

Programação Multimédia

Classes e Objectos

Conceitos Avançados

- Interacção (Rato e Teclado)
- **Classes e Objectos**
- Imagem, 3D e Captura de Vídeo
- Movimentos Reais
- Comunicação OSC
- DiABlu

Objectos

- Uma forma diferente de pensar nos programas
- Em vez de os dados e o comportamento de todo o programa serem definidos num ficheiro de código corrido podemos estruturar os elementos em “objectos”
- Objectos contêm **dados e comportamento**

Objectos

- Até agora os dados dos nossos programas podiam ser
 - simples (int, float, boolean)
 - complexos(vectores ou matrizes)
- A linguagem Processing (Java) é o que se chama uma linguagem orientada por objectos
- A partir de agora, os nossos “dados” podem agregar dados simples e operações sobre esses dados

Problema

- Criar um programa que desenha veículos em movimento no ecrã
 - O utilizador pode criar novos veículos com o pressionar de uma tecla
 - Cada veículo tem associado
 - Cor da carroçaria
 - Cor das rodas
 - Velocidade
 - Posição
 - Queremos controlar os veículos
 - Mandando parar
 - Mandando avançar
 - Dizendo a que velocidade deve avançar
 - Pintando-o

Exemplo #1

- Ver programa

Classes

- Os seguintes “objectos”
 - Opel Corsa 52-44-XR
 - Opel Corsa 34-49-AF
 - Opel Corsa 87-23-HJ
 - Opel Corsa 23-62-RB
- São objectos diferentes, mas todos “iguais”
- São todos da classe Opel Corsa

Classes

- Uma classe é definida por um conjunto de propriedades e comportamentos
 - Todos os objectos de uma mesma classe exibem o mesmo comportamento e o mesmo tipo de propriedades
 - Objectos diferentes podem ter valores específicos diferentes, para o mesmo tipo de propriedade
 - Cor diferente
 - Matrícula diferente

Classes e Objectos

- Na prática, uma classe funciona como um tipo de dados
 - Um objecto, funciona como um valor específico
 - Tal como temos várias variáveis do mesmo tipo, podemos ter vários objectos da mesma classe

Classes - Sintaxe

```
class NomeDaClasse {  
    [declaração de variáveis]  
  
    // construtor sem parametros  
    NomeDaClasse() {  
    }  
  
    // construtor com parametros  
    NomeDaClasse(<tipo> nomeParametro, etc) {  
    }  
  
    [metodos]  
}
```

Classes - Sintaxe

- O construtor é um método especial que serve para inicializar as variáveis do objecto construído
 - Tem o mesmo nome da classe
 - Pode ter parâmetros
 - Podemos ter vários construtores com parâmetros diferentes
 - Na criação do objecto usamos apenas um

Objectos

- Ao processo de criação de um objecto de uma determinada classe chama-se ***instanciação da classe***:

```
NomeDaClasse nomeVariavel;
```

Construtor
sem
parâmetros

```
nomeVariavel = new NomeDaClasse();
```

Ou

```
nomeVariavel = new NomeDaClasse(parametros);
```

Construtor
com
parâmetros

Objectos

- Os objectos têm dados (variáveis) e métodos
- Para lermos o valor de uma variável de um objecto:
`nomeDoObjecto.nomeDaVariável`
- Para invocarmos um método:
`nomeDoObjecto.nomeDoMétodo()`

Exercício #1

- Modifique o Exemplo #1 de forma criar um programa com 50 carros no ecrã.
 - Utilize um vector de objectos do tipo Veiculo

Texto (String)

- Em Processing, o texto é representado por uma classe: `String`

```
String nomeUtilizador;  
nomeUtilizador = new String("Jorge");  
print(nomeUtilizador);
```

String

- Métodos da classe String:
 - `charAt()` - Retorna o carácter no índice especificado.
 - `equals()` - Compara a String com outra.
 - `indexOf()` - Retorna o índice da primeira ocorrência de um carácter na String.
 - `length()` - Retorna o número de caracteres da String.
 - `substring()` – Retorna uma String que é uma parte da original.
 - `toLowerCase()` - Converte todos os caracteres para minúsculas.
 - `toUpperCase()` – Converte todos os caracteres para maiúsculas.

Exemplo #2

```
String nome;
```

```
void setup() {  
    nome = new String("Programa");  
}
```

```
void draw() {  
  
}
```

Exercício #2

- Complete o Exemplo #2 de forma a que no método `draw()`, os caracteres da String nome sejam escritos um por frame
 - Use a instrução `println()`

Frame	Carácter
1	'P'
2	'r'
3	'o'
4	'g'
5	'r'
6	'a'
7	'm'
8	'a'

Exercício #3

- Modifique o Exercício #2, de forma a que em cada frame seja escrito um carácter de uma posição aleatória da String.

String

- A String é um caso especial em Processing...
- `String nome = new String("antonio");`
=
- `String nome = "antonio";`

Projecto Semanal

- Modifique o Exercício #1 de forma a que o utilizador possa parar o movimento de todos os veículos pressionando a tecla “espaço”.