

Programando em Java

(Classes Simples e Tipos de Dados Básicos)

Lista de Exercícios

**Material didático elaborado pelas diferentes equipas de
Introdução à Programação**

Luís Caires (Responsável), Armanda Rodrigues, António Ravara, Carla Ferreira, Fernanda Barbosa, Fernando Birra, Jácome Cunha, João Araújo, Miguel Goulão, Miguel Pessoa Monteiro, e Sofia Cavaco.

Mestrado Integrado em Engenharia Informática FCT UNL

Objectivos

- O aluno deverá ser capaz de:
 - A partir de uma especificação simples, em língua natural, identificar a classe a definir e propor uma interface para essa classe .
 - Usar o Eclipse na construção de classes simples, definindo:
 - Variáveis de instância (privadas);
 - Construtor de instâncias da classe (público);
 - Operações modificadoras (públicas);
 - Operações de consulta (públicas).
 - Usar operações aritméticas simples na implementação da classe, que envolvam valores lógicos.
 - Testar cuidadosamente as classes desenvolvidas (classe Main no método main), analisando com espírito crítico os resultados produzidos pelos testes.
 - Interpretando correctamente as eventuais mensagens de erro resultantes de defeitos no código produzido.

Sensor de movimento

Sensor de movimento

- **Objectivo**
 - Simular um sensor de movimento.
- **Descrição**
 - Um sensor de movimento é capaz de detectar movimento. O seu estado pode ser: “em movimento” ou “parado”.
- **Funcionalidades**
 - A deteção de movimento ou não é realizada através das operações de “move” e “stop”, respectivamente.
 - Quando o sensor é criado o seu estado é “parado”.
 - Deve ser sempre possível consultar se o estado é “em movimento” com a operação “isMoving”.
- **Interacção com o utilizador**
 - Após criar um sensor, pode invocar as operações do sensor.

Sensor de movimento

- **Que objecto se deve definir?**
 - Um sensor de movimento (classe Sensor).
- **Interface:**

```
void move()  
    Detectar movimento  
void stop()  
    Detectar ausência de movimento  
boolean isMoving()  
    Consultar se há movimento detectado
```

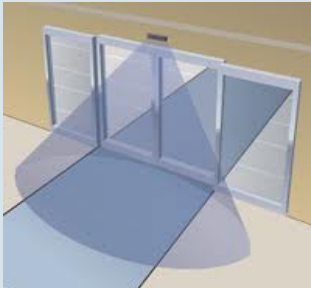
Sensor de movimento

(Cenário a testar)

```
Sensor s1 = new Sensor();  
Sensor s2 = new Sensor();  
s1.isMoving();  
// false (boolean)  
s2.isMoving();  
// false (boolean)  
s1.move();  
s1.isMoving();  
// true (boolean)  
s2.isMoving();  
// false (boolean)  
s2.stop();  
s2.isMoving();  
// false (boolean)  
s2.move();  
s2.isMoving();  
// true (boolean)  
s2.stop();  
s2.isMoving();  
// false (boolean)
```

Sensor de movimento

- Defina em Java uma classe Sensor, cujos objectos são sensores de movimento. Estes sensores são usados, por exemplo, em edifícios inteligentes, como forma de poupar energia, abrir automaticamente portas, ou detectar intrusos, em sistemas de entretenimento, etc
- Programe a sua classe no Eclipse
- Teste vários objectos da classe Sensor e verifique que se comportam tal como esperado (Cenário anterior).



FairPlay

FairPlay

- **Objectivo**

- Simular o estado de espírito de um adepto de futebol.

- **Descrição**

- O estado de espírito de um adepto de futebol pode ser “feliz” ou “deprimido”.
- O adepto apenas está feliz na sequência da vitória da própria equipa, ou do empate ou derrota de uma equipa rival. Tudo o resto é um drama para o adepto (e um caso de polícia evidente), deixando o adepto deprimido.

- **Funcionalidades**

- O adepto é sempre informado dos resultados dos jogos da sua equipa e das equipas rivais: ganhou, empatou ou perdeu.
- Quando o adepto é criado o seu estado de espírito é “feliz”.
- Deve ser sempre possível consultar se o adepto está “feliz”.

- **Interacção com o utilizador**

- Após criar um adepto, pode invocar as operações.

FairPlay

- **Interface** (classe FairPlay) :

boolean isHappy()

Devolve true se o adepto está feliz, false caso contrário.

void weWon()

A equipa do adepto venceu, ele fica feliz.

void weDrew()

A equipa do adepto empatou, ele não está feliz.

void weLost()

Trágica derrota, o adepto não está feliz.

FairPlay

- **Interface** (classe FairPlay) :

```
...  
void theyWon()  
    Lamentável vitória de um rival, o adepto fica triste.  
  
void theyDrew()  
    Finalmente o rival perdeu pontos, o adepto fica contente.  
  
void theyLost()  
    Glorioso momento, o rival está de rastos e o adepto fica  
feliz.
```

FairPlay

```
FairPlay b = new FairPlay();  
b.happy();  
// true      (boolean)  
b.weLost();  
b.isHappy();  
// false    (boolean)  
b.theyLost();  
b.isHappy();  
// true     (boolean)  
b.weLost();  
b.isHappy();  
// false    (boolean)  
b.weDrew();  
b.isHappy();  
// false    (boolean)  
b.theyDrew();  
b.isHappy();  
// true     (boolean)
```

FairPlay

- Defina em Java uma classe FairPlay, cujos objectos são usados para descobrir o estado de espírito de um adepto de futebol.
- Programe a sua classe no Eclipse
- Teste vários objectos da classe FairPlay e verifique que se comportam tal como esperado.

