Atividade T2 - ECM251

"MAsK_S0c13ty".v02

Prof. Murilo Zanini de Carvalho

Junho|2021

_Descrição:

O ano é algum, que muito provavelmente pode ser calculado com ano floor(atual +(tempo necessário para ninguém lembrar de uma série* Pi)/exp(0)), onde a sociedade moderna sofreu um grande ataque de um grupo de hackers chamados "MAsK_S0c13ty". Este grupo trabalha em duas frentes:

- como atividades conhecidas pela comunidade, eles mantêm repositórios e comunidades open-source que ensinam velinhos e crianças a programar.
- Agora, o que ninguém sabe é que esse grupo também arquiteta grandes revoluções criando softwares que mineram "S3rJA0_C01nS", a cripto moeda mais rentável daquela sociedade.

Mesmo com seu grupo conseguindo atuar em diversas áreas, eles não estão conseguindo gerenciar seus membros e as atividades que estes devem desenvolver. O maior problema deles, reside no fato que todo sistema de controle que eles elaboram, não é escalável e, logo, toda vez que eles crescem no número de participantes do grupo, quase todo o sistema precisa ser reescrito quase que do zero.

Você foi apresentado para um dos líderes regionais da "MAsK_S0c13ty", o "B1az3" (sim, todos eles misturam números nos nomes para ficar com uma aparência mais c00l no mundo Hacker - Robin, Jovens Titãs). Ele ouviu falar que você está aprendendo orientação objetos na sua faculdade e que você é muito bom em utilizar esses pilares para propor sistemas de gerenciamento.

Ele te contou que tudo que é realizado na "MAsK_S0c13ty" depende do tipo do membro que está responsável pela ação. Os membros que atuam apenas com seus celulares, são conhecidos como "Mobile Members". Os que atuam com estações de trabalho pesadas,

equipadas com 128 GB de memória RAM, suficiente para 4 abas abertas simultâneas de um dado navegador de internet e uma IDE aberta, são chamados de “Heavy Lifters”. Os que trabalham com seus notebooks, são os “Script Guys”. Por fim, os membros gestores são os “Big Brothers”, pois são responsáveis por monitorar se os demais membros estão trabalhando apenas nas suas tarefas e não invadindo as funções dos colegas.

O sistema proposto deve permitir que os Big Brothers possam cadastrar novos membros, atribuindo um papel específico para eles. Cada membro deve ser capaz de realizar uma série de ações características:

- “Mobile Members”: gerenciam postagens em redes sociais. Seus postos variam de acordo com o momento do dia. Durante as atividades regulares, todos eles devem assinar as postagens com “Happy Coding!”. Durante as atividades “extras” da “MAsK_S0c13ty”, eles devem assinar os posts como “Happy_C0d1ng. Maskers”.
- “Heavy Lifters”: São os responsáveis por manter os repositórios. Cada vez que mandam uma solicitação de push, assinam elas com “Podem contar conosco!”, durante atividade regular ou como “N00b_qu3_n_Se_r3pita.bat”, nas horas “extras”.
- “Script Guys”: fazem os primeiros contatos com novos clientes. As suas mensagens são “Prazer em ajudar novos amigos de código!”, quando estão vestindo roupas sociais leves e tomando um café da “*” Coffe Shop. Ou são assinadas como “QU3Ro_S3us_r3curs0s.py”, quando em suas cavernas pessoais, com seus achocolatados.
- “Big Brothers”: estão assinam suas mensagens com “Sempre ajudando as pessoas membros ou não S2!”, quando vestidos de pessoas carinhosas. Já quando estão levando as interações de sua companhia, são tão terríveis que apenas assinam “...”, para dar ainda mais suspense no coração de quem recebeu tal mensagem.

Todos os membros devem ser capazes de implementar a interface PostarMensagem (sim ele foi bem enfático quando disse a palavra INTERFACE). Os membros também devem ser capazes de perguntar

para o sistema se ele está em horário normal de trabalho ou se estão em horário de atividades “extras”. Os tipos de membros, bem como o horário do sistema devem ser implementados em duas ENUMERAÇÕES diferentes, cada uma delas contendo as características que forem relevantes para o sistema.

Seu sistema principal deve ser capaz de registrar os membros, informar para eles quando eles perguntarem qual o horário eles estão atuando e postar uma mensagem para todos os seus integrantes. Por falar nisso, a quantidade de integrantes do grupo pode variar, portanto ele precisa ser flexível para armazenar todos eles. Os integrantes devem ser cadastrados por seus nomes de usuário, e-mail e função. O sistema deve ser capaz de mudar o horário de trabalho quando for solicitado.

Em um mundo ideal, todas as interações sempre dão certo. Aqui no mundo da “MAsK_S0c13ty”, elas falham algumas vezes. Quando isso acontece, é preciso retirar algum membro do sistema.

Quando for preciso, é preciso exibir um relatório contendo as informações de cada um dos membros. Cada membro deve ser capaz de implementar a interface Apresentacao (de novo ele foi muito enfático na palavra INTERFACE), que deve permitir que ele se apresente quando o relatório for chamado.

_Implementação:

Para testar seu sistema, você deve:

- Apresentar uma tela de boas vindas, apresentar um menu de opções para o usuário e exibir qual o horário de trabalho;
- Cadastrar um usuário para cada cargo;
- Postar uma mensagem para cada funcionário;
- Trocar o horário de trabalho de regular para “extra”;
- Postar novamente um mensagem para cada funcionário;
- Cadastrar mais um usuário de cada do sistema;
- Voltar o horário de trabalho para regular;
- Excluir o primeiro “Heavy Lifter” e o segundo “Script Guys”;

- Postar uma mensagem com todos cadastrados;
- Encerrar o sistema.

Observação importante: o arquivo com os dados do usuário deve ser super secreto, logo ele deve estar nomeado como: “arquivo_super_Secreto_nao_abrir.csv”. Dentro dele, cada informação de usuário (sua categoria, seu nome e seu id) deve ser gravados separados por ‘;’, um registro por linha no arquivo.

A sua classe main.java deve apenas iniciar o seu sistema e TODA interação com o usuário deve acontecer apenas na sua classe principal do sistema.

Utilize os pacotes “models”, “interfaces” e “enums”, para as classes de modelo, as interfaces e para as enumerações, respectivamente. A utilização de outros pacotes fica a seu critério.

Classes que não serão instanciadas, devem estar marcadas como abstrata (sim estou falando da classe mãe Membro).

_Regras:

- A atividade pode ser realizada em duplas (apenas para deixar claro, uma dupla tem no máximo 2 pessoas).
- A cada nova classe que for criada, comitar ela no gerenciador de código que estiver sendo utilizado.
- Os nomes e os R.A.s da dupla devem estar na classe main.java do sistema.
- Utilizar a documentação do tipo Javadoc para todos os elementos que forem descritos no sistema.
- Notificar o professor que a atividade foi finalizada, enviando para ele uma mensagem na ferramenta de comunicação utilizada na disciplina (Ms Teams). Apenas um dos membros da dupla deve fazer isso. Aguardar o professor confirmar que já clonou o repositório.

_Boa Atividade!

_Critérios de Avaliação Utilizados:

Critério	Peso
Pilar de abstração da Orientação a Objetos.	2
Pilar do encapsulamento da Orientação a Objetos.	2
Pilar da herança da Orientação a Objetos.	2
Pilar do polimorfismo da Orientação a Objetos.	3
Utilização correta das enumerações (enums).	1
Utilização correta das interfaces.	2
Manipulação de entrada e saída de dados apenas na classe principal da aplicação.	1
Utilizar a clase main.java apenas para inicializar o sistema.	1
Utilização correta das estruturas de dados.	3
Documentação do sistema utilizando Javadocs.	3
Cadastro de novos membros.	1
Exclusão de membros cadastrados.	1
Comunicação entre classe do sistema e demais elementos para compartilhar informação da jornada de trabalho.	2
Utilização de boas práticas de desenvolvimento (escolha dos nomes das classes, atributos e métodos).	1
Uso correto dos pacotes.	1

_Método de Entrega: Link para o repositório no Github com o projeto.

ATENÇÃO: Todos os personagens, lugares, incidentes, organizações e religiões retratados são fictícios.