: trả về cho hàm qua lệnh return.

Hàm – Tham số

- Các cách truyền đối số





Hàm – Tham số

Tham số - truyền tham trị (call by value)

- Truyền đối số cho hàm ở dạng giá trị.
- Có thể truyền hằng, biến, biểu thức nhưng hàm chỉ nhận giá trị.
- Được sử dụng khi không có nhu cầu thay đổi giá trị của đối số sau khi thực hiện hàm.

```
public static void TruyenGiaTri(int x)
{
        …
        x++;
}
```

Hàm – Tham số

Tham số - truyền tham chiếu (call by reference)

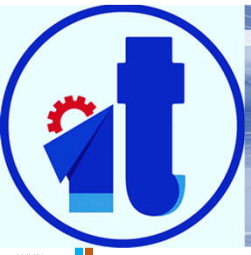
- Tham số có giá trị thay đổi trước và sau khi thực hiện phương thức, có thể đi sau các từ khóa: ref, out, params
- Tham số ref: Tương tự như truyền tham chiếu trong C/C++
  - ✓ Từ khoá ref phải được dùng lúc gọi hàm
  - ✓ Các tham số truyền dạng ref phải được khởi tạo giá trị trước.

```
public static int Main()
{
    int num1 = 5, num2 = 2;
    Swap(ref num1,ref num2);

    return 0;
}
public static void Swap(ref int a, ref int b)
{
    int temp;
    temp = a;
    a = b;
    b = temp;
}
```

Sử dụng ref cho tham số khi gọi hàm

Khai báo ref trước kiểu dữ liệu



Hàm – Tham số

- Tham số - truyền tham chiếu (call by reference)
  - Tham số out: Tương tự như ref
  - Khác ref là không cần khởi tạo giá trị trước khi truyền

```
public static int Main()
{
    int a = 10, b;
    Subtraction(a, out b);

    return 0;
}
public static void Subtraction(int x, out int y)
{
    y = x - 2;
}
```

Dùng trước tham số khi gọi hàm

Khai báo cho tham số

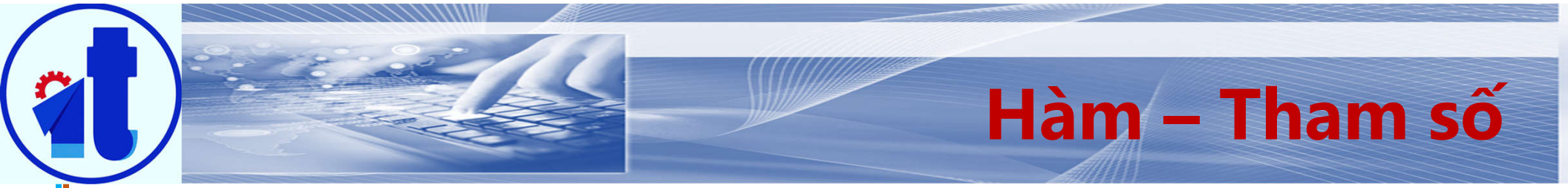
Hàm – Tham số

- Tham số - truyền tham chiếu (call by reference)
  - Tham số params: Tương tự như ref
  - Khác ref là cho phép truyền tham chiếu với đối tượng có số lượng phần tử nhiều.

```
public static void Sum( out int result, params int[] myArray)
{
     result = 0;
    for (int i = 0; i < myArray.Length; i++)
        result += myArray[i];

}
```

Luôn khai báo ở cuối danh sách các tham số



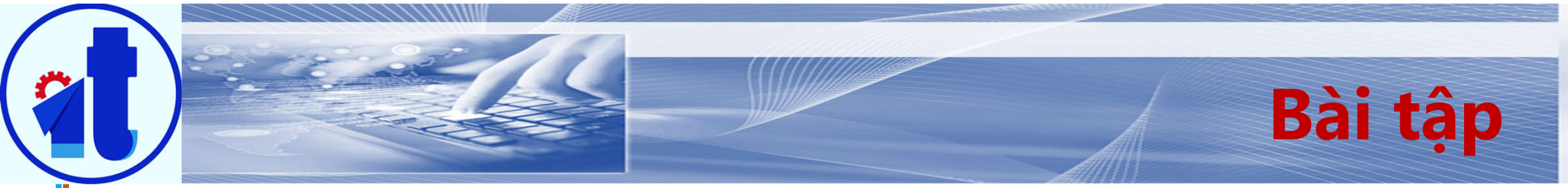
Hàm – Tham số

LVHN

- Ví dụ minh họa

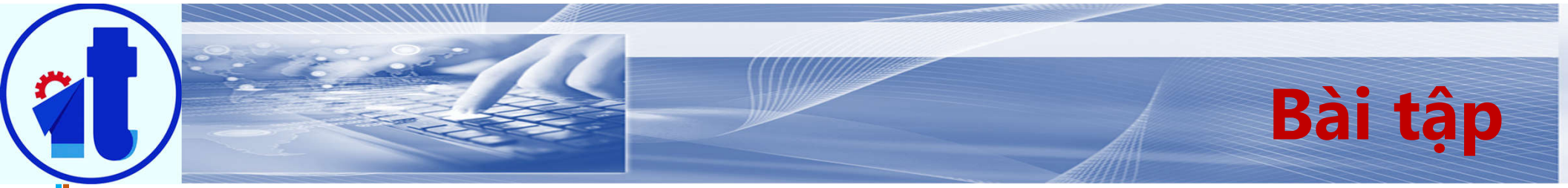
```
using System;
using System.Collections.Generic;
using System.Text;
namespace VD_HamConsole
{
    class Program
    {
        static void Nhap(ref int a, ref int b)
        {
            string s;
            Console.Write("Nhap a: ");
            s = Console.ReadLine();
            a = int.Parse(s);
            Console.Write("Nhap b: ");
            s = Console.ReadLine();
            b = int.Parse(s);
        }
    }
}
```

```
static int TinhGiaTriTong(int a, int b)
      {
           int tong;
           tong = a + b;
           return tong;
       }
       static void XuatKetQua(int tong)
       {
           Console.WriteLine(tong.ToString());
       }
       static void Main(string[] args)
       {
           int a, b;
           int tong;
           a = 0;
           b = 0;
           Nhap(ref a, ref b);
           tong = TinhGiaTriTong(a, b);
           XuatKetQua(tong);
       }
   }
}
```



Bài tập

29. Viết chương trình gồm các hàm
    - Viết hàm đổi một ký tự hoa sang ký tự thường.
    - Viết thủ tục giải phương trình bậc nhất.
    - Viết thủ tục giải phương trình bậc hai.
    - Viết hàm trả về giá trị nhỏ nhất của 4 số nguyên.
    - Viết thủ tục hoán vị hai số nguyên.



Bài tập

30. Hàm nhận vào một số nguyên dương n và thực hiện:
    - S = 1 + 2 + … + n
    - S = $1^2 + 2^2 + … + n^{ss}$
    - S = 1 + 1/2 + … + 1/n
    - S = 1 * 2 * … * n
    - S = 1! + 2! + … + n!

Q&A