

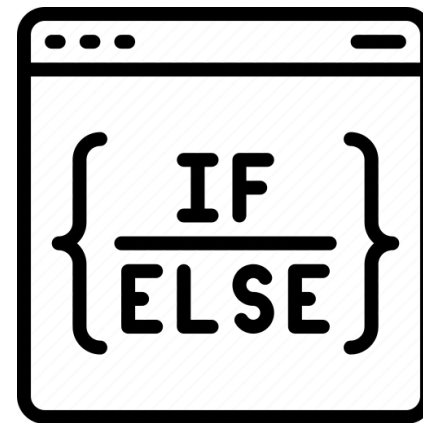
CÂU LỆNH

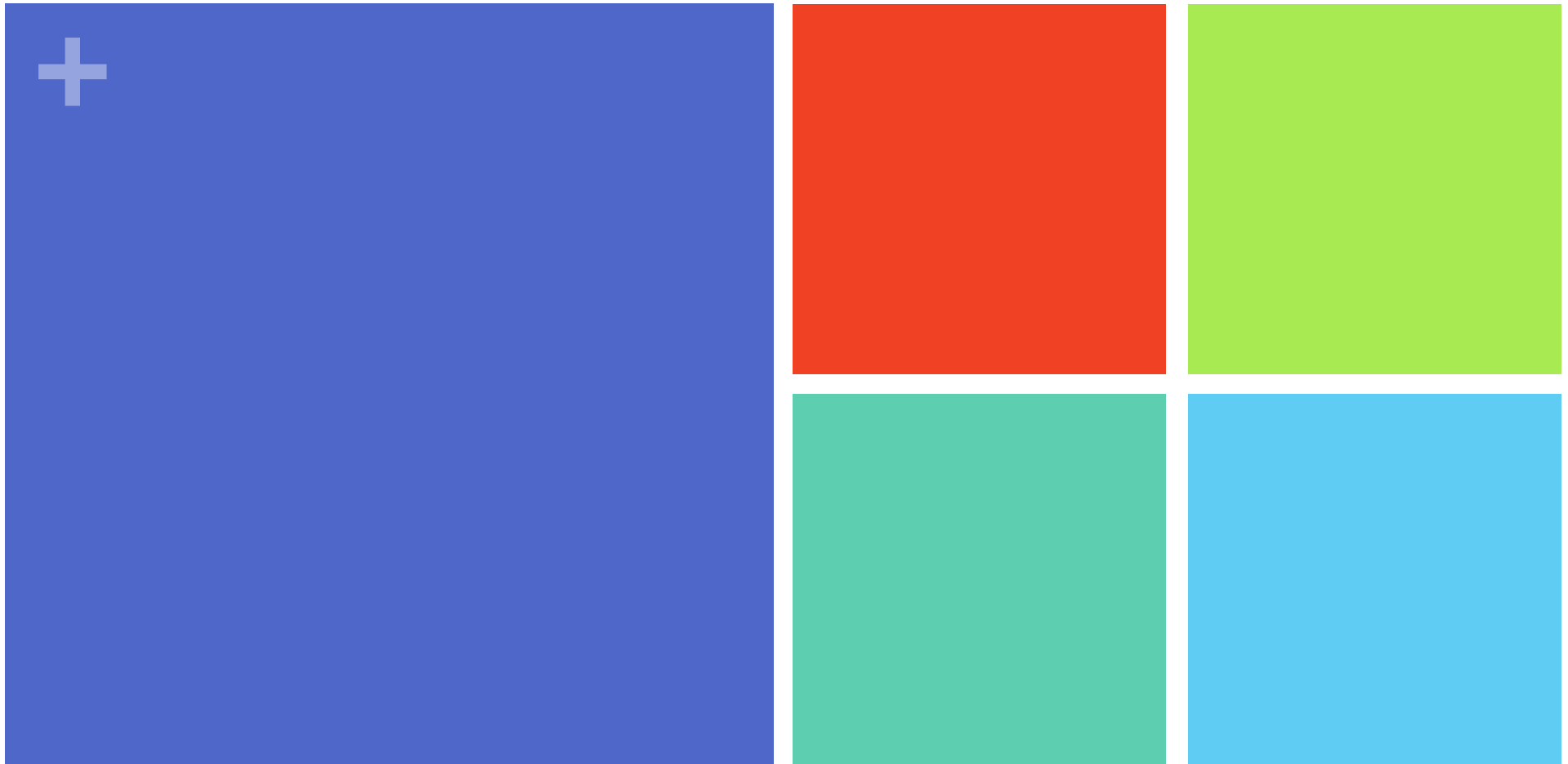
TS. ĐẶNG THÀNH TRUNG



NỘI DUNG

- Câu lệnh cơ bản
- Cấu trúc rẽ nhánh
- Cấu trúc lặp





CÂU LỆNH CƠ BẢN

TS. ĐẶNG THÀNH TRUNG



CÁC CÂU LỆNH ĐƠN GIẢN

- Lệnh rỗng

`;` // dấu “;” là kết thúc lệnh

- Lệnh gán

`x = 3 * y + 1;`

- Gọi phương thức

`string s = "a,b,c";`

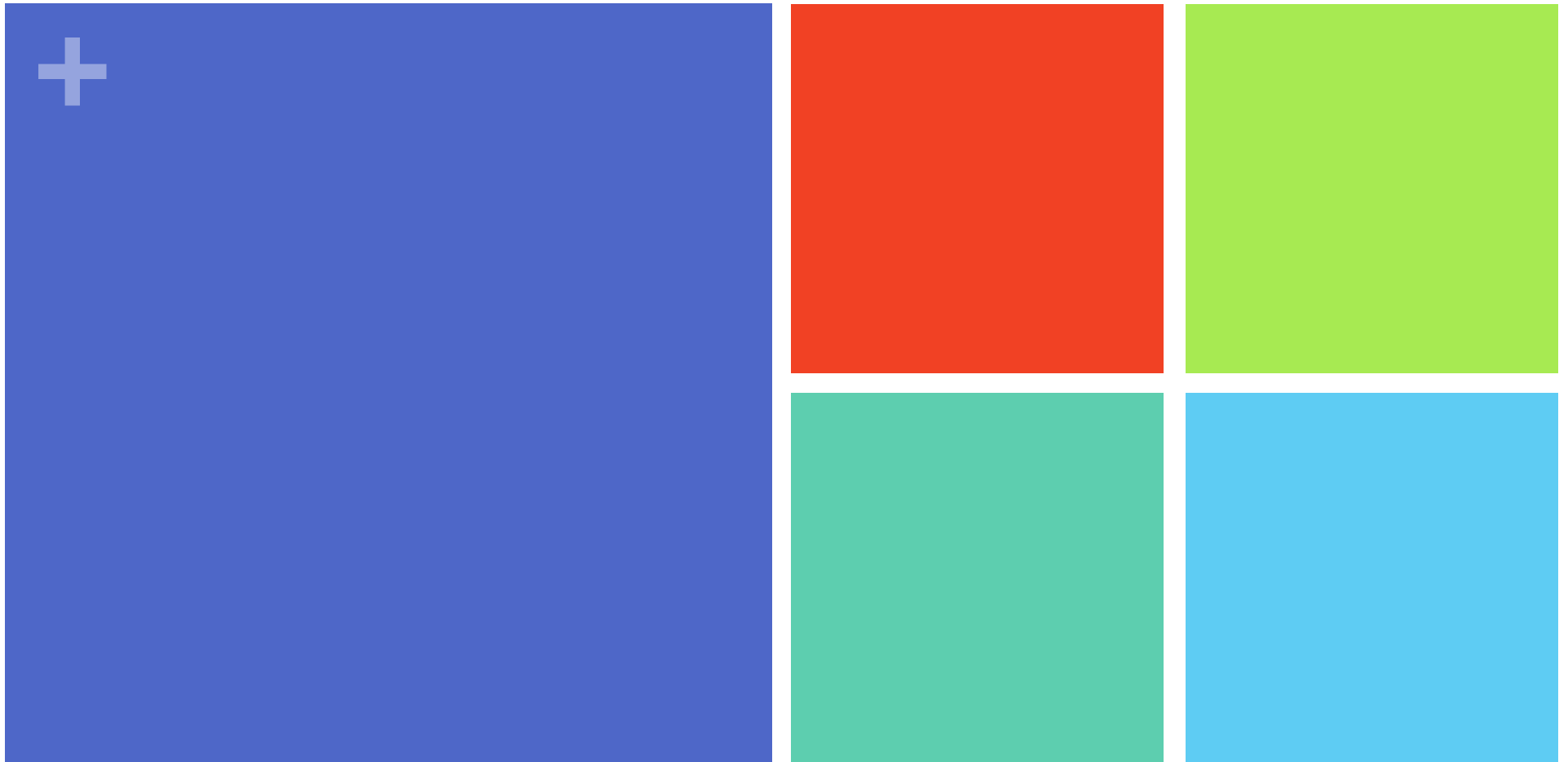
`string[] parts = s.Split(',');// viện dẫn tới phương thức của đối tượng
// (non-static)`

`s = String.Join(" + ", parts);// viện dẫn tới phương thức của class
(static)`



LỆNH VÀO/RA TRÊN CONSOLE

- **Hiển thị lên Console**
 - **Cú pháp :**
 - `System.Console.Write(<tham số>)`
 - `System.Console.WriteLine(<tham số>)`
 - **Ví dụ :**
 - `System.Console.Write("Hello everybody.");`
 - `System.Console.Write("x = {0} ",x);`
- **Đọc giá trị từ bàn phím**
 - **Cú pháp :**
 - `int System.Console.Read();`
 - `string System.Console.ReadLine();`
 - **Ví dụ :**
 - `int ch = System.Console.Read();`
 - `string line = System.Console.ReadLine();`



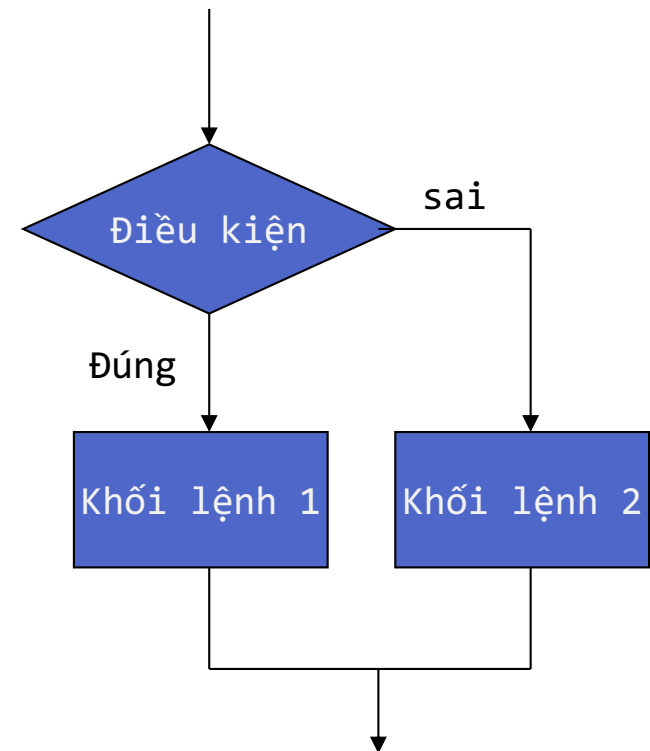
CẤU TRÚC RỄ NHÁNH

TS. ĐẶNG THÀNH TRUNG



LỆNH IF ... ELSE ...

- Lệnh **if ... else ...** phân nhánh dựa trên một điều kiện duy nhất.
- **else** là phần tùy chọn, và chỉ được thực hiện khi điều kiện khi điều kiện bị sai
- Cú pháp :
if (biểu thức điều kiện)
<khối lệnh 1>
[else <khối lệnh 2>]
- Nếu các khối lệnh trong điều kiện if có nhiều hơn một lệnh thì chúng phải được đặt trong cặp dấu {}





LỆNH SWITCH

- Lệnh **switch** phân nhánh tùy theo từng giá trị của của biểu thức điều kiện

- Cú pháp :

```
switch (biểu thức điều kiện) {
```

```
case <giá trị 1> :
```

```
    <khối lệnh 1>
```

```
    <lệnh nhảy>
```

```
case <giá trị 2> :
```

```
    <khối lệnh 2>
```

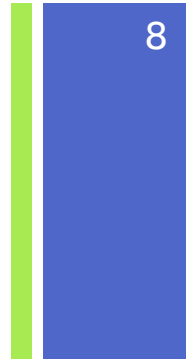
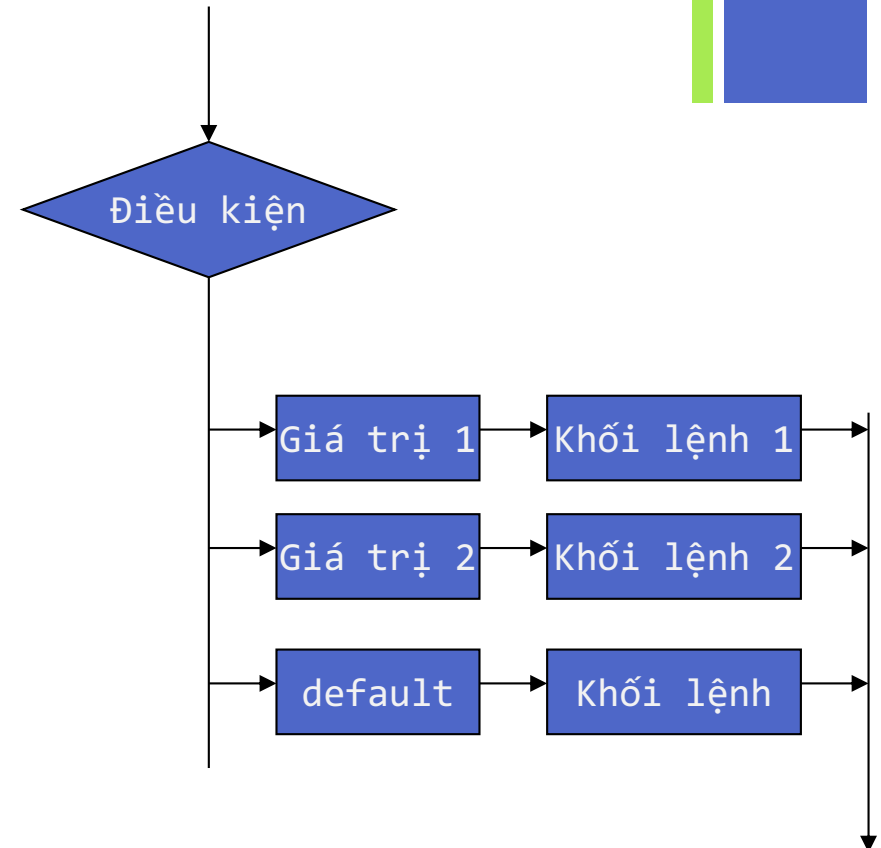
```
    <lệnh nhảy>
```

```
...
```

```
[default : a
```

```
    <khối lệnh default>]
```

```
}
```



- default là phần tùy chọn được thực hiện khi biểu thức điều kiện không khớp với phần case nào.
- Lệnh nhảy có thể là break, goto, ...

+ VÍ DỤ

- Cài đặt switch với lệnh nhảy break

```
switch (country) {  
    case "England":  
    case "USA":  
        language = "English"; break;  
    case "Germany":  
    case "Austria":  
    case "Switzerland":  
        language = "German"; break;  
    case null:  
        Console.WriteLine("no country specified"); break;  
    default:  
        Console.WriteLine("don't know the language of " +  
country);  
        break;  
}
```

+ THỰC HÀNH

Bài tập 2.4.1

10

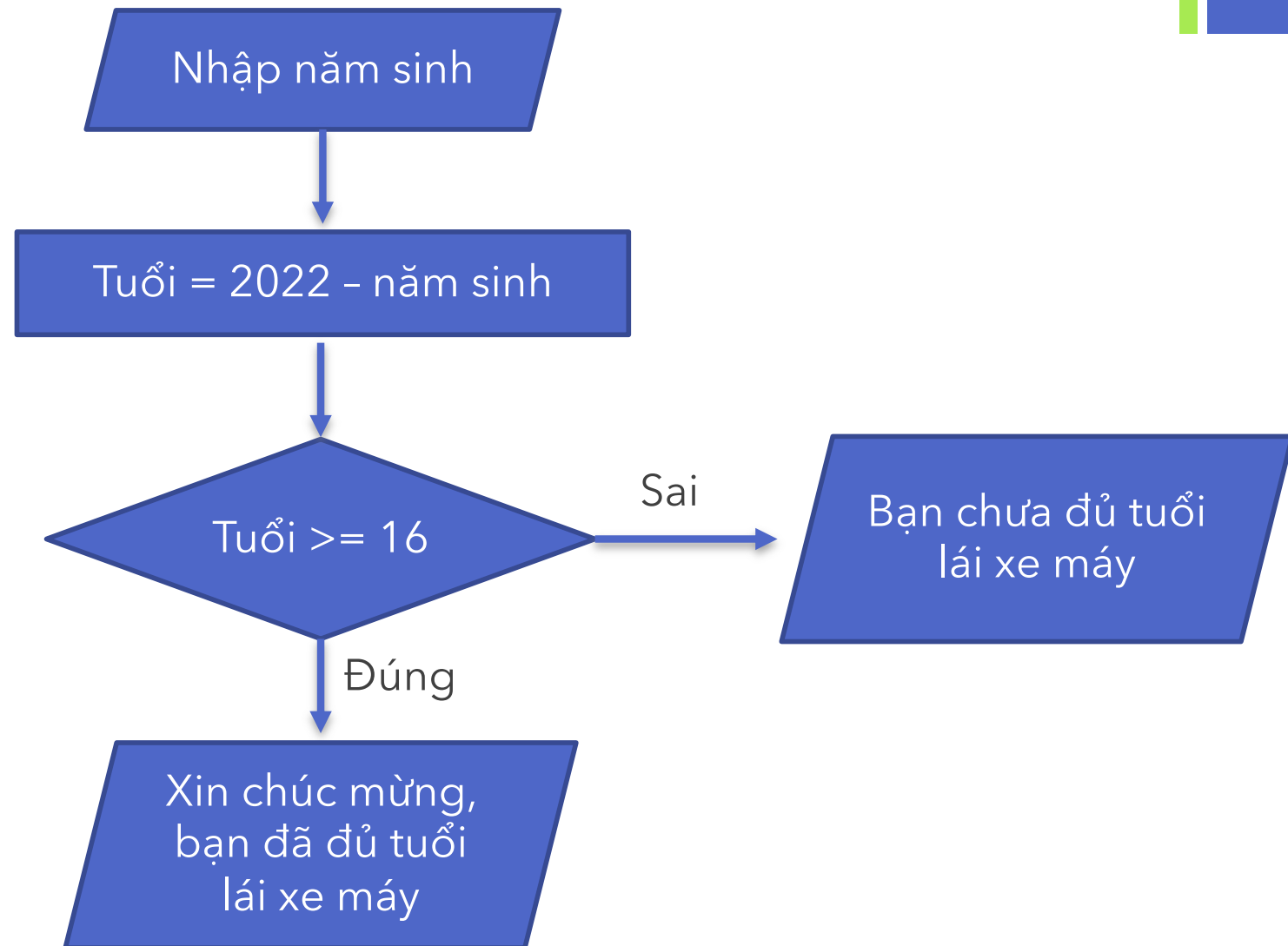
- Viết chương trình nhập vào năm sinh sau đó tính tuổi hiện tại. Biết năm hiện tại là 2022, nếu tuổi < 16 thì in ra màn hình thông báo “Bạn chưa đủ tuổi lái xe máy”. Ngược lại, in ra thông báo “Xin chúc mừng, bạn đã đủ tuổi lái xe máy”.

Đầu vào	Đầu ra
Nhập năm sinh: 2000	Bạn năm nay 22 tuổi Xin chúc mừng, bạn đã đủ tuổi lái xe máy
Nhập năm sinh: 2007	Bạn năm nay 15 tuổi Bạn chưa đủ tuổi lái xe máy

+ THỰC HÀNH

Bài tập 2.4.1

11



+ THỰC HÀNH

Bài tập 2.4.2

12

- Viết chương trình giải phương trình bậc nhất có dạng $ax + b = 0$. Với a, b là số thập phân nhập vào từ bàn phím.

Đầu vào	Đầu ra
Nhập a: 2 Nhập b: 5	Phương trình có nghiệm $x = -2.5$
Nhập a: 0 Nhập b: -6	Phương trình vô nghiệm

+ THỰC HÀNH

Bài tập 2.4.2

13

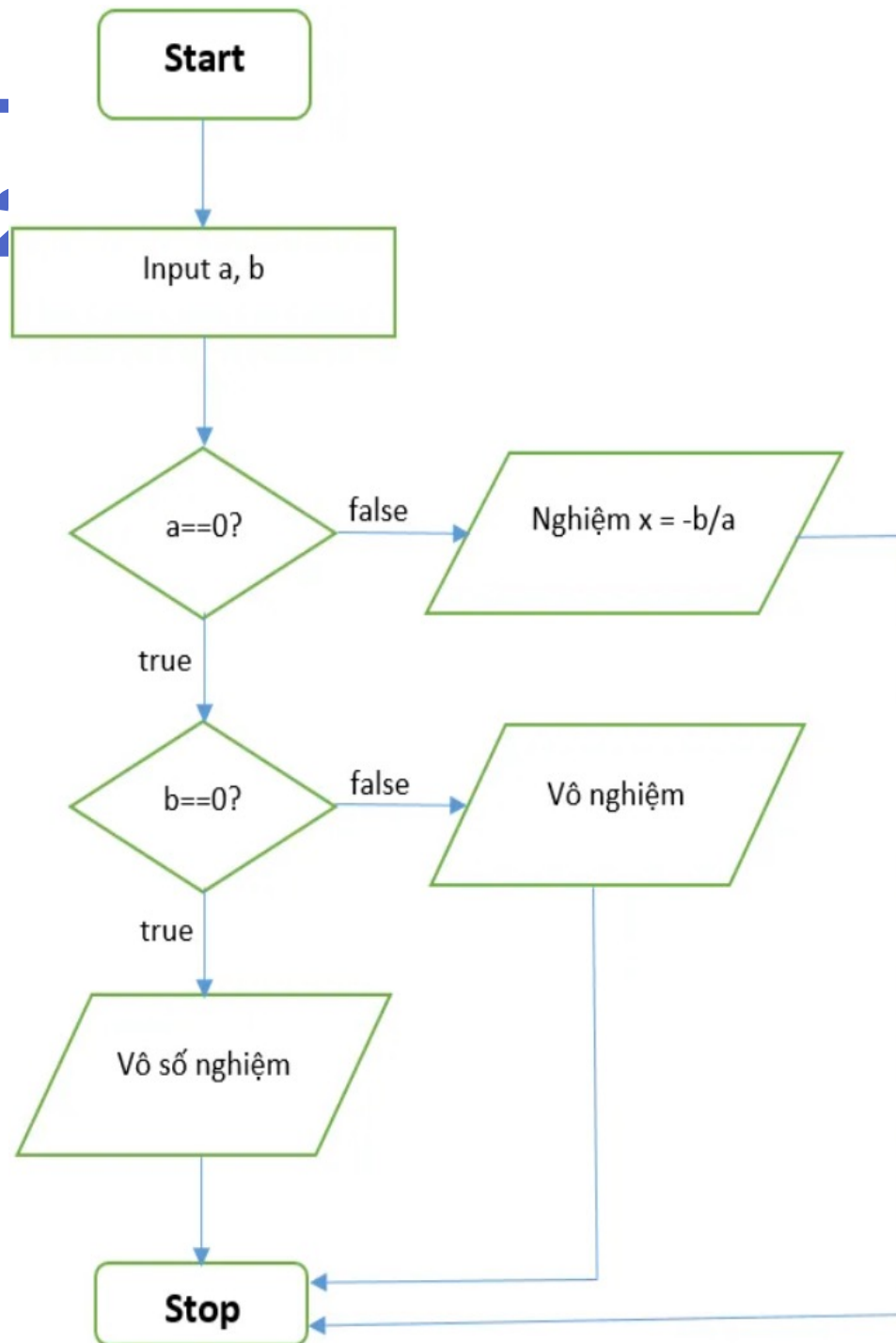
Gợi ý:

- Nếu $a == 0$:
 - Nếu $b == 0 \rightarrow$ Phương trình vô số nghiệm
 - Nếu $b \neq 0 \rightarrow$ Phương trình vô nghiệm
- Nếu $a \neq 0 \rightarrow$ Phương trình có nghiệm $x = -b/a$

+ THỰC HÀNH

Bài tập 2.4.1

14



+ THỰC HÀNH

Bài tập 2.4.3

15

Viết chương trình nhập vào điểm trung bình, phân loại học lực theo thang điểm sau:

- $8 \leq \text{điểm tb} \leq 10$: Giỏi
- $5 \leq \text{điểm tb} < 8$: Khá
- $0 \leq \text{Điểm tb} < 5$: Trung bình
- Khác: điểm tb không hợp lệ

+ THỰC HÀNH

Bài tập 2.4.3

- Kịch bản sử dụng:

Dữ liệu vào	Kết quả ra
Nhập điểm tb: 8	Học lực giỏi
Nhập điểm tb: 4	Học lực Trung bình
Nhập điểm tb: 14	Điểm tb không hợp lệ

+ THỰC HÀNH

Bài tập 2.4.4

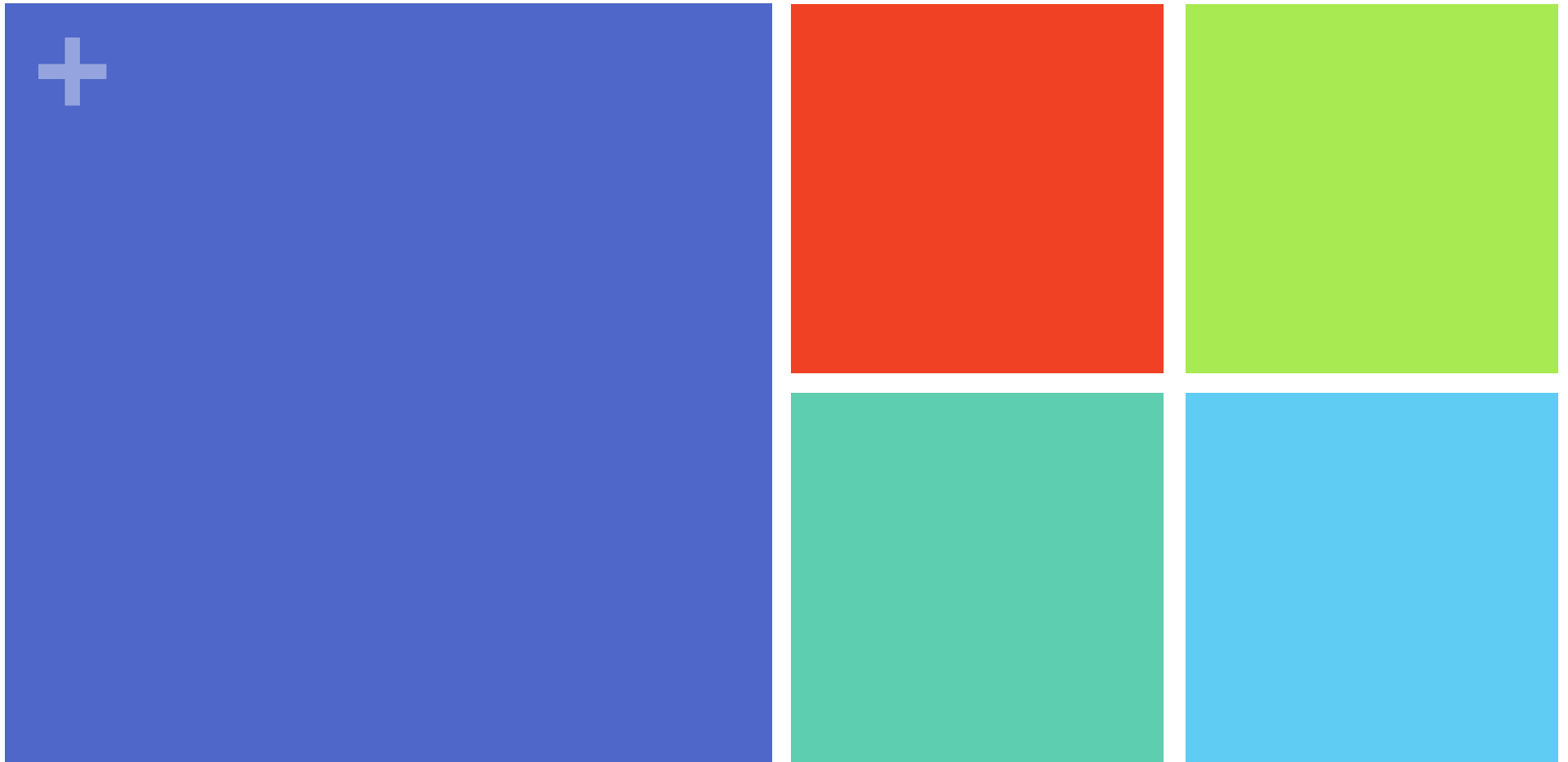
- Xây dựng trò chơi Kéo – Búa – Lá.
- Người dùng nhập vào 1 giá trị từ 1 đến 3
 - (1: Kéo | 2: Búa | 3: Lá).
- Máy tính sẽ đưa ra lựa chọn của mình. So sánh và đưa kết quả ra màn hình.

+ THỰC HÀNH

Bài tập 2.4.4

- Kịch bản sử dụng:

Dữ liệu vào	Kết quả ra
Nhập lựa chọn: 1	Bạn chọn Kéo Máy tính chọn Búa ==> Bạn đã thua
Nhập lựa chọn: 2	Bạn chọn Búa Máy tính chọn Kéo ==> Chúc mừng, bạn đã thắng!



CẤU TRÚC LẬP

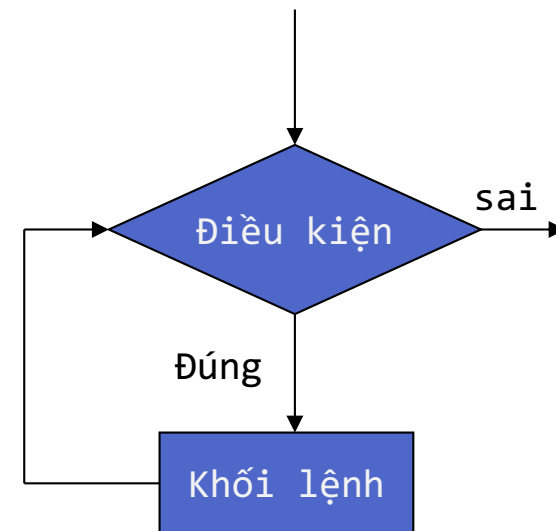
TS. ĐẶNG THÀNH TRUNG



LỆNH LẶP

20

- Lệnh **while** :
 - Ý nghĩa : Trong khi điều kiện đúng thì làm ...
 - Cú pháp :
while (biểu thức điều kiện)
 <khối lệnh>
 - Biểu thức điều kiện chỉ có giá trị đúng hoặc sai
 - Vòng lặp **while** luôn kiểm tra điều kiện trước khi thực hiện khối lệnh.
 - Nếu khối lệnh có nhiều hơn một lệnh thì các lệnh phải được đặt trong dấu {}

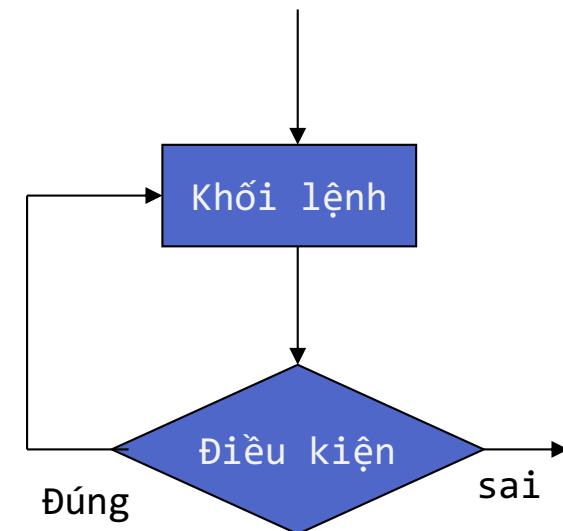




LỆNH LẶP

21

- Lệnh **do ... while** :
 - Ý nghĩa : Làm trong khi điều kiện đúng
 - Cú pháp :
do
 <khối lệnh>
while (biểu thức điều kiện)
- Biểu thức điều kiện chỉ có giá trị đúng hoặc sai
- Vòng lặp while luôn kiểm tra điều kiện trước sau khi thực hiện khối lệnh.
- Nếu khối lệnh có nhiều hơn một lệnh thì các lệnh phải được đặt trong cặp dấu {}



+ LỆNH LẶP

- Lệnh **for** :

- Gồm ba phần chính

- Khởi tạo biến đếm vòng lặp
 - Kiểm tra biến đếm, nếu đúng thì thực hiện khối lệnh trong vòng lặp
 - Thay đổi biến đếm

- Cú pháp :

`for ([phần khởi tạo];[biểu thức điều kiện];[thay đổi bước lặp])
 <khối lệnh>`

Ví dụ :

```
for (int i = 0; i<n; i++)
```

```
    sum += i;
```

- Nếu khối lệnh có nhiều hơn một lệnh thì các lệnh phải được đặt trong cặp dấu {}



LỆNH LẶP

- Lệnh **foreach** : Lặp trên một tập hợp hoặc một mảng
- Cú pháp
foreach (<kiểu tập hợp> <tên thành phần> in <tên tập hợp>)
 <khối lệnh>
- Ý nghĩa : Vòng lặp sẽ duyệt qua từng phần tử của toàn bộ tập hợp hoặc mảng.
- Ví dụ :

```
int[] a = {3, 17, 4, 8, 2, 29};  
foreach (int x in a) sum += x;
```



LỆNH NHẢY

24

- **break** : Dừng để “nhảy” ra khỏi lệnh lặp hoặc lệnh case.
- **continue** : Bỏ qua phần còn lại và tiếp tục vòng lặp tiếp theo.
- **goto** : Nhảy tới một nhãn xác định.
 - Không nhảy vào một khối lệnh
 - Không nhảy ra ngoài khối lệnh finally

+ THỰC HÀNH

Bài tập 2.4.5

25

- Viết chương trình nhập vào 2 số nguyên dương a, b từ bàn phím. Tính UCLN và BCNN của a, b và in ra màn hình.

Đầu vào	Đầu ra
Nhập a: 24 Nhập b: 16	UCLN = 8 BCNN = 48
Nhập a: 25 Nhập b: 18	UCLN = 1 BCNN = 450

+ THỰC HÀNH

Bài tập 2.4.6

- Viết chương trình cho phép tính số tiền thu được nếu gửi ngân hàng trong n tháng, lãi suất mỗi tháng là $x\%$ và số tiền gửi ban đầu là y đồng.

Đầu vào	Đầu ra
Nhập số tháng, $n = 10$ Lãi suất, $x = 7.2$ Số tiền gửi, $y = 10000000$	Tổng số tiền = 20042313 VNĐ
Nhập số tháng, $n = 12$ Lãi suất, $x = 0.5$ Số tiền gửi, $y = 60000000$	Tổng số tiền = 63700668 VNĐ

+ THỰC HÀNH

Bài tập 2.4.7

27

- Sử dụng vòng lặp viết chương trình in ra các bảng cửu chương từ 1 đến 10. Hình dưới là một ví dụ:

1 * 1 = 1	2 * 1 = 2	3 * 1 = 3	4 * 1 = 4	5 * 1 = 5
1 * 2 = 2	2 * 2 = 4	3 * 2 = 6	4 * 2 = 8	5 * 2 = 10
1 * 3 = 3	2 * 3 = 6	3 * 3 = 9	4 * 3 = 12	5 * 3 = 15
1 * 4 = 4	2 * 4 = 8	3 * 4 = 12	4 * 4 = 16	5 * 4 = 20
1 * 5 = 5	2 * 5 = 10	3 * 5 = 15	4 * 5 = 20	5 * 5 = 25
1 * 6 = 6	2 * 6 = 12	3 * 6 = 18	4 * 6 = 24	5 * 6 = 30
1 * 7 = 7	2 * 7 = 14	3 * 7 = 21	4 * 7 = 28	5 * 7 = 35
1 * 8 = 8	2 * 8 = 16	3 * 8 = 24	4 * 8 = 32	5 * 8 = 40
1 * 9 = 9	2 * 9 = 18	3 * 9 = 27	4 * 9 = 36	5 * 9 = 45

+ THỰC HÀNH

Bài tập 2.4.8

28

Xây dựng chương trình quay số trúng thưởng. Bạn được chọn 1 con số từ 1 đến 99.

- Máy tính sẽ quay thưởng tối đa 100 lần.
- Nếu con số máy tính quay trùng với con số bạn chọn thì dừng chương trình và in kết quả ra màn hình (Số lần quay thưởng).
- Hết 100 lần quay mà vẫn không ra số bạn chọn thì dừng và in kết quả ra màn hình.

+ THỰC HÀNH

Bài tập 2.4.8

■ Kịch bản sử dụng:

Đầu vào	Đầu ra
Nhập số của bạn: 22	Số bạn chọn: 22 Quay số lần 1: 15 ... Quay số lần n: 22 Chúc mừng bạn đã trúng thưởng sau n lần quay số...!
Nhập số của bạn: 90	Số bạn chọn: 90 Quay số lần 1: 15 ... Quay số lần 100: 27 Đã quay hết 100 lần. Chúc bạn may mắn lần sau!