

Programando em Java

(Programas com várias classes)

**Material didáctico 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

Objectivo

- Vamos programar em Java um programa que usa vários objectos de diferentes classes.



TwitterHappy

TwitterHappy

- **Objectivo**

- Controlar o estado de felicidade dum utilizador do Twitter num dia.

- **Descrição e Funcionalidades**

- Um utilizador do Twitter pode enviar mensagens através do Twitter, indicando sempre o período do dia em que se encontra. Esse período pode ser: manhã, tarde ou noite.
- Considera-se que uma pessoa está **feliz**, enquanto utilizador do Twitter, se enviar pelo menos **24** mensagens por dia. No entanto, considera-se que atingiu a **felicidade plena** se enviar pelo menos **8** mensagens de **manhã**, **8** à **tarde**, e **8** à **noite**.
- Deve ser sempre possível verificar se o utilizador está feliz e se já atingiu a felicidade plena.
- No início do dia, o utilizador ainda não enviou nenhuma mensagem.
- É sempre possível começar um novo dia.

TwitterHappy

- Interface:

void reset()

Começa um novo dia.

void sendMessage(**int** dayPeriod)

Lança mais uma mensagem no twitter num período do dia indicado.

pre: dayPeriod == ***MORNING*** || dayPeriod == ***AFTERNOON***
 || dayPeriod == ***EVENING***

int getNumberOfMessages(**int** dayPeriod)

Devolve o número de mensagens lançadas no twitter no período indicado.

pre: dayPeriod == ***MORNING*** || dayPeriod == ***AFTERNOON***
 || dayPeriod == ***EVENING***

boolean isHappy()

Indica se o utilizador está feliz.

boolean isFullyHappy()

Indica se o utilizador atingiu a felicidade plena.

TwitterHappy:

cenário de exemplo

- Crie um novo utilizador
- Conte quantos tweets ele tem de manhã (0)
- Teste se o utilizador atingiu a felicidade (false)
- Teste se o utilizador atingiu a felicidade plena (false)
- Faça 7 tweets de manhã e 2 tweets à tarde
- Conte quantos tweets ele tem de manhã (7), à tarde (2) e à noite (0)
- Teste se o utilizador atingiu a felicidade (false)
- Teste se o utilizador atingiu a felicidade plena (false)
- Faça 20 tweets à noite
- Teste se o utilizador atingiu a felicidade (true)
- Teste se o utilizador atingiu a felicidade plena (false)
- Conte quantos tweets ele tem de manhã (7), à tarde (2) e à noite (20)
- Mude de dia
- Faça um tweet à tarde
- Conte quantos tweets ele tem de manhã (0), à tarde (1) e à noite (0)

TwitterHappy:

cenário de exemplo com constantes

```
public static void main(String[] args) {
    TwitterHappy user = new TwitterHappy();
    System.out.println(user.getNumberOfMessages(TwitterHappy.MORNING));
    // 0 (int)
    System.out.println(user.isHappy()); // false (boolean)
    user.isFullyHappy(); // false (boolean)
    user.sendMessage(TwitterHappy.MORNING);
    user.sendMessage(TwitterHappy.MORNING);

    ...

    user.sendMessage(TwitterHappy.MORNING);
    user.sendMessage(TwitterHappy.AFTERNOON);
    user.sendMessage(TwitterHappy.AFTERNOON);
    System.out.println(user.getNumberOfMessages(TwitterHappy.MORNING));
    // 7 (int)
    System.out.println(user.getNumberOfMessages(TwitterHappy.AFTERNOON));
    // 2 (int)
    ...
}
```

Fora da classe, podemos usar as constantes com o seu **NOME** qualificado pelo nome da classe! e.g. `TwitterHappy.AFTERNOON`

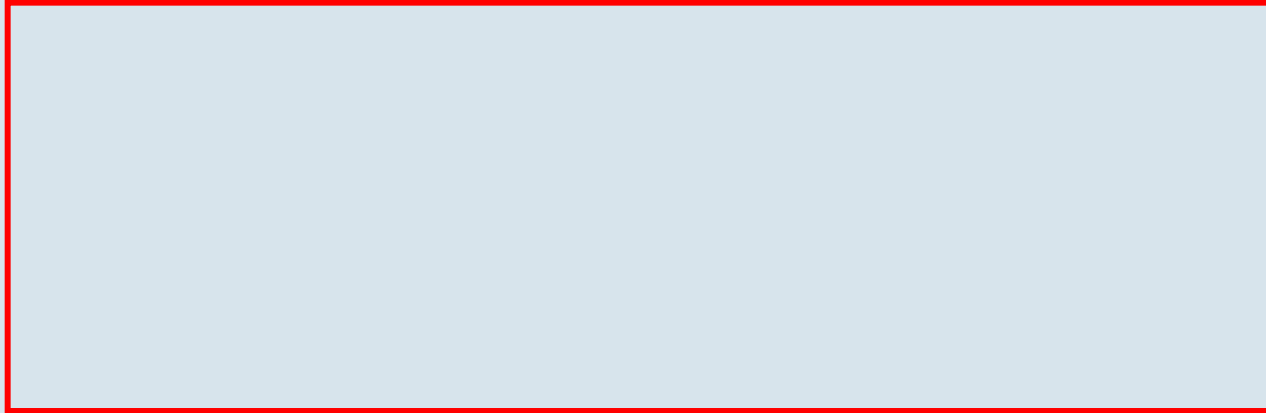
TwitterHappy

- Defina em Java a classe `TwitterHappy`.
- Programe a sua classe no Eclipse.
- Teste um (ou vários) objectos, e verifique se se comportam como esperado.



TwitterHappy

```
public class TwitterHappy {
```



Quais as
constantes?
Quais as
variáveis de
instância?

```
public TwitterHappy() {...}  
public void reset() {...}  
public void sendMessage(int dayPeriod) {...}  
public int getNumberOfMessages(int dayPeriod) {...}  
public boolean isHappy() {...}  
public boolean isFullyHappy() {...}  
}
```

TwitterHappy

Constantes

```
public class TwitterHappy {
```

```
    public static final int MORNING = 1;  
    public static final int AFTERNOON = 2;  
    public static final int EVENING = 3;  
    public static final int HAPPYNESS = 24;  
    public static final int FULL_HAPPYNESS = 8;
```

Constantes
para definir os
períodos do
dia e os
limiares
mínimos de
felicidade.

```
    public TwitterHappy() {...}  
    public void reset() {...}  
    public void sendMessage(int dayPeriod) {...}  
    public int getNumberOfMessages(int dayPeriod) {...}  
    public boolean isHappy() {...}  
    public boolean isFullyHappy() {...}  
}
```

TwitterHappy

Variáveis de instância

```
public class TwitterHappy {  
    public static final int MORNING = 1;  
    public static final int AFTERNOON = 2;  
    public static final int EVENING = 3;  
    ...  
    private morning, afternoon, evening;  
  
    public TwitterHappy() {...}  
    public void reset() {...}  
    public void sendMessage(int dayPeriod) {...}  
    public int getNumberOfMessages(int dayPeriod) {...}  
    public boolean isHappy() {...}  
    public boolean isFullyHappy() {...}  
}
```

Porque não usar um contador?

É necessário contar as mensagens enviadas em cada período do dia

TwitterHappy

Variáveis de instância

```
public class TwitterHappy {  
    public static final int MORNING = 1;  
    public static final int AFTERNOON = 2;  
    public static final int EVENING = 3;  
    ...  
}
```

Classe Counter
já implementada

```
private Counter morning, afternoon, evening;
```

```
public TwitterHappy() {...}  
public void reset() {...}  
public void sendMessage(int dayPeriod) {...}  
public int getNumberOfMessages(int dayPeriod) {...}  
public boolean isHappy() {...}  
public boolean isFullyHappy() {...}
```

É necessário
contar as
mensagens
enviadas em
cada período do
dia

}

Recorde o Contador

- **Interface (classe Counter):**

`void inc()`

soma 1 (um) ao contador

`void dec()`

subtrai 1 (um) ao contador

`void reset()`

coloca o contador a 0 (zero)

`int status()`

consulta o valor corrente do contador



TwitterHappy

```
public class TwitterHappy {  
    public static final int MORNING = 1;  
    public static final int AFTERNOON = 2;  
    public static final int EVENING = 3;  
    public static final int HAPPYNESS = 24;  
    public static final int FULL_HAPPYNESS = 8;  
    private Counter morning, afternoon, evening;
```

```
    public TwitterHappy() {  
        morning = new Counter();  
        afternoon = new Counter();  
        evening = new Counter();  
    }
```

...

```
}
```

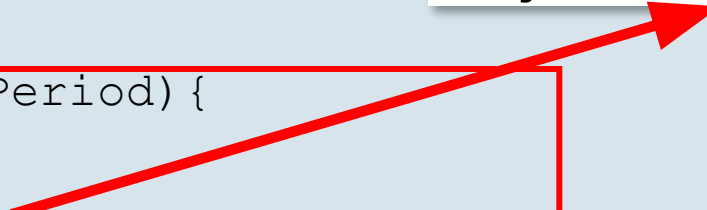


Criação dos
objectos
Counter

TwitterHappy

```
public class TwitterHappy {  
    public static final int MORNING = 1;  
    public static final int AFTERNOON = 2;  
    public static final int EVENING = 3;  
    ...  
    private Counter morning, afternoon, evening;  
    ...  
    public void sendMessage(int dayPeriod){  
        switch (dayPeriod){  
            case MORNING: morning.inc(); break;  
            case AFTERNOON: afternoon.inc(); break;  
            case EVENING: evening.inc();  
        }  
    }  
    ...  
}
```

Invocação do
método “inc” no
objecto Counter



TwitterHappy

Implemente o que falta e crie um programa para testar

```
public class TwitterHappy {  
    public static final int MORNING = 1;  
    public static final int AFTERNOON = 2;  
    public static final int EVENING = 3;  
    public static final int HAPPYNESS = 24;  
    public static final int FULL_HAPPYNESS = 8;  
    public TwitterHappy() {...}  
    public void reset() {...}  
    public void sendMessage(int dayPeriod) {...}  
    public int getNumberOfMessages(int dayPeriod) {...}  
    public boolean isHappy() {...}  
    public boolean isFullyHappy() {...}  
}
```

TwitterHappy

- Definimos em Java a classe `TwitterHappy`.
- Declarámos variáveis de instância do tipo da classe `Counter`.
- Testámos objectos, acedendo às constantes de `TwitterHappy` a partir de outra classe (neste caso, a classe onde temos o programa principal).

