

# UPskill – Java+.NET

Programação Orientada a Objetos – Exceções

### EXERCÍCIO ANIVERSÁRIOS

Pretende-se uma aplicação para guardar, num ficheiro de texto, um conjunto de datas de nascimento de pessoas e respetivos nomes, como é ilustrado na Figura 1. Cada linha do ficheiro deve ter o seguinte formato: Nome; Data de Nascimento. Esta informação deve ser fornecida pelo utilizador.

```
Pedro; Segunda-Feira, 22 de Maio de 2000
Ana; Segunda-Feira, 19 de Setembro de 1966
Rui; Domingo, 8 de Fevereiro de 1998
```

Figura 1 – Exemplo de um ficheiro de texto

A aplicação deve ser implementada para a Consola e deve ter o comportamento exemplificado na Figura 2. Inicialmente tem de poder ler o nome completo do ficheiro de texto. No caso de não ser possível criar o ficheiro, o programa tem de permitir ao utilizador indicar outro nome completo do ficheiro. Depois de criado o ficheiro, o programa deve possibilitar ao utilizador a introdução de uma sequência de nomes de pessoas e respetivas datas de nascimento. Estas datas devem ser fornecidas no formato ano/mês/dia. No caso de uma data ser inválida, ou ter um formato inválido, deve ser apresentada uma mensagem de erro apropriada, indicando o tipo de erro ocorrido (formato ou mês ou dia inválido) e, de seguida, tem de permitir ao utilizador a correção do erro. A leitura de nomes e datas de nascimento deve terminar quando, após o pedido de um novo nome, o utilizador responder apenas com o ENTER.

```
Nome do ficheiro: c:\aniversarios.txt

Impossível criar ficheiro!!
c:\aniversarios.txt (Acesso negado)

Digite novamente o nome do ficheiro: aniversarios.txt
Ficheiro criado.

Nome ("ENTER para terminar"): Pedro
Data de aniversário (ano/mes/dia): 20000522
Formato é inválido!!
Digite novamente a data de aniversário (ano/mes/dia): 2000/05/22

Nome ("ENTER para terminar"): Ana
Data de aniversário (ano/mes/dia): 1996/19/09
Mês 19 é inválido!!
Digite novamente a data de aniversário (ano/mes/dia): 1966/09/19

Nome ("ENTER para terminar"): Rui
Data de aniversário (ano/mes/dia): 1998/02/08

Nome ("ENTER para terminar"):

Aplicação terminada.
```

Figura 2 – Exemplo de leitura de dados da aplicação

Implementar a aplicação, representando as datas de aniversário através da classe *Data* fornecida. Este desenvolvimento deve ser realizado da seguinte forma:

1. Criar duas classes de exceção próprias e *unchecked*, designadas *MesInvalidoException* e *DiaInvalidoException*, para validar os parâmetros *mes* e *dia* da classe *Data*, respetivamente.
2. Atualizar a classe *Data* de modo a validar todos os parâmetros *mes* e *dia*. Esta validação deve ser baseada no lançamento de exceções das classes *MesInvalidoException* e *DiaInvalidoException*.
3. Implementar testes unitários do método *setData* atualizado, pertencente à classe *Data*.
4. Criar uma classe não instanciável que disponibilize um método para ler da Consola uma data com o seguinte formato: ano/mês/dia. Esta leitura deve ser repetida enquanto o utilizador fornecer uma data inválida ou com formato inválido, conforme é exemplificado na Figura 2.

A mensagem de erro fornecida ao utilizador deve indicar o tipo de erro:

- (1) formato inválido,
- (2) mês inválido ou
- (3) dia inválido.

No final da execução, este método deve retornar uma instância de *Data*, representando a data válida lida.

5. Criar a classe *Main* da aplicação pretendida.
6. Completar a implementação da aplicação de modo a satisfazer o comportamento apresentado na Figura 2.

Para a criação do ficheiro considere o seguinte exemplo de código:

```
import java.util.Formatter;

...

private Formatter output;

...

output = new Formatter(nomeFicheiro);

output.format("%s; %s %n", nome, dataNasc);
output.close();
```