**algorithme** remplissageDesDonnees

**fonction** convertirSTLtoDonnees (stl : **Fichier**) : **Mesh**

**début**

1.    points ← Tableau[0..n]

2.    vertices ← Tableau[0..n]

3.    faces : Tableau[0..n/3]

4.

5. **Tantque non** EOF **faire**

6.

7. p1 ← lire\_ligne(stl)

8. p2 ← lire\_ligne(stl)

9. p3 ← lire\_ligne(stl)

10.

11. **si** points contient p1alors

      12. p1  ← p1bis

      13. sinon ajouter(p1, points)

14. **fsi**

15. //pareil pour les deux autres points

16.

17. v1 ← new Vertex(p1)

18. h1 ← new Halfedge(v1)

19. v1.halfedge ← h1

20. v2 ← new Vertex(p2)

21. h2 ← new Halfedge(v2)

22. v2.halfedge ← h2

23. v3 ← new Vertex(p3)

24. h3 ← new Halfedge(v3)

25. v3.halfedge ← h3

26.

27. h1.next ← h2

28. h2.previous ← h1

29. h2.next ← h3

30. h3.previous ← h2

31. h3.next ← h1

32. h1.previous ← h3

33. nouvelleFace ← new Face(h1)

34. h1.face ← nouvelleFace

35. h2.face ← nouvelleFace

36. h3.face ← nouvelleFace

37.

38. ajouter(faces, nouvelleFace)

39.

40. //détection des arêtes opposées pour compléter la structure de données

41. vertexPoint1 ← vertices.filter( element → element.point == p1 )

42. **si** vertexPoint1.length > 0

43.      **alors**

44.             halfedgeOppose ← vertexPoint1.map(element -> element.halfedge.previous)

          .filtrer(element -> element.vertex.point == p2)

          .findFirst()

45.             **si** halfedgeOppose non null alors

46.                   **si** halfedgeOppose.opposite est null alors

47.                        halfedgeOppose.opposite ← h1

48.                        h1.opposite ← halfedgeOppose.opposite

49.                        sinon erreur

50.                   **finsi**

51.            **finsi**

52. **finsi**

53. //pareil pour les deux autres points

54.

55. ajouter(vertices, v1)

56. ajouter(vertices, v2)

57. ajouter(vertices, v3)

58. **ftant**

59.

60. points ← trierPoints(points)

61. //Trier vertex par points

62. //Trier face par vertex

63.

64. mesh ← Mesh(faces, vertices, points)

65. **retourne** mesh

**fin**