

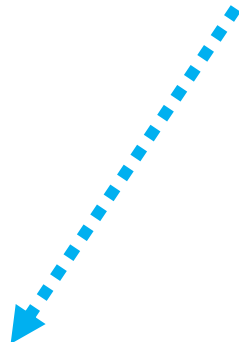
RAPPEL DU SUJET : Pour le 10 novembre, vous pusherez sur github l'exercice 2 (caisse du primeur) complètement implémenté : les 3 classes codées et leurs fichiers de tests unitaires codés. Vous ajouterez dans les fichiers suivi un document PDF qui présente vos classes et que vous appellerez "diagrammeUML.pdf".

Nous devons implémenter le code d'une caisse d'un primeur qui vend des fruits et des légumes au kilo ou à l'unité. Le prix d'un produit vendu au kilo est calculé à partir de son poids et de son prix au kilo. Le prix d'un produit vendu à l'unité est calculé en fonction du nombre de produit vendu et de son prix à l'unité.

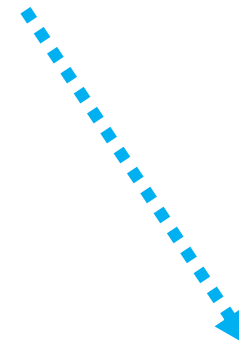
Proposé une hiérarchie de classe à trois classes pour répondre à ce problème : Produit, produitKilo, produitUnite.

Produit est une classe abstraite et sa fonction de calcul du prix est abstraite.

PRODUIT
<ul style="list-style-type: none"> <li>- nomprod : STRING</li> <li>- poids : INT</li> <li>- prixunite : INT</li> <li>- prixkilo : INT</li> <li>- nombreprod: INT</li> </ul>
<ul style="list-style-type: none"> <li>+ Produit ()</li> <li>+ __toString()</li> <li>+ __construct()</li> <li>+ calculprix : INT</li> <li>+ compteur : INT</li> </ul>



PRODUITKILO
<ul style="list-style-type: none"> <li>+poids : INT</li> <li>+nomprod : STRING</li> <li>+prixkilo : INT</li> </ul>
<ul style="list-style-type: none"> <li>+Produitkilo ()</li> <li>+__toString()</li> <li>+__construct()</li> <li>+ calculprix : INT</li> </ul>



PRODUITUNITE
<ul style="list-style-type: none"> <li>+ poids : INT</li> <li>+ nomprod : STRING</li> <li>+ prixunite</li> </ul>
<ul style="list-style-type: none"> <li>+ Produitunite ()</li> <li>+ __toString()</li> <li>+ __construct()</li> <li>+ calculprix : INT</li> </ul>