Bài 3
Cấu trúc điều kiện

Module: BOOTCAMP PREPARATION

Kiểm tra bài trước

Hỏi và trao đổi về các khó khăn gặp phải trong bài “Biến, kiểu dữ liệu và toán tử"

Tóm tắt lại các phần đã học từ bài “Biến, kiểu dữ liệu và toán tử”

Mục tiêu

- Trình bày được câu lệnh điều kiện
- Mô tả cú pháp của câu lệnh if, if – else, if lồng nhau, if bậc thang
- Trình bày cú pháp câu lệnh switch-case
- So sánh giữa if bậc thang và switch-case
- Trình bày được biểu thức điều kiện
- Sử dụng được câu lệnh điều kiện if, if – else, if lồng nhau, if bậc thang
- Sử dụng được câu lệnh điều kiện switch-case
- Sử dụng được biểu thức điều kiện
- Sử dụng được từ kháo break, default

Thảo luận

Câu lệnh điều kiện

Câu lệnh if

Câu lệnh if-else

Các câu lệnh điều khiển

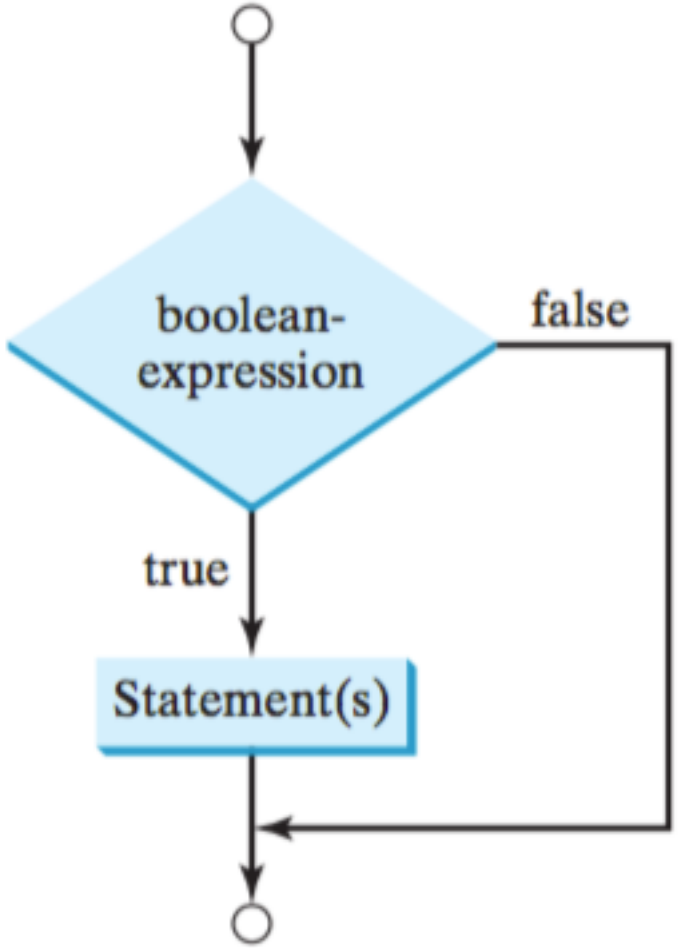
- Một chương trình phần mềm thực thi các câu lệnh theo trật tự từ trên xuống dưới
- Có thể thay đổi luồng thực thi của một chương trình bằng cách sử dụng các câu lệnh điều khiển luồng (*control flow statement*)
- Các câu lệnh điều khiển của JavaScript:
  - Câu lệnh điều kiện (conditional statement)
  - Câu lệnh lặp (Loop statement)
  - Câu lệnh nhảy (jump statement)

Câu lệnh điều kiện

- Câu lệnh điều kiện còn được gọi là câu lệnh ra quyết định (decision making)
- Cho phép thay đổi luồng thực thi của chương trình
- Lựa chọn thực thi một khối lệnh dựa trên việc đánh giá một điều kiện cho trước
- JavaScript hỗ trợ các câu lệnh điều kiện:
  - if
  - switch-case

Câu lệnh if

- Đánh giá một điều kiện và đưa ra lựa chọn thực thi một khối lệnh dựa trên điều kiện đó
- Nếu điều kiện có giá trị đúng (true) thì khối lệnh bên trong if sẽ được thực thi
- Nếu điều kiện có giá trị sai (false) thì khối lệnh bên trong if sẽ được bỏ qua. Luồng thực thi sẽ chuyển xuống ngay sau khối lệnh if.



Cú pháp câu lệnh if

- Cú pháp:

```
if (condition) {
    // one or more statements;
}
```

Trong đó:

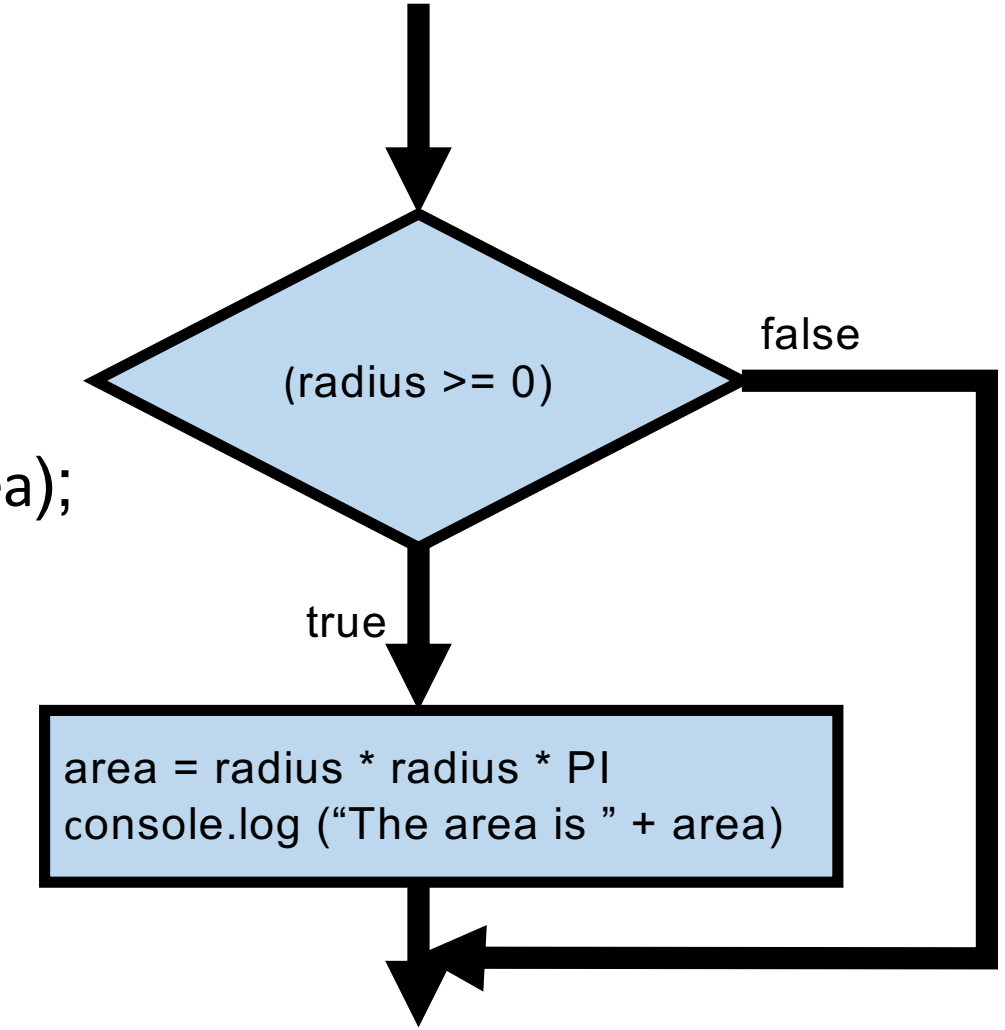
- `condition`: là biểu thức trả về giá trị kiểu boolean
- `statements`: Các câu lệnh sẽ được thực thi nếu điều kiện trả về **true**

Câu lệnh if: Ví dụ

- Tính diện tích của hình tròn nếu bán kính lớn hơn hoặc bằng 0:

```
if (radius >= 0) {
        area = radius * radius * PI;
        console.log("The area is: " + area);
}
```

- Nếu giá trị của radius nhỏ hơn 0 thì khối lệnh tính diện tích sẽ được bỏ qua



if với một câu lệnh bên trong

- Khi chỉ có một câu lệnh bên trong **if** thì có thể bỏ qua các dấu ngoặc
- Ví dụ:

```
if (i > 0) {
        console.log("i is positive");
}
```

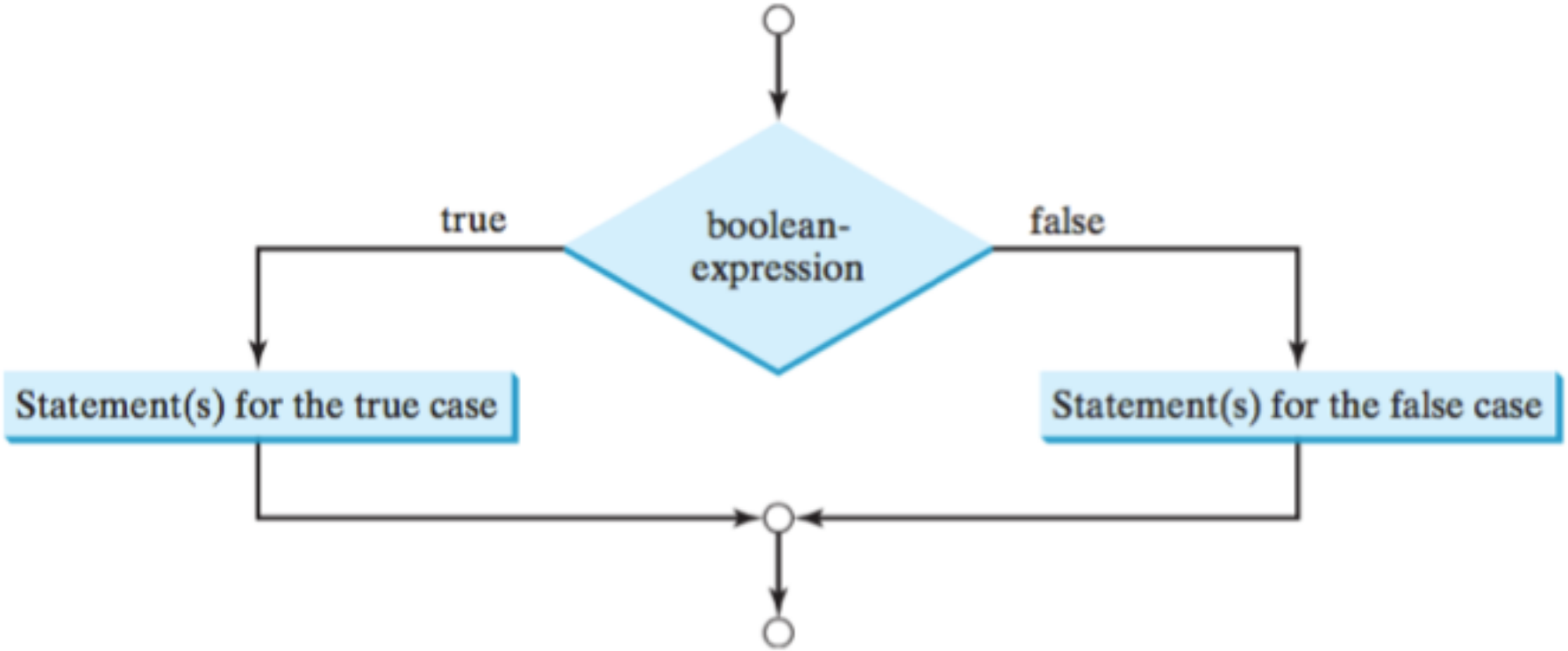
- Có thể được viết ngắn gọn:

```
if (i > 0)
        console.log("i is positive");
```

Lưu ý: Việc bỏ dấu ngoặc của if trong trường hợp này có thể giúp cho mã nguồn ngắn gọn hơn, tuy nhiên nó lại thiếu tính tường minh và ẩn chứa nguy cơ gây ra lỗi sau này. Chẳng hạn, sau này chúng ta thêm 1 dòng lệnh nữa vào if nhưng lại quên không bổ sung dấu ngoặc.

Câu lệnh if-else

- Câu lệnh if-else lựa chọn thực thi 1 trong 2 khối lệnh thông qua việc đánh giá một điều kiện



Cú pháp if-else

- Cú pháp:

```
if (condition) {
    // one or more statements;
}
else {
    // one or more statements;
}
```

Trong đó:

- condition: điều kiện để đánh giá. Nếu condition trả về **true** thì khối lệnh bên trong **if** được thực thi. Nếu condition trả về **false** thì khối lệnh trong **else** được thực thi.

if-else: Ví dụ

- Đoạn mã sau xác định một số là số chẵn hay là số lẻ:

```
if (number % 2 == 0){
        console.log(number + " is even.");
}
else{
        console.log(number + " is odd.");
}
```

if-else với một dòng lệnh bên trong

- Trong trường hợp chỉ có một dòng lệnh bên trong if hoặc else thì có thể bỏ dấu ngoặc
- Chẳng hạn:

```
if (number % 2 == 0)
        console.log(number + " is even.");
else
        console.log(number + " is odd.");
```

Demo

Lệnh if

Lệnh if-else

Thảo luận

Viết mã sạch

Câu lệnh if lổng nhau

If-else bậc thang

Lưu ý: Viết mã sạch (1)

- Trong ví dụ về kiểm tra tính chẵn/lẻ của một số nguyên, biểu thức điều kiện là “number % 2 == 0”, trong trường hợp này người đọc code cần suy luận để biết được ý nghĩa của biểu thức đó.
- Viết mã nguồn cũng như viết một bài văn, chúng ta cần viết thế nào để cho người đọc mã nguồn dễ dàng hiểu được ý nghĩa của từng dòng mã
- Một cách đơn giản để đạt được tính dễ hiểu đối với các biểu thức như trên đó là ”tách biến”: Đưa biểu thức của chúng ta ra bên ngoài khối lệnh if và gán giá trị của nó cho một biến (với tên gọi có ý nghĩa)

Lưu ý: Viết mã sạch (2)

- Có thể tái cấu trúc lại đoạn mã trước đây như sau:

```
var isEven = number % 2 == 0;
if (isEven)
    console.log(number + " is even.");
else
    console.log(number + " is odd.");
```

Với việc tách biểu thức ra và gán giá trị của nó cho biến *isEven* thì người đọc code rất dễ để hiểu rằng biểu thức trên là để kiểm tra xem một số có phải là chẵn hay không.

Câu lệnh if lồng nhau (nested if)

- Một câu lệnh if có thể được đặt trong câu lệnh if khác:

```
if(condition1) {
      if(condition2)
        true-block statement(s);
      else
        false-block statement(s);
      }
}
else {
        false-block statement(s);
}
```

if lồng nhau: ví dụ

```
if (a > b){
    if (a > c) {
        console.log("Greatest number is a = " + a);
    } else {
        console.log("Greatest number is c = " + c);
    }
} else {
    if (b > c){
        console.log("Greatest number is b = " + b);
    } else {
        console.log("Greatest number is c = " + c);
    }
}
```

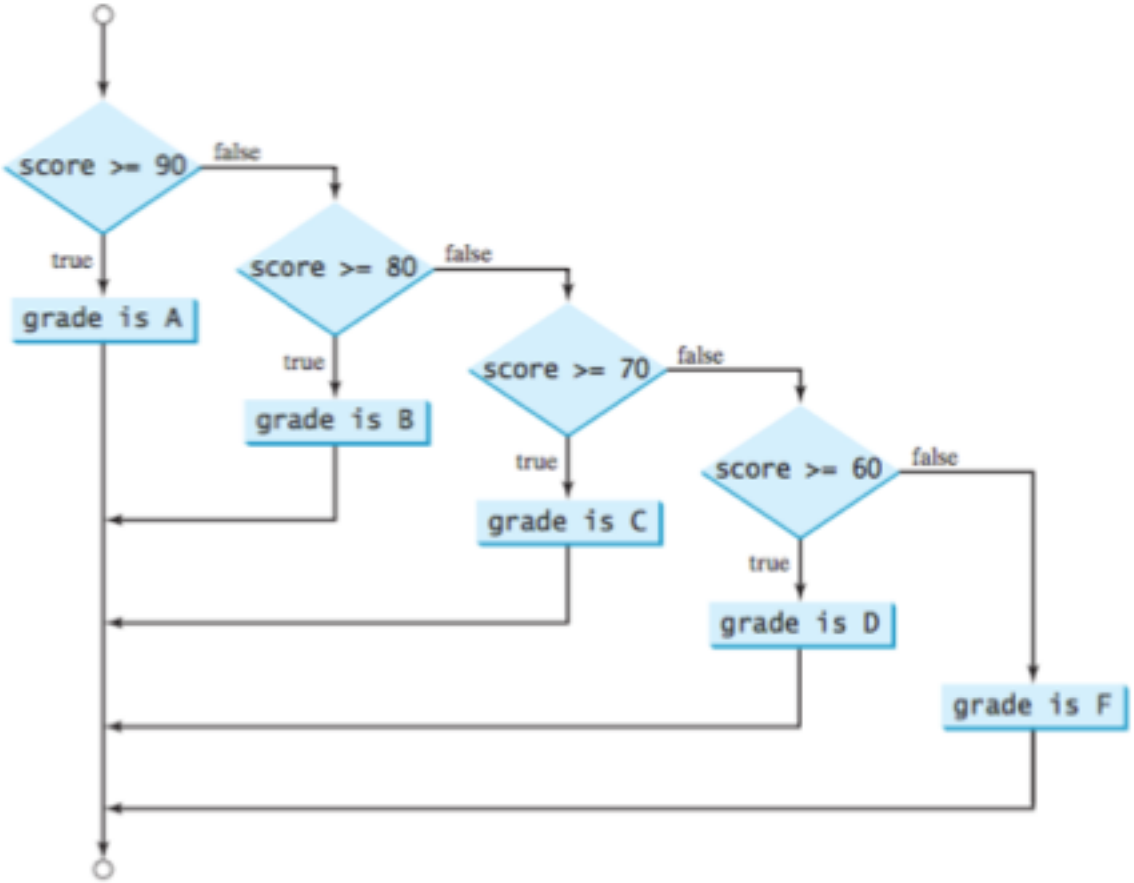
Câu lệnh if bậc thang

- Có thể đặt các câu lệnh điều kiện if-else liên tiếp nhau

```
if(condition) {
    // one or more statements
}
else if (condition) {
    // one or more statements
}
else {
    // one or more statements
}
```

if-else bậc thang: Ví dụ

```
if (score >= 90.0)
    console.log("A");
else if (score >= 80.0)
    console.log("B");
else if (score >= 70.0)
    console.log("C");
else if (score >= 60.0)
    console.log("D");
else
    console.log("F");
```



Demo

Lệnh if lồng nhau

Lệnh if bậc thang

Các lỗi thường gặp #1: Quên dấu ngoặc

- Ví dụ, đoạn mã sau sẽ thực thi sai (không hiển thị area):

```
if (radius >= 0)
    area = radius * radius * PI;
console.log("The area is " + area);
```

- Cần bổ sung dấu ngoặc:

```
if (radius >= 0) {
    area = radius * radius * PI;
    console.log("The area is " + area);
}
```

Các lỗi thường gặp #2: Dấu ; ở cuối

- Ví dụ: Đoạn mã sau sẽ thực thi sai ý đồ:

```
if (radius >= 0);
{
   area = radius * radius * PI;
   console.log("The area is " + area);
}
```

- Cần xoá bỏ dấu (;) ở cuối câu lệnh if:

```
if (radius >= 0)
{
   area = radius * radius * PI;
   console.log("The area is " + area);
}
```

Các lỗi thường gặp #3: So sánh giá trị boolean

- Đây thực chất không phải là một lỗi mà chỉ là việc so sánh không cần thiết
- Ví dụ:

```
if (even == true)
    console.log("It is even.");
```

- Sẽ tốt hơn nếu viết lại:

```
if (even)
    console.log("It is even.");
```

Các lỗi thường gặp #4: Nhầm lẫn khối if-else

- Ví dụ:

```
if (i > j)
    if (i > k)
        console.log("A");
else
        console.log("B");
```

- Câu lệnh else được áp dụng cho câu lệnh if thứ 2 (ở bên trong) chứ không phải câu lệnh if đầu tiên. Có thể viết lại chính xác hơn:

```
if (i > j)
    if (i > k)
        console.log("A");
  else
        console.log("B");
```

Các lỗi thường gặp #5: So sánh số thực

- Khi thao tác với các giá trị số thực (kiểu float hoặc double) thì có thể gặp những trường hợp mà độ chính xác không đúng
- Việc so sánh bằng (==) giữa hai số thực là không đáng tin cậy
- Ví dụ, đoạn mã sau sẽ hiển thị kết quả là false thay vì true như chúng ta thường nghĩ:

```
var x = 1.0 - 0.1 - 0.1 - 0.1 - 0.1 -
0.1;
console.log(x == 0.5);
console.log(x);
```

- Giá trị của biến **x** không phải là **0.5** mà là **0.5000000000000001**

So sánh kiểu số thực: Giải pháp

- So sánh gần đúng: Hiệu số của hai giá trị là một số rất nhỏ (*epsilon*)
- Thông thường:
  - Giá trị epsilon để so sánh giá trị float là 10^{-7}
  - Giá trị epsilon để so sánh giá trị double là 10^{-14}
- Ví dụ:

```
const EPSILON = 1E-14;
var x = 1.0 - 0.1 - 0.1 - 0.1 - 0.1 -
0.1;
if (Math.abs(x - 0.5) < EPSILON)
      console.log(x + " is approximately
0.5");
```

Các lỗi thường gặp #6: Gán giá trị boolean

- Đây không phải là một lỗi, mà là việc viết mã nguồn dài dòng không cần thiết
- Ví dụ:

```
if (number % 2 == 0)
    even = true;
else
    even = false;
```

- Có thể được viết ngắn gọn:

```
var even = number % 2 == 0;
```

Các lỗi thường gặp #7: Lặp code

- Đây không phải là một lỗi, mà là việc lặp mã nguồn không cần thiết, nên tránh
- Ví dụ:

```
if (inState) {
    tuition = 5000;
    console.log("The tuition is " + tuition);
} else {
    tuition = 15000;
    console.log("The tuition is " + tuition);
}
```

- Có thể sửa thành:

```
if (inState) {
    tuition = 5000;
} else {
    tuition = 15000; }
console.log("The tuition is " + tuition);
```

Thảo luận

switch-case

switch-case với kiểu dữ liệu String

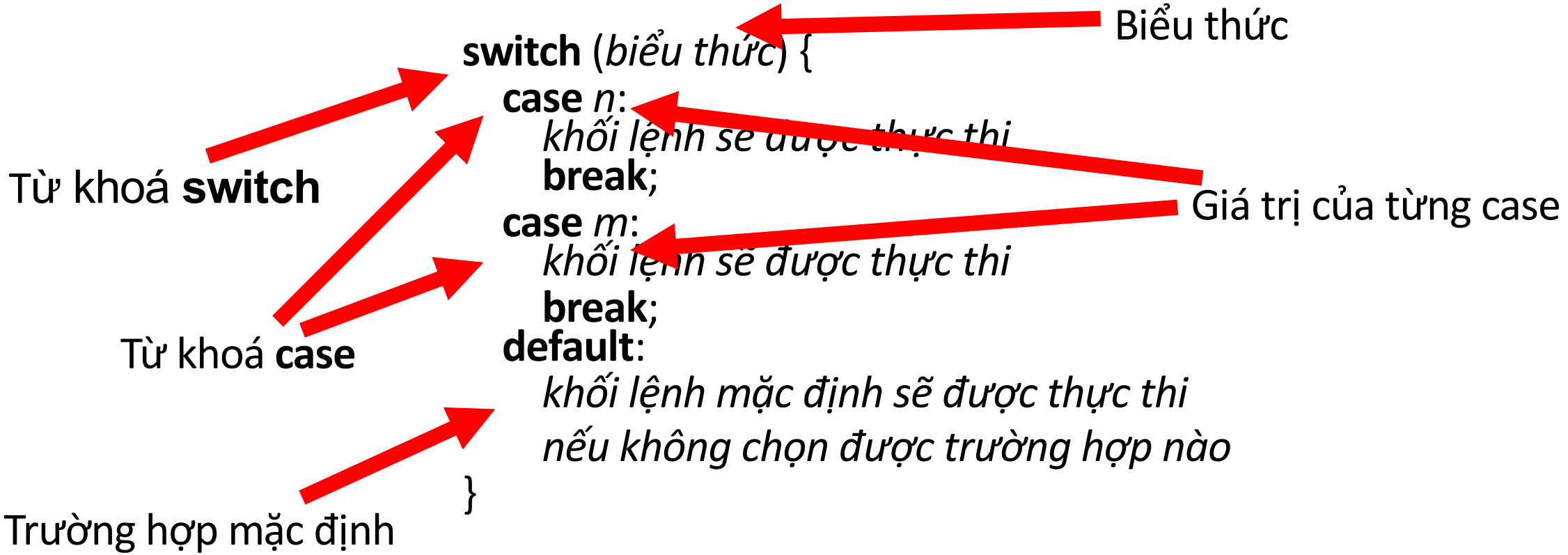
switch-case với kiểu enumeration

Từ khoá break và default

Câu lệnh switch-case

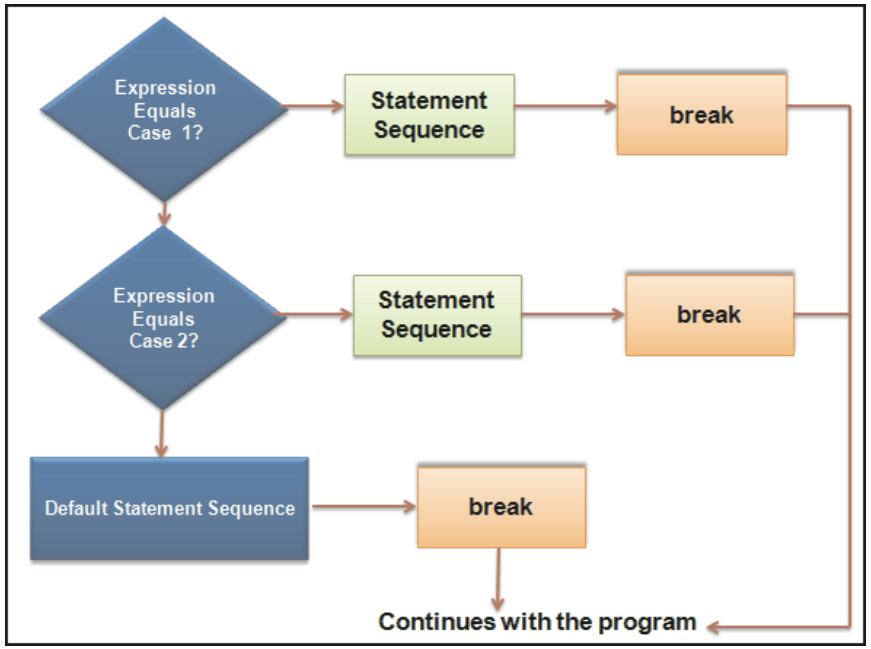
- switch-case là một cấu trúc điều kiện cho phép lựa chọn thực thi các khối lệnh khác nhau dựa trên kết quả của việc so sánh
- switch-case so sánh giá trị của một biến với lần lượt từng giá trị một, nếu có giá trị phù hợp với biến thì khối lệnh tương ứng sẽ được thực thi
- switch-case không thể thay thế if-else trong tất cả các trường hợp

Câu lệnh switch-case



switch-case: Mô tả

- Giá trị của biểu thức sẽ được so sánh với từng trường hợp (case)
- Nếu có trường hợp bằng nhau thì khối lệnh tương ứng sẽ được thực thi
- Nếu gặp câu lệnh break thì sẽ kết thúc thực thi khối switch-case
- Nếu gặp trường hợp bằng nhau, nhưng sau đó không có câu lệnh break thì tất cả những khối lệnh phía sau cũng được thực thi
- Nếu không có trường hợp nào bằng nhau thì khối lệnh trong default (nếu có) sẽ được thực thi



switch-case: Ví dụ tính thuế thu nhập cá nhân

```
switch (level) {
        case 0:
                //Tính thuế cho thu nhập đến 5tr
                break;
        case 1:
                //Tính thuế cho thu nhập từ 5tr đến 10tr
                break;
        case 2:
                //Tính thuế cho thu nhập từ 10tr đến 18tr
                break;
        ..... .....
                //Tính thuế cho thu nhập từ 18tr đến 32tr
                break;
        default:
                //Không hợp lệ
                break;
}
```

Gộp nhiều case

- Có thể lựa chọn thực thi một khối lệnh nếu giá trị bằng với một trong số nhiều case được gộp lại với nhau
- Ví dụ, phân loại ngày trong tuần với ngày cuối tuần:

```
switch  (day)  {
        case   1:
        case  2:
        case  3:
        case  4:
        case  5:

console.log("Weekday");
                break;
        case  0:
        case  6:

console.log("Weekend");
}
```

switch-case với kiểu dữ liệu String

- switch-case với kiểu dữ liệu String
- Ví dụ:

```
switch (greeting) {
    case "hello":
        console.log("You are speaking English"); break;
    case "hola":
        console.log("Usted está hablando español"); break;
    case "bonjour":
        console.log("Vous parlez français"); break;
    case "hallo":
        console.log("Du sprichst Deutsch"); break;
}
```

So sánh if và switch-case

if	switch-case
Có thể sử dụng để so sánh lớn hơn, nhỏ hơn…	Chỉ có thể sử dụng để so sánh bằng hoặc khác nhau
Mỗi câu lệnh if có một biểu thức điều kiện, với giá trị trả về là **true** hoặc **false**	Tất cả các trường hợp (case) đều so sánh với giá trị của biểu thức điều kiện duy nhất
Biểu thức điều kiện cần trả về giá trị kiểu **boolean**	Biểu thức điều kiện cần trả về giá trị là kiểu **byte**, **short**, **char**, **int**, hoặc **String**
Chỉ có một khối lệnh được thực thi nếu điều kiện đúng	Nếu điều kiện đúng mà không có câu lệnh break thì tất cả các khối lệnh ở phía sau cũng được thực thi

Lưu ý: Viết mã sạch (1)

- Trong ví dụ về phân loại ngày trong tuần, việc để các case là 0, 1, 2 … gây khó hiểu cho người đọc mã nguồn
- Có thể sử dụng phương pháp "tách hằng" để giúp cho mã nguồn dễ hiểu hơn
- Ví dụ:

```
const MONDAY = 1;
```

```
switch (day) {
        case 1:
        case 2:
        case 3:
        case 4:
        case 5:

console.log("Weekday");
                break;
        case 0:
        case 6:

console.log("Weekend");
}
```

Lưu ý: Viết mã sạch (2)

- Định nghĩa các hằng số:

```
const MONDAY = 1, TUESDAY = 2, WEDNESDAY = 3, THUSDAY = 4, FRIDAY = 5, SAT = 7, SUNDAY = 0;
```

- Viết lại switch-case:

```
switch (day) {
   case MONDAY:
   case TUESDAY:
   case WEDNESDAY:
   case THUSDAY:
   case FRIDAY:
      console.log("Weekday"); break;
   case SAT:
   case SUNDAY:
      console.log("Weekend");
}
```

Demo

switch - case

Thảo luận

Biểu thức điều kiện

Biểu thức điều kiện

- Biểu thức điều kiện đánh giá một biểu thức dựa vào một điều kiện cho trước
- Biểu thức điều kiện là một toán tử 3 ngôi
- Cú pháp:

condition ? expression_true_case : expression_false_case

Trong đó:

- *condition*: biểu thức điều kiện dùng để đánh giá
- *expression_true_case* : biểu thức sẽ được sử dụng trong trường hợp **true**
- *expression_false_case* : biểu thức sẽ được sử dụng trong trường hợp **false**

Biểu thức điều kiện: Ví dụ

- Tìm giá trị lớn nhất trong 2 số:

```
var maxValue = number1 > number2 ? number1 : number2;
```

- Tính số lượng dựa trên điều kiện khoảng cách:

```
var count = isHere ? getHereCount(index) : getAwayCount(index);
```

Tóm tắt bài học

- Các câu lệnh điều khiển giúp điều hướng luồng thực thi của ứng dụng
- Java hỗ trợ các câu lệnh điều khiển: điều kiện, lặp, nhảy
- Java hỗ trợ các câu lệnh điều kiện: if và switch-case
- Một số dạng khác của if:
  - if-else
  - if-else lồng nhau
  - if-else bậc thang
- switch-case là một cấu trúc điều kiện cho phép lựa chọn thực thi các khối lệnh khác nhau dựa trên kết quả của việc so sánh
- Cần lựa chọn sử dụng switch-case và if phù hợp với từng tình huống

Hướng dẫn

Hướng dẫn làm bài thực hành và bài tập

Chuẩn bị bài tiếp theo: *Câu lệnh lặp*