



Erros detectados
depois da entrega



Load de um ficheiro de estado

Reparamos que depois de fazer load de um estado do programa quando criamos novas instâncias das classes estas substituíam as já existentes. Isto devia-se ao facto de que durante o save do estado as variáveis static que ajudavam a definir os id's das nossas instâncias de classe não eram guardados e, por isso, quando reniciávamos o programa estas variáveis static davam reset, fazendo assim com que as nossas novas instâncias tivessem o mesmo id que instâncias mais antigas, substituindo estas.



Resolução

Para resolver este problema no momento do load calculamos o número de utilizadores, artigos, encomendas e transportadoras que existiam e mudamos os valores das variáveis static de acordo.

```
Utilizador.setNUtilizadores(model.getUtilizadores().size());  
Artigo.setNArtigos(model.getArtigos().size());  
Encomenda.setNEncomendas(model.getEncomendas().size());  
Transportadora.setnTransportadoras(model.getTransportadoras().size());
```

Estas foram as mudanças feitas para a resolução deste problema juntos com a criação dos métodos setNUtilizadores(), setNArtigos(), setNEncomendas().



Leitura de input de valores

Quando testamos melhor o nosso menu encontramos problemas em que o menu não nos deixava introduzir valores dando skip aos scans que devia fazer:

```
Introduza o seu email:  
Introduza o seu nome:  
█
```

Reparamos que este erro acontecia sempre depois de lermos um int ou um double, isto deve se ao facto que os métodos `nextInt()` e `nextDouble()` lerem os valores de forma automática sem necessidade de dar enter. Por isso, mal o int ou double ser introduzido o próximo `scanner.nextLine()` já estava a ler logo depois de introduzir os valores anteriores e, por isso quando davamos enter para, supostamente, confirmar os valores int's / double's o nosso scanner só lia `'\n'`.



Resolução

Para resolver este problema apenas necessitamos fazer as seguintes alterações:

```
sc.nextInt();
```

por

```
Integer.parseInt(sc.nextLine());
```

```
sc.nextDouble();
```

por

```
Double.parseDouble(sc.nextLine());
```



Possíveis alterações futuras

Após uma visão geral do trabalho, sentimos dificuldade a navegar pela informação fornecida pelo programa, por esta não estar bem organizada .

```
{ id='4', email='leo@sapo.pt', nome='Leandro Rio', morada='Barcelos', NIF='467209008', stock_produtos='[5, 6]', artigos_vendidos='{}', receitaVendas='0.0', faturas='{{ id='2', data='2023-05-08', comprador='4', artigos='{3=Malas [dimensao=4.0, material=pele de zebra, anoDaColecao=1990, nDonos=0, estadoUtilizacao='100.0, descricao='Linda, marca='LV, precoBase='400.0, idTransportadora='1], 12=Malas [dimensao=2.0, material=pele de anaconda, anoDaColecao=1990, nDonos=1, estadoUtilizacao='99.0, descricao='Luxo em bom estado, marca='Versace, precoBase='300.0, idTransportadora='2]}}', dimensoes='{1=1, 2=1}', precoArtigos='530.0', taxaVintage='0.75', precoExpedicao='2.4907500000000002', ESTADO='EXPEDIDA'}=533.24075}'}
```

Como podemos ver aqui, a informação é apresentada de forma bastante confusa, mesmo havendo pouca informação presente.

Por este motivo, sentimos que a alteração mais importante seria organizar melhor como a informação é apresentada, ajudando assim a perceber esta melhor.