

Programação Multimédia

Seleção

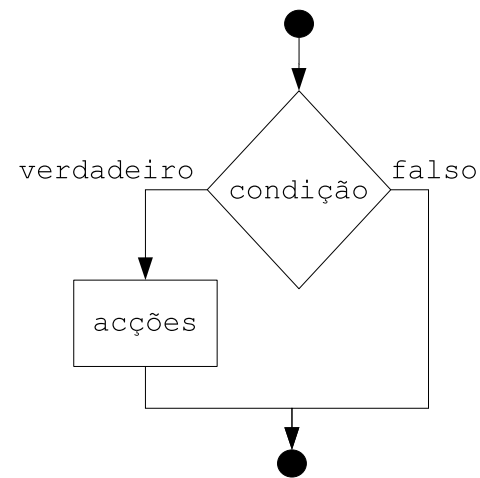
Seleccção do ramo de execução

- Em quase todos os programas é necessário determinar se uma condição é verdadeira antes de executar um conjunto de acções

If...

- Na sua forma mais simples a estrutura if resume-se a executar um conjunto de acções se determinada condição for verdadeira

```
if ( [condição] ) {  
    [acções]  
}
```



If...

- Exemplo

```
int x;

void setup() {
  size (200, 200);
  x = 0;
}

void draw() {
  background(255);

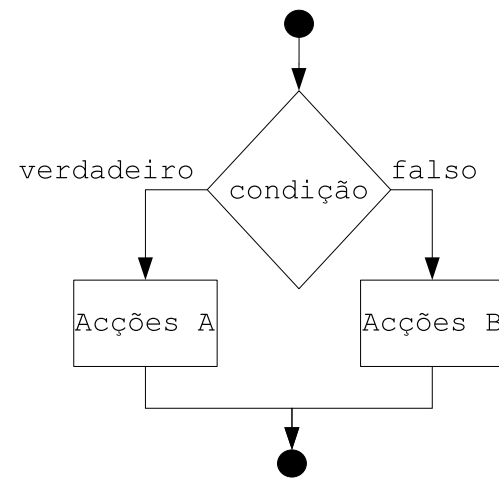
  point(x, 100);

  if (x < 100) {
    x = x + 1;
  }
}
```

If... else...

- O if também nos permite executar acções se a condição for falsa

```
if ( [condição] ) {  
    [acções A]  
} else {  
    [acções B]  
}
```



If... else...

- Exemplo

```
int x;

void setup() {
  size (200, 200);
  x = 0;
}

void draw() {
  background(255);

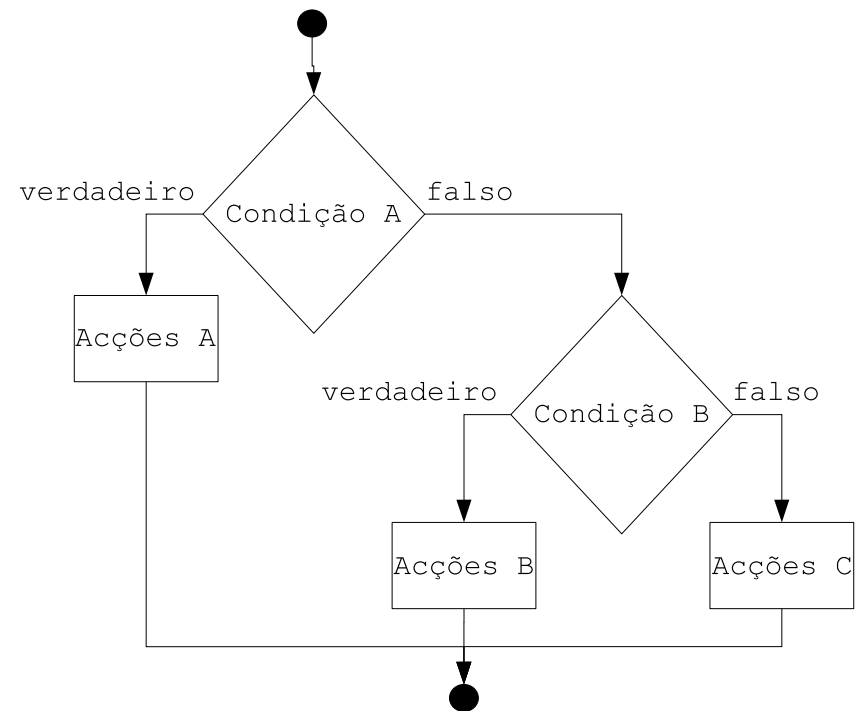
  point(x, 100);

  if (x < 100) {
    x = x + 1;
  } else {
    x = 0;
  }
}
```

If... else... if...

- É possível combinar ifs sequencialmente

```
if ([condição A]) {  
    [acções A]  
} else if ([condição B]) {  
    [acções B]  
} else {  
    [acções C]  
}
```



If... else... if...

- Exemplo

```
int x;

void setup() {
  size (200, 200);
  x = 0;
}

void draw() {
  background(255);

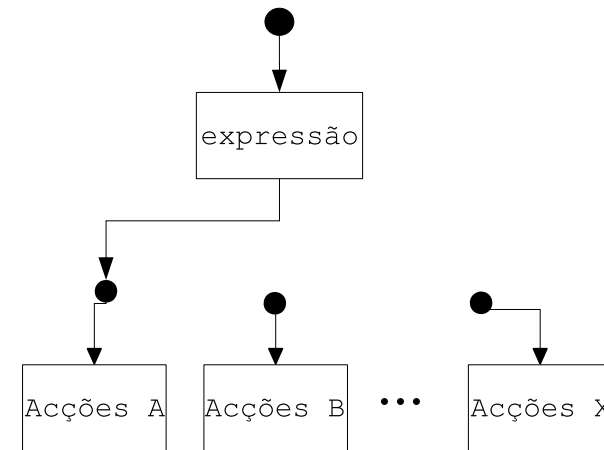
  point(x, 100);

  if (x < 100) {
    x = x + 1;
  } else if (x < 200){
    x = x + 5;
  } else {
    x = 0;
  }
}
```

Switch

- É possível escolher de forma mais prática o caminho a executar se a escolha for baseada num valor numérico

```
switch ([expressão]) {  
  case 1:  
    [Acções A]  
    break;  
  case 2:  
    [Acções B]  
    break;  
  case [X]:  
    [Acções X]  
    break;  
}
```



Switch

- Exemplo

```
int mes = 11;

switch (mes) {
case 1:
    System.out.println("Janeiro");
    break;
case 2:
    System.out.println("Fevereiro");
    break;
case 3:
    System.out.println("Março");
    break;
case 4:
    System.out.println("Abril");
    break;
case 5:
    System.out.println("Maio");
    break;
case 6:
    System.out.println("Junho");
    break;
```

```
case 7:
    System.out.println("Julho");
    break;
case 8:
    System.out.println("Agosto");
    break;
case 9:
    System.out.println("Setembro");
    break;
case 10:
    System.out.println("Outubro");
    break;
case 11:
    System.out.println("Novembro");
    break;
case 12:
    System.out.println("Dezembro");
    break;
default:
    System.out.println("Não é um mês
    Válido!");
    break;
}
```

Operadores Condicionais

- As condições usadas nos testes usam os operadores condicionais:
 - igual a: `==` (atenção!)
 - maior do que: `>`
 - menor do que: `<`
 - maior ou igual a: `>=`
 - menor ou igual a: `<=`
 - diferente: `!=`

Operadores Condicionais

- O resultado de um teste com operadores condicionais é um valor lógico (booleano):

```
boolean maior;
```

```
maior = 1 > 3;
```

Operadores Condicionais

- Exemplo

```
int idadeJoao;
int idadeAna;

idadeJoao = 10;
idadeAna = 10;

if (idadeJoao > idadeAna) {
    print("O João é mais velho do que a Ana. ");
}

if (idadeJoao < idadeAna) {
    print("O João é mais novo do que a Ana. ");
}

if (idadeJoao == idadeAna) {
    print("O João tem a mesma idade que a Ana. ");
}

if (idadeJoao != idadeAna) {
    print("O João não tem a mesma idade que a Ana. ");
}
```

Operadores Lógicos

- É possível testar condições complexas através dos operadores lógicos: operadores que combinam comparações simples:
- E : As duas comparações têm de ser verdadeiras ao mesmo tempo
- Ou : Pelo menos uma das comparações tem de ser verdadeira
- Não : Inverte o valor da comparação

Operadores Lógicos

- **Keywords**

- **E** : &&

```
if (x > 10 && x < 20) {  
    print ("Entre 10 e 20");  
}
```

- **Ou** : ||

```
if (x < 10 || x > 20) {  
    print ("Fora do intervalo [10; 20]");  
}
```

- **Não** : !

```
if (!(x < 10)) {  
    print ("X não é menor do que 10");  
}
```

Tabelas de Verdade - Ou

|| - (“ou”)

A	B	A B
False	False	False
False	True	True
True	False	True
True	True	True

Tabelas de Verdade - E

$\&\&$ - (“e”)

A	B	A && B
False	False	False
False	True	False
True	False	False
True	True	True

Tabelas de Verdade - Não

! – (“não”)

A

!A

False

True

True

False

Operadores Lógicos

- Exemplo

```
int x;

void setup() {
  size(200, 200);
  framerate(20);

  x = 0;
}

void draw() {
  background(0);

  if (x > 50 && x < 150) {
    fill(255, 0, 0);    //vermelho
  } else {
    fill(255, 255, 255); //branco
  }
  ellipse(x, 100, 10, 10);

  x = x + 5;
}
```

Projecto Semanal

1. Altere o Projecto 3.2 do capítulo anterior, de modo a que a linha se mantenha sempre dentro do ecrã e sempre animada.

Projecto 3.2: “Crie um programa que anime uma linha no ecrã. Irá ter de utilizar o método `draw()` (veja um dos exemplos e altere).

Pense nas variáveis que terá de utilizar, onde as deverá inicializar e onde as deverá actualizar. Não faz mal se a sua linha desaparecer do ecrã a certa altura, mas pense no que teria de fazer para evitar isso.”