**TP List, Set et Map**

1. Soit la classe Etudiant définit par : le nom, le prénom, le CNE (int) et le CIN(String).
2. Déclarez un constructeur avec paramètre, les getters/setters.
3. Redéfinissez les méthodes suivantes : toString, equals, hashCode(CNE modulo 512) et clone ;
4. Implémentez l’interface Comparable (permettant de comparer les étudiants par leur CIN);
5. Créez un comparateur permettant de comparer les étudiants par leur CNE.
6. Soit la classe ListEtudiant ;
7. Déclarez une List qui permet de stocker des étudiants;
8. Codez une fonction permettant de vérifier l’existence d’un étudiant ;
9. Codez une fonction permettant d’afficher les étudiants (utilisez un iterateur) ;
10. Codez une fonction permettant d’ajouter un étudiant;
11. Codez une fonction permettant de supprimer un étudiant;
12. Codez une fonction permettant de trier la Liste ;
13. Soit la classe HashSetEtudiant ;
14. Déclarez une hashSet qui permet de stocker des étudiants;
15. Codez une fonction permettant de vérifier l’existence d’un étudiant ;
16. Codez une fonction permettant d’afficher les étudiants (utilisez un iterateur) ;
17. Codez une fonction permettant d’ajouter un étudiant;
18. Codez une fonction permettant de supprimer un étudiant;
19. Codez une fonction permettant de videz la hashSet ;
20. Soit la classe HashMapEtudiant :
21. Déclarez une hashMap qui permet de stocker les pairs (CIN,Etudiant) ;
22. Codez une fonction permettant de rechercher un étudiant ;
23. Codez une fonction permettant de vérifier l’existence d’un CIN ;
24. Codez une fonction permettant d’afficher les étudiants (CIN => Etudiant) ;
25. Codez une fonction permettant d’afficher la liste des étudiants ;
26. Codez une fonction permettant d’afficher la liste des CINs;
27. Codez une fonction permettant de renvoyer l’étudiant associé à un CIN spécifique ;
28. Codez une fonction permettant de trouver le CIN associé à un étudiant spécifique ;
29. Codez une fonction permettant d’ajouter un étudiant à la map ;
30. Codez une fonction permettant de supprimer un étudiant dans la map ;
31. Codez une fonction permettant de joindre une map dans une autre ;
32. Soit la classe TreeMapEtudiant :
33. Déclarez une TreeMap (CinTreeMap) qui permet de stocker les pairs (CIN,Etudiant) ;
34. Codez une fonction permettant de rechercher le premier étudiant ;
35. Codez une fonction permettant de rechercher le dernier étudiant ;
36. Codez une fonction permettant d’afficher les étudiants (triés par CIN) ;
37. Codez une fonction permettant de retourner les étudiants dont les CINs sont strictement inférieures à celle fournie en premier paramètre et supérieures ou égales à celle fournie en second paramètre ;
38. Codez une fonction permettant de retourner les étudiants dont les CINs sont supérieures ou égales à celle fournie en paramètre
39. Codez une fonction permettant de retourner les étudiants dont les CINs sont strictement inférieures à celle fournie en paramètre ;
40. Codez une fonction permettant de retourner le comparateur utilisé dans la TreeMap ;
41. Déclarez une deuxième TreeHashMap (CneTreeMap) qui permet de trier les pairs (CIN, Etudiant) en fonction de la CNE.
42. Codez une fonction permettant de trouver le CIN associé à un étudiant spécifique ;
43. Codez une fonction permettant d’ajouter un étudiant à la map ;
44. Codez une fonction permettant de supprimer un étudiant dans la map ;
45. Codez une fonction permettant de joindre une map dans une autre.