Guião

# Apresentação:

Olá e bem-vindos ao último curso do conjunto de cursos de *java*.

O meu nome é João Barreiras, e vou vos acompanhar ao longo deste curso avançado de *Java*. Como já devem saber, este curso tem como pré-requisito que tenham conhecimentos intermédios de *java*, e por isso aconselho-vos a terem frequentado o curso de j*ava* intermédio.

Então, e o que vamos aprender aqui?

Na primeira semana vamos aprender a criar programas que utilizem a internet, vamos falar principalmente sobre como criar programas java que acedam à internet e sobre Regular Expressions. Na segunda semana vamos falar sobre *webservices* e dos dois formatos populares de estruturação de dados, *JSON* e *XML*. O tema da 3ª semana será a programação paralela. Na semana final deste curso temos um projeto final onde vamos aplicar tudo o que aprendemos.

Como os cursos anteriores o nosso *IDE* de seleção é o *Eclipse Neon* e vamos utilizar o *Java 8*, contudo são livres de usar outras alternativas equiparáveis temos na seção de links úteis alguns tutoriais de como instalar o *Eclipse* (em *Windows*, *MAC OS X* e *Ubuntu*) e garantir que o Java está funcional, também nos links úteis têm o *link* para a nossa página do *GitHub*, onde poderão encontrar todos os exercícios e exemplos aqui apresentados, bem como outros recursos que consideramos relevantes.

Finalmente queria dar-vos os parabéns por terem chegado até este ponto, e desejo-vos boa sorte, vamos começar então?

# Semana 1

## Ligação à internet

### PowerPoint

Slide1

Bem-vindos à primeira aula do nosso curso avançado de java. Esta aula introdutória está divida em três fases. Na primeira iremos abordar o tema da ligação à internet, começando com uma revisão sobre alguns conceitos, sendo eles o TCP, o HTTP e o HTML. De seguida será apresentado um excerto de código java que permite a ligação à internet. Por fim, será proposto ao participante a realização de um exercício para aperfeiçoamento dos conhecimentos.

Slide2

Começaremos então pelo modelo TCP/IP, responsável por todas as ligações entre entidades na Internet. Este é um modelo complexo, composto por várias camadas que comunicam entre si para garantir a ligação e transferência de dados entre duas máquinas ligadas à Internet. Iremos, no entanto, abstrair-mo-nos da sua complexidade e vamos concentrar-nos na ligação peer-to-peer. Para isto vamos recorrer a uma classe java, a classe URL. Esta classe permite a criação de um objecto URL a partir de uma string, e oferece também vários métodos para a sua manipulação e utilização. Para quem esteja interessado em mais informações, disponibilizamos aqui a documentação sobre esta classe, bem como um link para mais informação sobre o funcionamento da Internet. (e outro para a origem da imagem relative ao modelo TCP/IP)

Slide3

As nossas comunicações através da Web utilizarão o HTTP, um protocolo que faz parte da camada aplicacional do modelo TCP/IP. Um simples protocolo baseado em pedidos pelo cliente, normalmente um browser web, a conteúdos alojados num servidor web, o qual responde aos pedidos enviando o conteúdo requisitado ao cliente. Há também, como se vê na imagem, a possibilidade de o servidor enviar cookies para o cliente, os quais são depois enviados para o servidor juntamente com futuros pedidos, permitindo aprofundar a comunicação entre servidor e cliente. No java existem três classes principais que utilizam este protocolo para a ligação à Internet: a classe URL/URL connection, na qual nos iremos focar nesta aula, a classe HttpComponents, pertencente à biblioteca Apache, e a classe Socket.

Slide4

Além do HTTP, responsável pela comunicação de dados, é também do nosso interesse conhecer o HTML, a linguagem utilizada para formatar páginas web. Esta é uma linguagem que permite a representação estruturada de conteúdos texto e multimedia, focada no uso de tags. Uma tag consiste num texto padrão, delimitado pelos sinais de menor que e menor que, e que é iniciada, aparecendo depois desta o conteúdo ao qual queremos que esta se aplique, e finalmente a tag é fechada escrevendo novamente a tag com uma barra no início do seu texto, marcando o fim do conteúdo afectado pela tag. Este conteúdo pode ser texto, links para imagens ou para outros websites, etc. Ao trabalhar com a internet é essencial perceber esta linguagem e conhecer as principais tags, e as nossas aulas não serão excepção.

Slide5

Vamos agora passar ao código java. Este excerto de código, encapsulado num try/catch para evitar erros em tempo de execução que podem ocorrer ao trabalhar num ambiente web, começa num primeiro momento por criar um novo objeto da classe URL, o qual será inicializado com a hiperligação para a página do github referente a este exercício. Uma String onde armazenaremos a resposta do servidor web também é criada neste momento. De seguida iniciamos uma ligação utilizando o URL anterior, que irá resultar numa resposta do servidor web contendo a página web pedida. Os dados devolvidos pelo servidor são então recebidos por uma InputStream, e armazenados num buffer. Agora é apenas necessário guardar o texto, que neste momento está armazenado no buffer, na String criada para receber a informação da página. Um ciclo while permite receber todo o texto do buffer, que é por fim escrito na consola.

Slide6

O exercício anterior permite-nos receber, armazenar e apresentar o conteúdo de uma página, mas não é uma solução ideal. Como se pode ver no código html no topo, que corresponde ao resultado do código anterior, é difícil ler a informação contida na página, já que o HTML não foi criado para ser utilizado para armazenar dados, mas para os estruturar para apresentação em páginas web. Assim, iremos na próxima semana abordar o XML e o JSON, duas alternativas interessantes para este tipo de transferência \

Slide7

Para terminar, temos então uma proposta de exercício, para quem quiser praticar o que aprendemos neste vídeo. Este exercício simples não é muito diferente do nosso exemplo, mas desta vez o objetivo é especificamente obter as hiperligações das imagens presentes na página indicada. Se tiver dificuldades e não conseguir resolver o exercício não se preocupe, que o nosso próximo vídeo começará com a resolução deste exercício. Chegámos assim ao fim da nossa primeira aula, obrigado pela vossa presença e esperamos ver-vos no próximo vídeo!

### Exercício

#### Enunciado

Escrever um programa java que aceda ao url: <https://java-mooc.github.io/Advanced-Java/ex2.html> e faça o download de todas as imagens.

## Regular Expressions

### PowerPoint

Slide 1

Olá e bem-vindos à nossa segunda aula. Hoje iremos abordar o tema de Regular Expressions, que, como veremos de seguida, são uma ferramenta de grande utilidade para o programador, simplificando certos desafios que este possa encontrar. Iremos então introduzir regular expressions, aprofundar um pouco sobre os seus benefícios e utilidade, e deixar-vos com algumas ferramentas para poderem praticar. Antes disso, como mencionado na aula anterior, vamos começar por mostrar uma resolução para o problema apresentado na aula anterior, que pode ser melhorada com recurso a regular expressions.

Slide 2

Vamos então revisitar o exercício da última aula. No final do nosso último vídeo, foi proposta a criação de um programa, a partir do código de exemplo explorado na aula, que obtivesse as hiperligações das imagens de uma página indicada por nós. O código que aqui temos é, portanto, as alterações necessárias para que tenhamos uma String que guarde as hiperligações das imagens em vez de todo o conteúdo. No entanto, esta solução é confusa e feia, não se percebendo facilmente o propósito do código. É, por isso, possível torna-la mais elegante com o uso de regular expressions, como iremos ver em seguida.

Slide 3

Então, o que são regular expressions? São um mecanismo, criado por Stephen Cole Kleene, que permite procurar e substituir expressões numa String, que podem ser caracteres, palavras ou mesmo padrões de caracteres. Estas expressões são escritas numa linguagem formal, que pode ser interpretada por um processador de regular expressions. Também é possível validar formatos de texto com recurso a esta ferramenta. Com um realce na sintaxe e uma grande flexibilidade que não é possível encontrar nos métodos tradicionais de procura, que simplesmente comparam um conjunto fixo de caracteres com uma String, as regular expressions são uma grande ferramenta de ajuda à filtragem de informação.

Slide 4

Estes são os símbolos e a sintaxe da linguagem de regular expressions, que podem ser interligados entre eles para dar origem a expressões úteis, flexíveis e abrangentes. Os primeiros símbolos indicam o início e o fim de uma linha de texto, o ponto final indica a presença de qualquer caracter, \s indica espaço em branco, enquanto que \S indica tudo menos espaço em branco. \* indica zero ou mais ocorrências da expressão anterior, enquanto que + indica uma ou mais ocorrências. Estes símbolos são “greedy” por defeito, ou seja, caso uma expressão seja válida para uma expressão e também para outra expressão contida nesta, uma expressão “greedy” retornará sempre o resultado mais extenso possível. Para contrariar isto, adiciona-se um “?” a seguir ao \* ou . , consoante o caso. Um conjunto de caracteres dentro de parêntesis retos indica os caracteres que se pretende encontrar naquela posição, enquanto que um “^” antes dos caracteres indica que não se pretende esses caracteres. Um traço entre dois caracteres indica o conjunto de caracteres que existe entre eles, e por fim os parenteses curvos indicam onde começar e onde acabar a extrair uma String do texto original.

Slide 5

Acabamos assim a teoria das regular expressions, como sabemos que este sistema pode ser inicialmente confuso de perceber deixamos aqui um site onde poderão testar regular expressions. Neste site basta colocar uma String que será testada e a regular expression que queremos testar, e recebemos imediatamente todas as matches encontradas na string, bem como uma explicação de todas as partes da expressão escrita por nós, indicando o seu significado. Este site foi testado por nós e acreditamos que será uma ferramenta muito benéfica para aqueles que estiverem interessados em praticar e testar regular expressions.

Slide 6

Por fim, como estamos num curso de Java, ficam aqui exemplos de como utilizar esta nova ferramenta. Temos os métodos mais básicos, que fazem parte da classe String do Java, e que permitem encontrar e dividir uma String segundo uma regular expression, e temos ainda um método que substitui a primeira ocorrência de uma regular expression por uma outra String, e outro que faz essa mesma substituição, mas para todas as ocorrências da regular expression. Para aqueles que estejam interessados em mais utilizações de regular expressions, existem ainda as Classes Matcher e Pattern, que podem ser muito interessantes para quem queira utilizar regular expressions frequentemente ou para fins específicos. Chegamos assim ao fim da nossa segunda aula, obrigado pela vossa presença e vemo-nos na próxima aula!

### Exercício

## Quizz Final

# Semana 2

JSON/XML

### PowerPoint

## Webservices

### PowerPoint

### Exercício

## Quizz Final

# Semana 3

## Paralelismo de tarefas

### PowerPoint

### Exercício

## Threads

### PowerPoint

### Exercício

## Sincronização de dados e pipelining

### PowerPoint

### Exercício

## Quizz Final

# Semana 4 (Projeto Final)

## Enunciado

## Objetivos