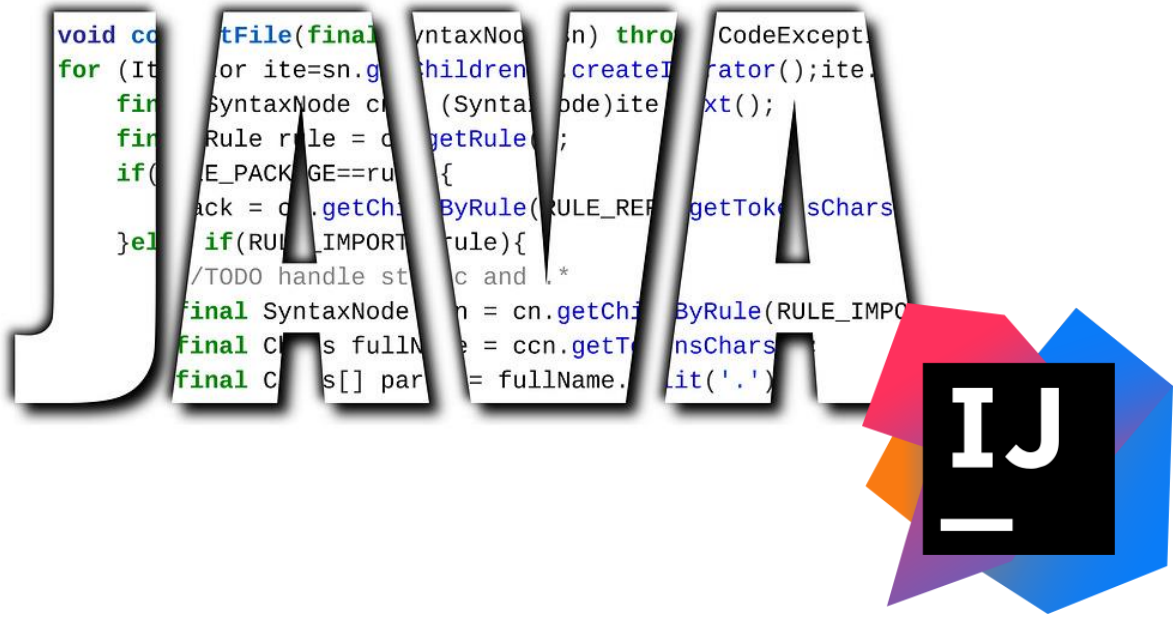
1 2 9 0

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D COIMBRA



Dados pessoais

Nome do Aluno: Gonçalo Tavares Antão Folhas Ferreira

Número do Aluno: 2019214765

Dados do Trabalho

Trabalho: Projeto Final [POO]

Software utilizado: IntelliJ

Introdução

No âmbito a disciplina de Programação Orientada a Objetos (POO) e com base nos conhecimentos adquiridos ao longo do semestre, foi proposta a realização de um projeto individual.

Neste projeto, toda a matéria foi incluída nas perguntas, tendo principal foco na manipulação de classes através de herança, polimorfismo, implementação de interfaces e ainda o uso de ficheiros de texto e ficheiros de objeto para permitir um acesso mais fácil aos dados em que vamos trabalhar.

Estrutura Geral

O projeto envolve um centro de investigação com diversos grupos de investigação, compostos por 2 tipos de investigadores, membros efetivos e estudantes, cada um com caraterísticas em comum e outras específicas. Estes podem ter publicações, sendo que estas têm variantes, umas sendo livros, artigos de conferência, etc. Para além disso, cada publicação têm também atributos específicos e podem ser organizadas com base na sua data de publicação, no seu fator de impacto e pelo seu tipo de publicação.

Para além disso, são também necessários métodos que permitam verificar a existência de ficheiros objeto com as informações que precisamos. Se estes existem, retiramos a informação deles e avançamos para a etapa seguinte. Se não existem, a informação é retirada de ficheiros texto e no final do programa, a informação com as alterações desejadas será armazenada num conjunto de ficheiros objeto.

A informação é armazenada em ArrayLists no programa de forma a ser facilmente acessível. Após a criação destas estruturas, avançamos para as perguntas.

Exercício 1

Neste exercício é nos pedido para apresentar todos os membros, o número de membros de cada categoria (efetivos e estudantes), o total de publicações dos últimos 5 anos e o número de publicações de cada tipo.

Para mostrar todos os investigadores, basta fazer um ciclo onde se mostre no ecrã todos os elementos das listas de membros efetivos e estudantes. Para calcular o número de elementos de cada lista basta fazer um contador que incremente a cada iteração da lista.

Relativamente às publicações, teremos de fazer uma filtragem com base no atributo destas que contém a data, fazendo uma comparação com o ano atual (definido como 2020). Se a diferença do ano atual e do ano de publicação for não maior do que 5, esta publicação será mostrada.

Por fim, para saber o número de cada tipo de publicação, teremos de percorrer o ArrayList das publicações e fazer um getType() da publicação em questão. Após isto, incrementamos 1 ao valor correspondente ao número de publicações desse tipo.

Exercício 2

Neste exercício, é nos pedido para organizar as publicações dos últimos 5 anos de um grupo de investigação, mas através de 3 tipos de ordenação diferentes. Ordenar por data, fator de impacto e tipo de publicação.

Para qualquer uma destas ordenações, recorri ao uso de interfaces e a Collections, de forma a que em cada interface existe um **compare** com o tipo de ordenação exigida pelo problema e depois basta chamar o Collections.sort(arg1, arg2), em que o arg1 representa a lista de todas as publicações e o agr2 uma nova interface do tipo pedido.

Após a ordenação, criei uma ArrayList vazia e percorri o ciclo de publicações do grupo de investigação desejado, assim como a lista de membros e comparo os autores da publicação a ser lida com o nome do membro a ser lido. Se ambos corresponderem, adiciono a publicação à ArrayList previamente vazia e continuo o processo. No fim dos ciclos, basta apenas percorrer a nova ArrayList e mostrar no ecrã as publicações que lhe foram adicionadas, sendo que estas agora se encontram organizadas pela ordem desejada.

Relativamente aos tipos de organização, defini que por fator de impacto seria apropriado mostrar na seguinte ordem (A->B->C), por data seriam agrupados do mais antigo (não mais do que 5 anos) até ao mais recente e por tipo foram organizados alfabeticamente.

Exercício 3

Neste exercício é pedido para listar os membros de um determinado grupo de investigação agrupados pela sua categoria. No meu programa em específico, eu desde o início que decidi criar ArrayLists separados para cada categoria, uma vez que facilitava a implementação de soluções para problemas como este, sendo que basta mostrar primeiro uma das ArrayLists e depois a outra para os investigadores se encontrarem organizados como pedido. Contudo, se eu tivesse de os guardar numa ArrayList conjunta, bastava utilizar polimorfismo para mostrar primeiro os membros que fossem de um tipo e posteriormente os membros do outro tipo. Não sendo especificado no enunciado que estes teriam de ser armazenados na mesma lista, decidi optar por 2 listas e a sua organização já está então feita.

Exercício 4

Para listar as publicações de um dado investigador, com determinada organização, tive de mais uma vez utilizar as interfaces desenvolvidas no ponto 2 para organizar as publicações. Posteriormente basta percorrer um ciclo com todas as publicações e verificar quais pertencer ao investigador pedido e apresentá-las no ecrã. Neste ponto, como já mencionei, o objetivo é organizar um grupo de informação, neste caso publicações, com determinada ordenação específica, sendo possível reaproveitar as interfaces utilizadas anteriormente, sendo que a única diferença será nos ciclos a percorrer posteriormente à ordenação da informação.

Exercício 5

Por último, é pedido para apresentar para todos os grupos de investigação o seu total de membros, o número de membros de cada categoria (estudante ou efetivo), o total de publicações dos últimos 5 anos e o número de publicações dos últimos 5 anos, agrupadas por ano, tipo de publicação e fator de impacto.

Relativamente ao total de membros e ao seu número, fiz um ciclo que percorre os grupos de investigação um a um, e depois ciclos que percorrem as minhas ArrayLists de estudantes e de membros efetivos. Fazendo uma verificação, comparando o grupo de investigação do membro a ser analisado com o grupo de investigação da iteração do ciclo, podemos concluir que se estes forem iguais, o membro pertence ao grupo, sendo por isso apresentado no ecrã (ou adicionada a uma lista apresentada no final da iteração do ciclo) e com a ajuda de duas variáveis counter (uma para o número de estudantes e outra para o número de membros efetivos), inicializada a 0, incrementada sempre que isto acontece, podemos assim apresentar os membros do grupo, assim como o número de membros efetivos e estudantes do grupo.

Quando se muda de grupo de investigação, os counters voltam a 0 e o processo é repetido.

Para apresentar as publicações dos últimos 5 anos de cada grupo de investigação, tive de fazer um ciclo para ir pegar em cada publicação individualmente, verificar se o/os autor/autores pertencem ao grupo de investigação e verificar se a publicação foi de facto publicada à 5 anos ou menos. Se isto tudo se verificar, a publicação é adicionada a uma ArrayList que vai conter todas as publicações que sejam aceites nestes modos. Assim, será possível no fim do ciclo apresentar cada publicação individualmente para cada grupo de investigação específico.

No último ponto, (5d) eu senti algumas dificuldades a perceber o objetivo do problema, uma vez que nos pede o número de publicações dos últimos 5 anos, organizadas pelos diferentes modos utilizados previamente. Contudo o número de publicações será sempre igual, quer estas sejam ordenadas por ano, tipo ou fator de impacto. Tendo por isso apenas apresentado o número das publicações de cada grupo de investigação, sendo a ordenação irrelevante para este problema em específico.

NOTA: Ponto 5D: Como, a meu entender, era irrelevante, não utilizei as interfaces de sort previamente utilizadas, mas se tivesse de as aplicar, usava-as como usei anteriormente, sendo que cada uma tem um tipo de organização específico e o seu uso seria apenas uma chamada como “Collections.sort(pubs, new InterfaceSort())” para as ordenar.

Reflexões Finais:

O balanço geral que retiro do projeto é positivo, não encontrei muitas dificuldades no desenvolvimento do mesmo, sendo que as que encontrei foram facilmente detetadas posteriormente após alguma reflexão e análise do que tinha feito. Se tivesse de refazer este projeto, provavelmente implementava apenas 1 ArrayList de investigadores em vez de ter 1 para cada tipo de investigador, mas de resto sinto me satisfeito com o trabalho no geral. Serviu para expandir alguns conhecimentos de matéria que ainda não tinha trabalhado muito.