

1. A Computer KXPO, startup de grande sucesso e recém-criada por alunos do SENAC, foi convidada para desenvolver protótipos de vários produtos relacionados à locomoção de pessoas e cargas, enfim, e você, por ser um excelente estudante do Projeto Jovem Programador, foi indicado a participar. A sua parte será o desenvolvimento de uma aplicação desktop, com a biblioteca Swing, em Java, de um **Elevador** para um edifício de apenas cinco andares. Após reunião com o cliente e demais envolvidos, definiu-se que o desenvolvimento será incremental, como segue:

PASSO 1

Nesta primeira etapa, o time de desenvolvimento pediu uma Janela do tipo JFrame com os botões Abrir e Fechar, para simular a abertura e fechamento das portas do elevador, e um botão para cada andar.

Ainda na primeira etapa será preciso "setar" as características da janela: nome, tamanho, posição, defaultClose, layout, panel etc. O método main instanciará esta janela que deve ser aprovada pelo cliente. Só então será autorizada a próxima etapa incremental.

Salve a primeira codificação do seu APP e coloque abaixo o link do Github *

2. PASSO 2

Considerando que o cliente aprovou sua interface gráfica, você vai desenvolver melhor a sua lógica. Uma variável "andar" do tipo int ou Integer deve armazenar o andar que o elevador está. Através da classe ActionListener, você adicionará um evento aos botões para simular a chamada do elevador quando pressionados.

O método chamarElevador(int andar) por enquanto apenas imprime uma mensagem indicando o andar para o qual o elevador foi chamado e em que andar ele está parado. Mostre estas funcionalidades ao cliente e, mediante aprovação, siga para o próximo passo.

O código desta segunda codificação deve aparecer em uma pasta separada, imediatamente abaixo do primeiro código que funcionou de modo estável e foi aprovado pelo cliente. Coloque o link do github com a continuidade. *

3. PASSO 3

Implementar o método moverElevador() com a classe Timer e TimerTask para sobrescrever o método run() e um atraso de alguns segundos entre um andar e outro. Assim, se uma pessoa está no quinto andar ela consegue ver que o elevador passou pelo primeiro, segundo, terceiro e quarto andar até chegar ao destino. Aprovada essa etapa, seguir adiante.

Da mesma forma que na etapa anterior, o código deve aparecer imediatamente abaixo do segundo código estável e aprovado pelo cliente. Deste modo, esta é a sua terceira pasta do repositório do projeto. Coloque o link do github abaixo: *

4. PASSO 4

Atualizar o método chamarElevador(andar) para setar a cor dos botões com o setBackground(Color.nome-da-cor) para verde caso o elevador esteja neste andar ou amarelo para uma solicitação pendente para esse andar ou cinza para nenhum dos dois casos.

Um ActionListener para os botões Abrir e Fechar simulará a abertura e o fechamento das portas do elevador, ou seja, o método abrirPorta(), vinculado ao botão Abrir, muda a cor do botão Abrir para vermelho para indicar que as portas estão abertas.

O Timer que você criou para simular um atraso e dar a ideia de que o elevador está andando, agora também permite o ajuste das cores. Se a cor é cinza indica que o botão está disponível para clicar. Chame o cliente quando estiver pronto e finalize o projeto.

FINALIZANDO

Meus parabéns! Ao chegar nesta etapa você concluiu o projeto de acordo com o interesse do cliente e atendeu a todos os requisitos segundo as etapas definidas previamente de modo incremental. Publique abaixo o link do seu github com a solução final. *

5. Seu nome : *

Este conteúdo não é criado nem endossado pela Microsoft. Os dados que você enviar serão enviados ao proprietário do formulário.



Microsoft Forms