

Programação Multimédia

Iteracção

Conceitos Básicos de Um Programa

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

Iteracção

- Executar um conjunto de instruções várias vezes
 - Não confundir com "interacção"
- A estrutura de programação que faz isto é o "ciclo"

Problema

- Animar 50 rectângulos no ecrã, ao mesmo tempo...
- 50 linhas com a instrução **rect(...)**?

Exemplo #1

```
float rectX;

float velocidadeX;

// inicializacao do programa
void setup() {
  size(500, 500);
  frameRate(10);

  rectX = 0;

  // os rectangulos movem-se
  // a 4 pixeis por frame
  velocidadeX = 4;
}

void draw() {
  int i;

  // desenhar 50 rectângulos verticalmente,
  // todos na mesma posicao horizontal
  for (i = 0; i < 50; i = i + 1) {
    rect(rectX, i*10, 5, 5);
  }

  rectX = rectX + velocidadeX;

  if (rectX > 500) {
    velocidadeX = -4;
  }

  if (rectX < 0) {
    velocidadeX = 4;
  }
}
```

Iteracção

- Duas classes de ciclos
 - Ciclos que executam enquanto uma condição é verdadeira
 - Ciclos que executam um número pré-determinado de vezes

Classes de Ciclos

- Ciclos que executam enquanto uma condição é verdadeira
 - `while`
 - `do`
- Ciclos que executam um número pré-determinado de vezes
 - `for`

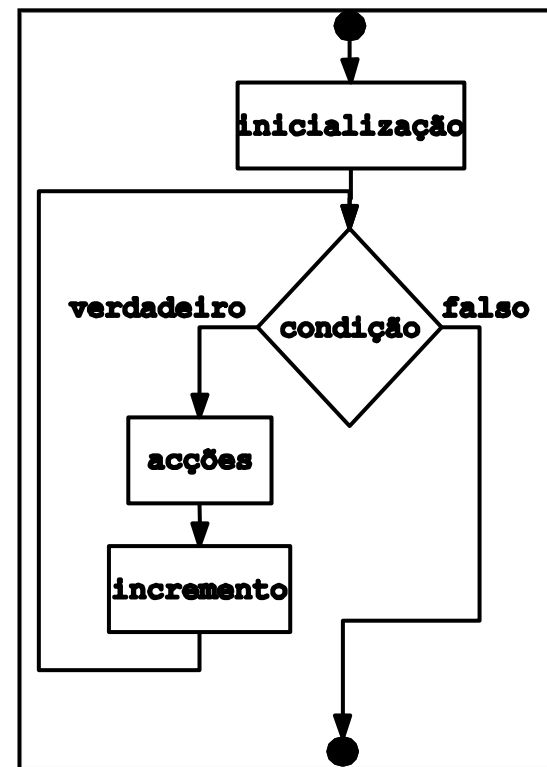
Exercício #1

- Altere o Exemplo #1, de forma a que sejam desenhados, em cada frame, 50 rectângulos com larguras crescentes
 - O primeiro rectângulo (o do topo) pode ter de largura 0 pixels e o último uma largura de 50 pixels

for

- Executa um número pré-determinado de vezes

```
for (<inicialização>; <condição>; <incremento>) {  
    <acções>  
}
```



for: uso típico

```
int i;  
for (i = 0; i < 10; i = i + 1) {  
  
}
```

- Variável de ciclo inicia-se em zero
- Condição simples com o limite de iterações
- Incremento da variável em uma unidade

for: uso típico compactado

```
for (int i = 0; i < 10; i++) {  
  
}
```

- Declaração da variável de ciclo no próprio ciclo
- Incremento da variável com operador ++

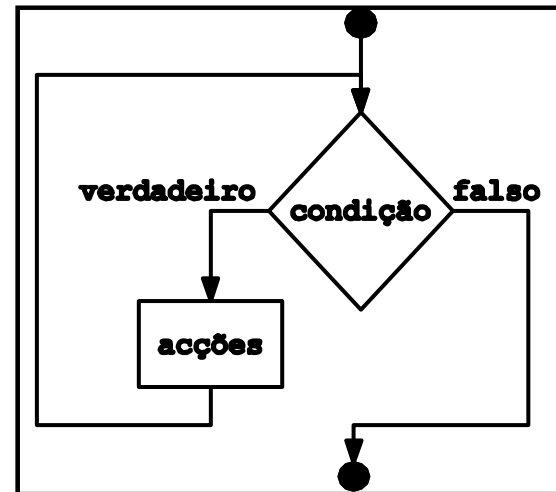
Exercício #2

- Modifique o Exercício #1 de forma a que os primeiros 25 rectângulos sejam desenhados com a cor vermelha e os últimos 25 a verde.

while

- Executa um conjunto de instruções enquanto uma condição for verdadeira
- O teste é feito **antes** da execução

```
while (<condição>) {  
    <acções>  
}
```



while - Exemplo

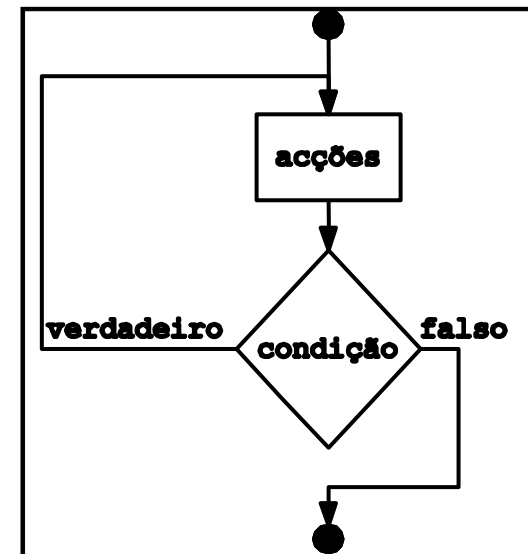
```
int i = 0;

while (i <= 10) {
    println(i);
    i = i + 1;
}
```

do

- Executa um conjunto de instruções enquanto uma condição for verdadeira
- O teste é feito **depois** da execução

```
do {  
    <acções>  
} while (<condição>);
```



do - Exemplo

```
int i = 0;

do {
    println(i);
    i = i + 1;
} while (i <= 10);
```

Exercício #3

- Modifique o Exercício #2, de forma a utilizar o ciclo `while`, em vez do `for`

Intervalo

Problema

- Animar 50 rectângulos, todos com velocidades diferentes

Exemplo #2 : Solução

```
• // posicao de cada rectangulo
• float rectX[];

• // velocidade de cada rectangulo
• float velocidadeX[];

• // inicializacao do programa
• void setup() {
•   size(500, 500);
•   frameRate(10);

•   // criar um vector de 50 floats para a posicao X
•   rectX = new float[50];

•   // criar um vector de 50 floats para a velocidade
•   velocidadeX = new float[50];

•   for (int i = 0; i < 50; i++) {
•     // os rectangulos movem-se com velocidades
•     // aleatorias, entre 1 e 4 pixeis por frame
•     velocidadeX[i] = random(1, 4);

•     // os rectangulos comecam todos em 0
•     rectX[i] = 0;
•   }
• }

• void draw() {
•   int i;

•   // desenhar 50 rectângulos verticalmente,
•   // todos na mesma posicao horizontal
•   for (i = 0; i < 50; i = i + 1) {
•     rect(rectX[i], i*10, 5, 5);
•     rectX[i] = rectX[i] + velocidadeX[i];

•     if (rectX[i] > 500) {
•       velocidadeX[i] = -velocidadeX[i];
•     }

•     if (rectX[i] < 0) {
•       velocidadeX[i] = -velocidadeX[i];
•     }
•     ;
•   }
• }
• }
```

Tipos Complexos

- **Vector:**
 - Uma forma de juntar várias variáveis do mesmo tipo
 - Um vector tem um nome e várias posições que podem ser acedidas através de um índice:
 - `idade[1] = 20`

Vectores

- `idade[1] = 20`
- Genericamente :
 - `nomeVariavel[indice]`
- Índice é uma expressão do tipo inteiro
 - Literal
 - `Idade[1]`
 - Variável
 - `Idade[i]`
 - Expressão complexa
 - `Idade[2*i+n]`

Vectores

- Para Quê?
 - Vamos supor que precisavamos de guardar as idades dos alunos de uma turma
- | | |
|-------------|----|
| – Pedro | 20 |
| – Rui | 19 |
| – Osvaldo | 19 |
| – Guilherme | 20 |
| – Romão | 18 |
| – Ana | 19 |
| – Maria | 18 |
| – Anacleto | 20 |
| – Carlos | 21 |
| – António | 17 |
| – Osório | 18 |
| – Manuela | 17 |
| – Ulmira | 18 |
| – Luís | 20 |
| – Tibério | 19 |
| – Irina | 20 |
| – Mafalda | 18 |
| – Emília | 19 |
| – Diogo | 20 |
| – Isabel | 19 |
| – Aleixo | 20 |

Vectores

- Solução:

```
int pedro      = 20;
int rui        = 19;
int osvaldo    = 19;
int guilherme  = 20;
int romão     = 18;
int ana        = 19;
int maria      = 18;
int anacleto   = 20;
int carlos     = 21;
int antónio    = 17;
int osório     = 18;
int manuela    = 17;
int ulmira     = 18;
int luís       = 20;
int tibério    = 19;
int irina      = 20;
int mafalda    = 18;
int emília     = 19;
int diogo      = 20;
int isabel     = 19;
int aleixo     = 20;
```

- E agora, se quiséssemos adicionar 5 anos a cada aluno?
- E calcular a média de idades?

Vectores

- Adicionar 5 anos:

```
pedro      = pedro      + 5;
rui        = rui        + 5;
osvaldo    = osvaldo    + 5;
guilherme  = guilherme  + 5;
romão      = romão      + 5;
ana        = ana        + 5;
maria      = maria      + 5;
anacleto   = anacleto   + 5;
carlos     = carlos     + 5;
antónio    = antónio    + 5;
osório     = osório     + 5;
manuela    = manuela    + 5;
ulmira     = ulmira     + 5;
luís       = luís       + 5;
tibério    = tibério    + 5;
irina      = irina      + 5;
mafalda    = mafalda    + 5;
emília     = emília     + 5;
diogo      = diogo      + 5;
isabel     = isabel     + 5;
aleixo     = aleixo     + 5;
```

- Calcular a média:

```
float media;

media = (pedro + rui + osvaldo +
        guilherme + romão + ana + maria +
        anacleto + carlos + antónio + osório +
        manuela + ulmira + luís + tibério +
        irina + mafalda + emília + diogo +
        isabel + aleixo)/21;
```

•Tem de haver uma forma mais prática!!

Vectores

- Podemos usar um vector para guardar as idades:
 - A inicialização continua a ser trabalhosa (não há nada a fazer neste caso)

```
int idades[];  
  
idades = new int[21];  
  
idades[0] = 20; // pedro  
idades[1] = 19; // rui  
idades[2] = 19; // osvaldo  
idades[3] = 20; // guilherme  
idades[4] = 18; // romão  
idades[5] = 19; // ana  
idades[6] = 18; // maria  
idades[7] = 20; // anacleto  
idades[8] = 21; // carlos  
idades[9] = 17; // antónio  
idades[10] = 18; // osório  
idades[11] = 17; // manuela  
idades[12] = 18; // ulmira  
idades[13] = 20; // luís  
idades[14] = 19; // tibério  
idades[15] = 20; // irina  
idades[16] = 18; // mafalda  
idades[17] = 19; // emília  
idades[18] = 20; // diogo  
idades[19] = 19; // isabel  
idades[20] = 20; // aleixo
```

Vectores

- Mas adicionar 5 anos e calcular a média é muito mais fácil!

- Média:

```
int i;
float media = 0;

for (i = 0; i < 21; i = i + 1) {
    media = media + idades[i];
}
```

- Adicionar 5 anos:

```
int i;
for (i = 0; i < 21; i = i + 1) {
    idades[i] = idades[i] + 5;
}
```

Declarar e Inicializar Vectores

- `<tipo> nomeVector [] ;`
 - Declara um vector (variável) do tipo `<tipo>`
- `<tipo>` é **qualquer** tipo de dados
 - Pode mesmo ser do tipo vector!
- Antes de podermos usar o vector temos de indicar o seu tamanho:
 - `nomeVector = new <tipo> [<tamanhoVector>]`

Declarar e Inicializar Vectors – Exemplo

- Um vector para indicar quais os alunos, de uma turma de 100, passaram um teste:

```
boolean passouTeste[];
```

```
passouTeste = new boolean[100];
```

Percorrer Vectores: Exemplo

- Aqui entra o ciclo `for`!

```
int posicaoY[] = new int[100];
```

```
for (int i = 0; i < 100; i = i + 1) {  
    posicaoY[i] = posicaoY[i] + 1;  
}
```

Percorrer Vectores: Genericamente

```
for (int i = 0; i < [tamanhoVector]; i = i + 1) {  
    [aceder/escrever no vector]  
}
```

Exercício #4

- Modifique o Exemplo #2 de forma a que os rectângulos se movam também no eixo dos yy

Projecto Semanal

- Tente recriar o seguinte programa
– [ver site da disciplina]