|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Donner | Classe Valide | Représentation Valide | Classe non Valide | Représentation  Non Valide |
| Nom\_Produit : String | CVn est String | Rvn=‘Produit A’ | CIn n’est pas String | RIn= 20 |
| Prix\_unit : double | CVp est int  Si(‘Produit A’) CVp=5 | RVp=5 | Si(‘Produit A’) {RIp !=5}  Sininon Rip[-infini,0]  RIp n’est pas double | RIp=-10  Rip = ‘ff’ |
| Quantite : int | Cvq est [0,+infini] | RVq=10 | RVq est[-inifni,-1] RVq n’est pas int | RIq=-50  RIq=’dfdfd’ |
| Reduction : double | Si( prix<200) CVr=0% Si(prix>=200)CVr=5% Si(prix>=1000)CVr=20% =>Si(quantite de Produit A>=45)= + Cvr=10% | Rvr = 20% | Si(prix >=200)CIr !=5% Si(prix>=1000)CIr !=20% =>Si(quantite de Produit A>=45) != + CIr !=10% CVr n’est pas double | RIr=200%  Rir=’kjkj’ |

Nombre de test =(1\*2\*1\*3)+(1+3+2+4)=16

(Produit A, 5, 100, 5% + 10%)

(Produit A, 5, 1000, 20%)

(Autre type que String, 20, 10, -10)

(Produit A, 20, 10, -10)

(Produit A, -10, 10, -10)

(Produit A, 'ff', 10, -10)

(Produit A, 5, -50, -10)

(Produit A, 5, 'dfdfd', -10)

(Autre type que String, 5, 10, -10)

(Produit A, 5, 10, -10)

(Produit A, 5, 10, 20%)

(Produit A, 5, 100, 20%)

(Produit A, 5, 1000, 20%)

(Produit A, 5, 45, 20% + 10%)

(Produit A, 5, 46, 20% + 10%)

(Produit A, 5, 44, 20%)  
Analyse aux limites pour la quantité de produits commandés :

* Test de la limite inférieure : Quantité = 0
* Test de la limite supérieure : Quantité = 1000 (juste en dessous de la limite pour la réduction de 20%)