

Processing

Seleção

Conceitos Básicos de Um Programa

- Memória (armazenamento de dados)
 - Guardar dados temporários (durante a execução do programa)
- **Seleção (...de caminhos de execução)**
 - **O nosso programa pode ter ramos que são executados em determinadas circunstâncias**
- Iteração (execução repetida das instruções)
 - Executar várias vezes a mesma acção (sobre dados diferentes)
- Módulos

Exemplo #1

```
float rectX;  
float rectY;
```

```
// inicializacao do programa
```

```
void setup() {  
    size(400, 400);  
    frameRate(10);
```

```
    rectX = 10;  
    rectY = 10;  
}
```

```
// desenho das varias frames
```

```
void draw() {  
    rect(rectX, rectY, 40, 40);  
  
    rectX = rectX + 3;  
}
```

- Como impedir o rectângulo de sair da janela?

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
- No exemplo anterior podemos ter um ramo verificar se o rectângulo “bateu” no lado direito ou esquerdo do ecrã

Exemplo #2

```
float rectX;
float rectY;
float velocidade;

// inicializacao do programa
void setup() {
  size(400, 400);
  frameRate(10);

  rectX = 10;
  rectY = 10;

  velocidade = 5;
}

// desenho das varias frames
void draw() {
  rect(rectX, rectY, 40, 40);

  rectX = rectX + velocidade;

  if (rectX > 400) {
    velocidade = -5;
  }

  if (rectX < 0) {
    velocidade = 5;
  }
}
```

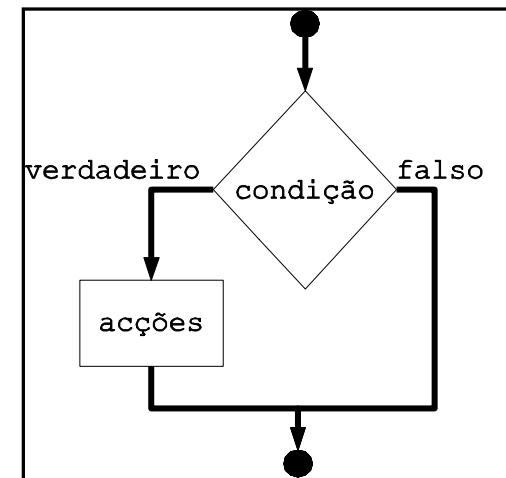
Exercício #1

- Modifique o Exemplo #2 de forma a que o rectângulo se mova na diagonal, sem sair do ecrã

Sintaxe do 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]  
}
```



Condição

- A condição é uma expressão do tipo **boolean**
 - Literal **true** ou **false**
 - Variável com o valor true ou false
 - Expressão com operadores condicionais

Operadores Condicionais

- As condições usadas nos testes podem usar os operadores condicionais:
 - igual a: == (atenção!)
 - maior do que: >
 - menor do que: <
 - maior ou igual a: >=
 - menor ou igual a: <=
 - diferente: !=
- O resultado de uma operação condicional é um valor **true** ou **false**

Operadores Condicionais

- O resultado de um teste com operadores condicionais é um valor lógico (booleano)
 - Estes operadores também podem ser utilizados nas expressões de atribuição:

```
boolean maior;
```

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

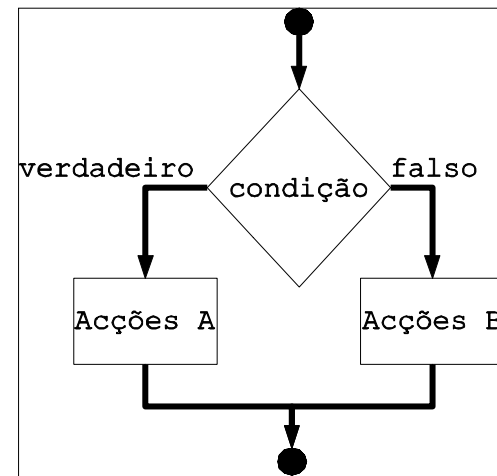
Variantes do if

- If... Else...
- If... Else if...
- If... Else if... Else...

If... else...

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

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



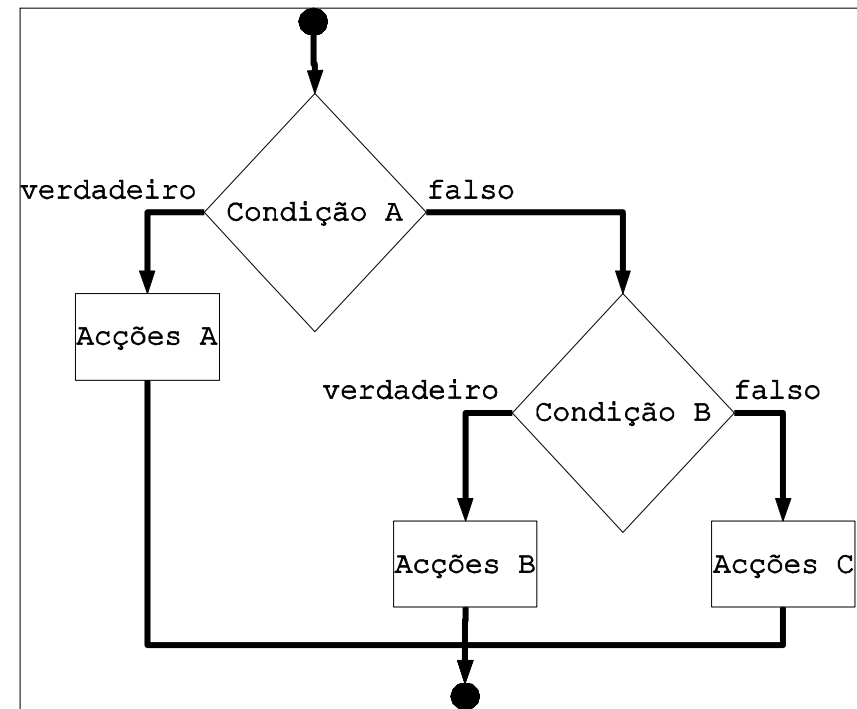
Exercício #2

- Modifique o Exercício #1, de forma a que a cor do rectângulo seja vermelha se o rectângulo estiver na metade esquerda do ecrã e azul se estiver na metade direita

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

- É possível combinar ifs sequencialmente

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



Exemplo #3 : If... else if...

```
float rectX;
float rectY;
float velocidadeX;
float velocidadeY;

// inicializacao do programa
void setup() {
  size(400, 400);
  frameRate(10);

  rectX = 10;
  rectY = 10;

  velocidadeX = 5;
  velocidadeY = 3;
}

// desenho das varias frames
void draw() {
  rect(rectX, rectY, 40, 40);

  rectX = rectX + velocidadeX;
  rectY = rectY + velocidadeY;

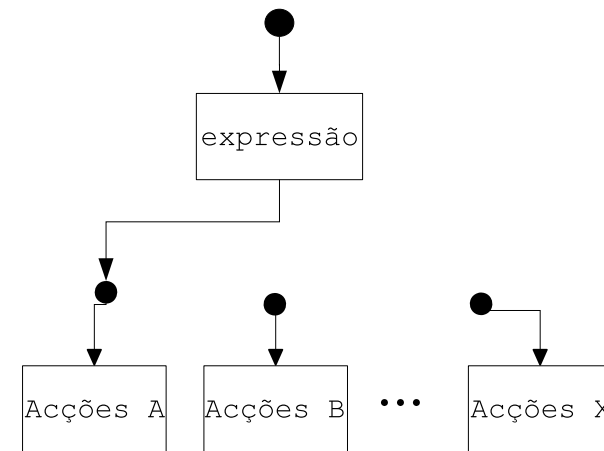
  if (rectX > 400) {
    velocidadeX = -5;
  } else if (rectX < 0) {
    velocidadeX = 5;
  }

  if (rectY > 400) {
    velocidadeY = -3;
  } else if (rectY < 0) {
    velocidadeY = 3;
  }
}
```

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 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

Exemplo #4 : Operadores Lógicos

```
float rectX;
float rectY;
float velocidadeX;
float velocidadeY;

void setup() {
  size(400, 400);
  frameRate(10);

  rectX = 10;
  rectY = 10;

  velocidadeX = 5;
  velocidadeY = 3;
}
```

```
// desenho das varias frames
void draw() {
  if (rectX < 50) {
    fill(255, 0, 0);
  } else if ( rectX > 50 && rectX < 100) {
    fill(255, 255, 0);
  } else if (rectX > 100 && rectX < 150) {
    fill(0, 255, 255);
  } else if (rectX > 150 && rectX < 200) {
    fill(255, 0, 255);
  } else {
    fill(255, 255, 255);
  }
  rect(rectX, rectY, 40, 40);
}
```

```
rectX = rectX + velocidadeX;
rectY = rectY + velocidadeY;

if (rectX > 400) {
  velocidadeX = -5;
}

if (rectX < 0) {
  velocidadeX = 5;
}

if (rectY > 400) {
  velocidadeY = -3;
}

if (rectY < 0) {
  velocidadeY = 3;
}
}
```

Exercício #3

- Modifique o Exemplo #4 de forma a eliminar os rectângulos brancos entre os coloridos

Projecto Semanal

- Parta do último exemplo da aula.
- Crie um programa que desenha formas gráficas diferentes consoante a posição no ecrã.
- Tente variar também a cor das formas.