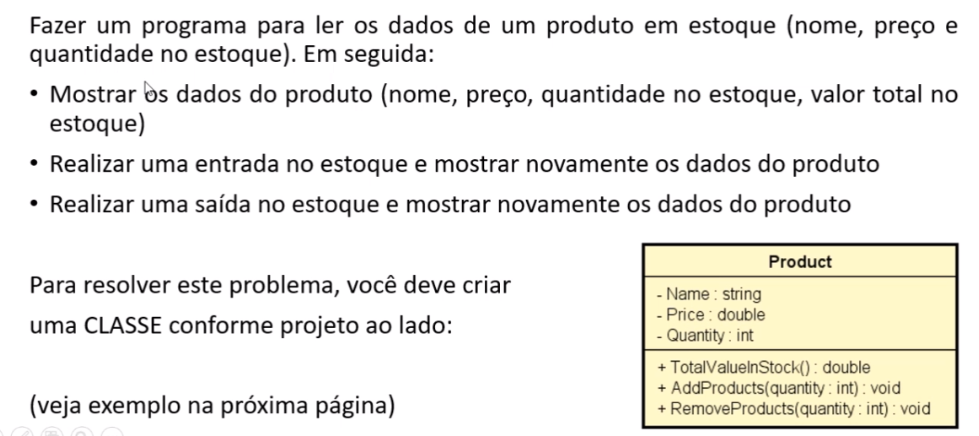
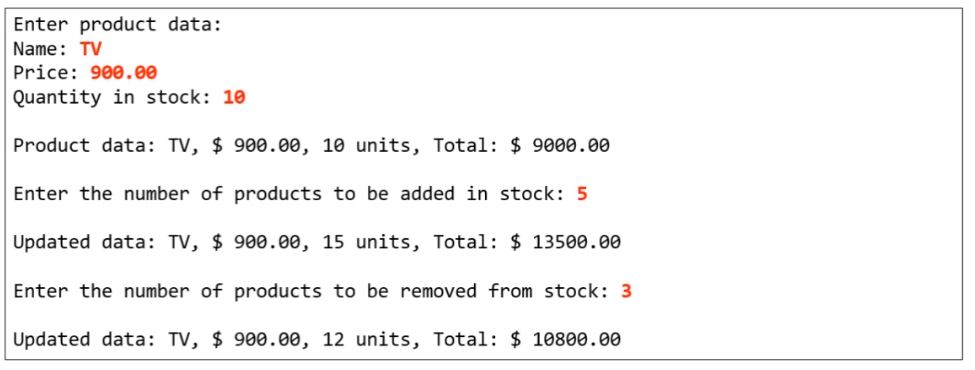
**Programação Orientada a Objetos -Java**

**Segundo Programa**



**Ao executarmos nosso programa ele ficará da seguinte forma:**

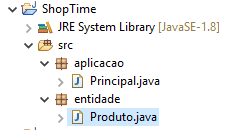


**Abrir o eclipse e criar um novo projeto**

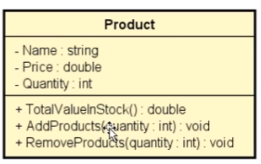
1 – Criar um projeto Java chamado ShopTime

2 – Criar classe com nome principal e pacote aplicação [marcar opção public static void main();

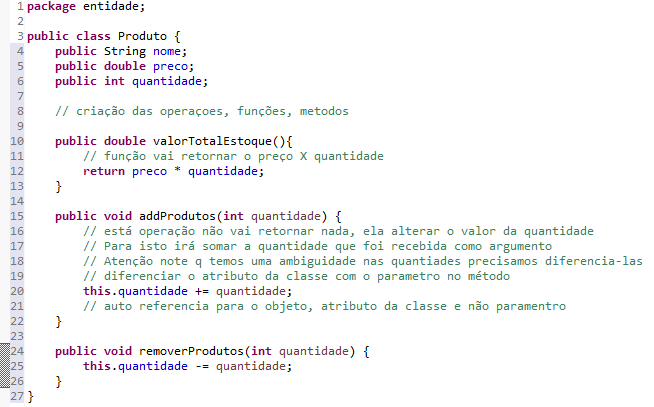
3 – criar classe produto no pacote entidade.



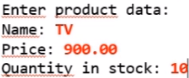
**Vamos agora implementar a classe Produto**

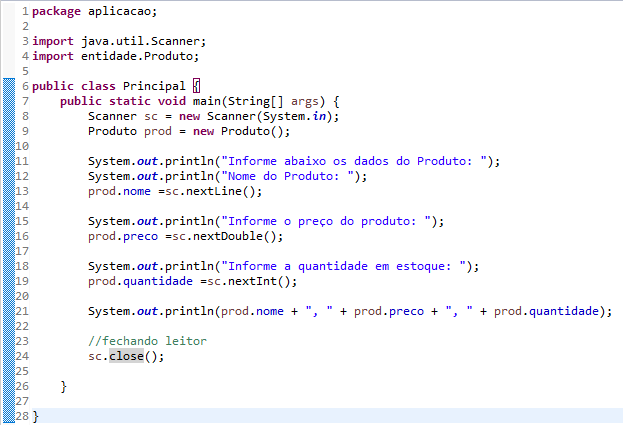


**Criando classe Produto**

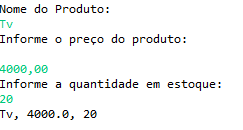


**Construção do Programa Para que ele leia os dados**

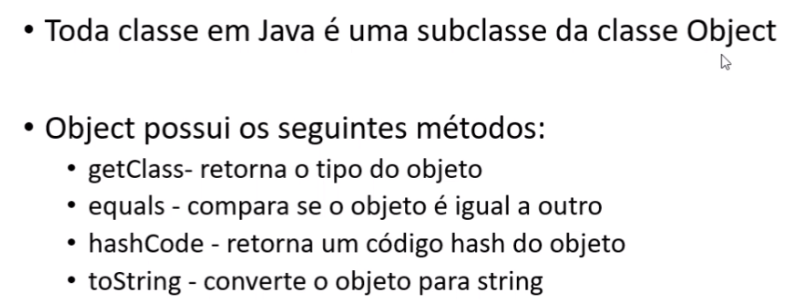




**Chegou a hora de fazer um teste seu programa.**



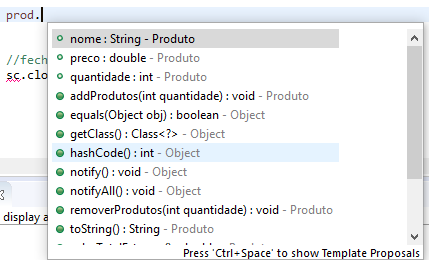
**Classe Object e método toString()**



Está questão de que uma classe é subclasse da outra falaremos melhor quando formos tratar do assunto de herança

Entenda que qualquer variável que tivermos no programa terá um tipo, por exemplo, tipo triangulo, tipo produto, etc. Cada um destes tipos é do tipo object, sendo que este tipo possui ações padrões como citado acima.

Observe os membros da classe Produto



Nome,preço,addProdutos.... equals não existe na classe produto, mas existe na variável prod.

O toString que iremos usar converte o objeto em uma String.

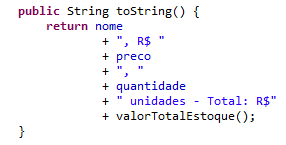
Informando que o Prod aparecerá em forma de string



Acima foi declarado uma operação. Veja bem, como a classe Produto também é um object, estamos sobrepondo a operação toString padrão que já vem no object

Preparando método toString para receber a formatação abaixo

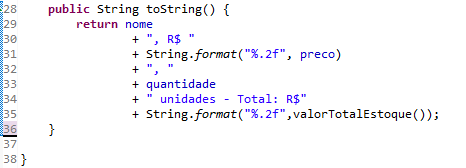




Teste



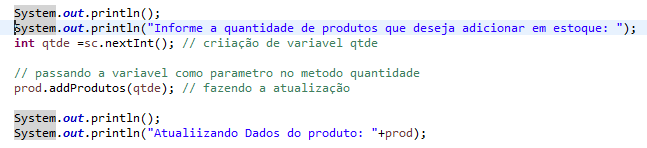
Acertando número de casas decimais



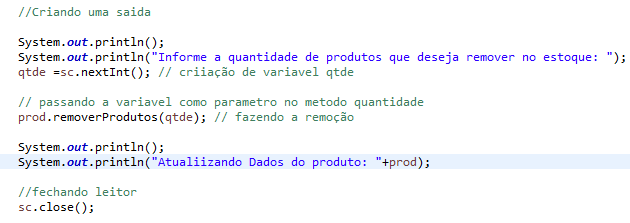
Mostrando na tela



**Adicionando novas quantidades em estoque**

****

**Criando a saída de produtos**

****