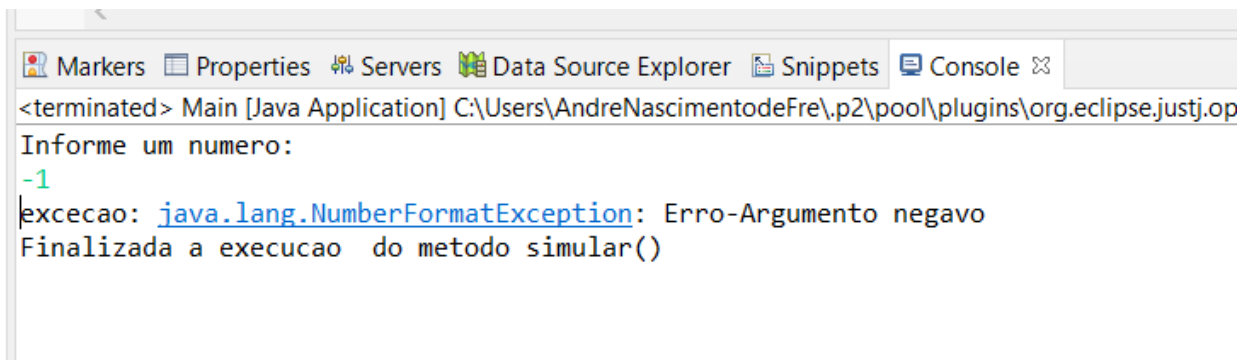


1. Utilizando throw, throws, try, catch e finally, crie uma classe que contenha um para capturar as exceções abaixo e imprima mensagens por meio do tratamento exceções.

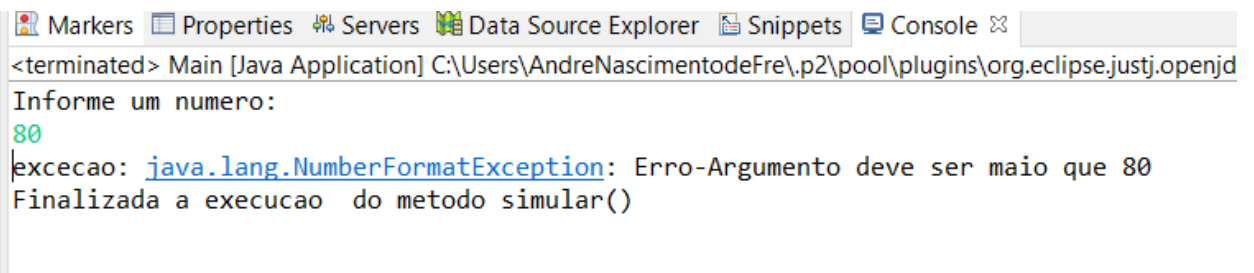
Teste cada uma das situações acima e mostre o resultado (pode ser com print da tela)

Se valor do argumento x for negavo, mensagem "Erro-Argumento negavo":



```
<terminated> Main [Java Application] C:\Users\AndreNascimento\Fre\p2\pool\plugins\org.eclipse.justj.openjd
Informe um numero:
-1
Excecao: java.lang.NumberFormatException: Erro-Argumento negavo
Finalizada a execucao do metodo simular()
```

Se valor do argumento x for menor ou igual a 1000, mensagem "Erro-Argumento deve ser maior que x"



```
<terminated> Main [Java Application] C:\Users\AndreNascimento\Fre\p2\pool\plugins\org.eclipse.justj.openjd
Informe um numero:
80
Excecao: java.lang.NumberFormatException: Erro-Argumento deve ser maio que 80
Finalizada a execucao do metodo simular()
```

Se o valor do argumento for maior que 1000, mensagem "Erro-Argumento muito grande":

```
<terminated> Main [Java Application] C:\Users\AndreNascimentoFre\.p2\pool\plugins\org.eclipse.just  
Informe um numero:  
10000  
Excecao: java.lang.NumberFormatException: Erro-Argumento muito grande  
Finalizada a execucao do metodo simular()
```

Simule uma divisão por zero, lançando automaticamente uma `ArithmeticException`

```
Markers Properties Servers Data Source Explorer Snippets  
<terminated> Main [Java Application] C:\Users\AndreNascimentoFre\.p2\  
Informe um numero:  
0  
Excecao: java.lang.ArithmeticException: / by zero  
Finalizada a execucao do metodo simular()
```

Crie um bloco catch que capture qualquer outro tipo de exceção (erro genérico), diferente das solicitadas e mostre tod.

```
Markers Properties Servers Data Source Explorer Snippets Console  
<terminated> Main [Java Application] C:\Users\AndreNascimentoFre\.p2\pool\plu  
Informe um numero:  
AAA  
Excecao: java.util.InputMismatchException
```