

	Avantages	Désavantages
Design 2	<ul style="list-style-type: none"> <li>-Le code est très simple(les calculs pour le changements de coordonnées ont déjà été fournis dans PointCP)</li> <li>-Il est facile de créer des instances pour cette classe</li> </ul>	<ul style="list-style-type: none"> <li>-Utilise plus de mémoire que Design 6</li> </ul>
Design3	<ul style="list-style-type: none"> <li>-Le code est très simple(les calculs pour le changements de coordonnées ont déjà été fournis dans PointCP)</li> <li>-Il est très facile de créer des instances de cette classe</li> </ul>	<ul style="list-style-type: none"> <li>-Le temps d'exécution est moins rapide que pour le design 2 et 6</li> <li>-Utilise plus de mémoire que Design 6</li> </ul>
Design 6	<ul style="list-style-type: none"> <li>-Le code est encore plus simple car il suffit de mettre les méthodes présentes dans design2 et 3</li> <li>-On remarque que le temps d'exécution est toujours un peu plus rapide comparé au design 3</li> <li>Utilise moins de mémoire que Design 2 et 3</li> </ul>	<ul style="list-style-type: none"> <li>-On ne peut pas créer des instances d'une interface</li> <li>-Le temps d'exécution est moins rapide que pour le design 2</li> </ul>

Le professeur a précisé que le calcul ne se faisait pas pour chaque méthode mais pour l'ensemble des méthodes , raison pour laquelle on a 1 conversion(cartesian-polar ou vice versa des valeurs donnés en entrée), 1000000 getDistance() et 1000000 rotatePoint() car on a créé 1000000 instances avec des valeurs aléatoires et on a appelé ces méthodes pour chacune.

Operations (calcul temps execution)	Design 2	Design 3	Design 6
Test1: 1 conversion, 1000000 getDistance() , 1000000 rotatePoint() xorRho=5.6 yorTheta=8.9	379940587 nanoseconds	495633468 nanoseconds	Avec Design2 388515659 nanoseconds  Avec Design3 472010985 nanoseconds()
Test2: 1 conversion, 1000000 getDistance() , 1000000 rotatePoint() xorRho=2.5 yorTheta=1.9	366697568 nanoseconds	488583738 nanoseconds	Avec Design2 376396230 nanoseconds  Avec Design3 477544401 nanoseconds
Average Test1	379 nanoseconds	495 nanoseconds	Avec Design2 388 nanoseconds  Avec Design3 472 nanoseconds
Average Test2	366 nanoseconds	488 nanoseconds	Avec Design2 376 nanoseconds  Avec Design3 477 nanoseconds

Pour design 2, on a choisi (C) donc les valeurs dans le tableau ont été pris comme étant  $x$  et  $y$ . La sortie donne des coordonnées polaires.

Pour design 3, on a choisi (P) donc les valeurs dans le tableau ci-dessus ont été pris comme étant  $\rho$  et  $\theta$ . La sortie donne des coordonnées cartésiennes.

Nous avons comparé Design 2 et 6 puis noté les sorties. On a ensuite comparé Design 3 et 6 puis noté les sorties. Par exemple dans la colonne Design 6 on a des valeurs pour Design 2 et pour Design 3. Celui du design2 doit être comparé avec celui de la colonne design2. Celui du design3 doit être comparé avec celui de la colonne design3.

Les résultats obtenus nous ont permis de confirmer les observations de l'exercice 26.

Notre code peut comparer les temps d'exécution du design2 et 6 ou bien du design 3 et 6. Pour avoir une comparaison de coordonnées du même type, si on choisit (C) on mettra (C) pour Design6 également et on mettra les mêmes valeurs de  $x$  et  $y$  pour les 2 designs. Il en est de même pour Design3.

Car notre code peut également comparer le design 2 au design 6 ayant un type d'entrée (P) et avec n'importe quelles valeurs pour  $x, y, \rho$  et  $\theta$ . Donc on peut aussi comparer le temps pour des coordonnées polaires et des coordonnées cartésiennes directement.