

Agenda

1. Variable and Constant Declaration Defects (VC)

1. No, por ejemplo el parámetro que se pasa a la función addEntry(Entry p).
2. No, no hay confusión. Por ejemplo de línea 22 a 175.
3. No, en la función loadAgenda la variable cad no se inicializa y justo después la compara con una condición.
4. No, por ejemplo las variables first y numEntries se usan en varias funciones.
5. No, no hay.
6. No, no hay macros en Java.
7. No, tal y como está programado tiene que quedarse así, por ejemplo numEntries.

2. Function Definition Defects (FD)

8. Si, se entienden bien los nombres de las funciones. Por ejemplo loadAgenda
9. Si los checkea.
10. Si, por ejemplo en la función addEntry tiene que retornar boolean.

3. Class Definition Defects (CD)

11. Si, son correctos. Por ejemplo línea 16 y 26.
12. Si, pueden acceder.
13. No, no es necesaria una clase base.
14. No, ya está muy simplificada. Ejemplo: clase agendaNode.

4. Computation/Numeric Defects (CN)

15. No se produce overflow porque no hay una carga de datos excesiva. No se produce underflow, ya que no hay un procesamiento mayor a la carga.
16. Si, es correcta la suposición.
17. Hay comentarios que evitan la ambigüedad pero no paréntesis con ese fin.

5. Comparision/Relational Defects (CR)

18. No, en loadAgenda hay comparaciones que no son válidas.
19. Si, todos los boolean están correctos.
20. Sí, puede haber problemas en loadAgenda.

6. Control Flow Defects (CF)

21. Si, ejemplo línea 133.
22. No, es posible que de algún error por igualar cad a la línea del buffer línea 171.

- 23. Si, no tenemos bucles con más de una salida.
- 24. No hay ningún switch.
- 25. No hay ningún switch.
- 26. Si, es correcta pero no muy profundo.
- 27. No, no sería necesario. Línea 78.
- 28. Parecen correctas pero no están marcadas de ninguna forma.
- 29. No, es probable que falle en la línea 171 .
- 30. Si, se evitan porque no se encuentran en el código.

7. Input-Output Defects (IO)

- 31. Sí, línea 129.
- 32. No son consistentes con el idioma.
- 33. No, línea 131.
- 34. No, función saveAgenda.
- 35. No se ha encontrado ninguno.
- 36. Sí, línea 134.

8. Module Interface Defects (MI)

- 37. Sí, línea 55.
- 38. Sí, ya que no tenemos este tipo de valores.

9. Comment Defects (CM)

- 39. Si, todo tiene comentarios.
- 40. No, en las líneas 13 y 14.
- 41. Si, se entiende correctamente.
- 42. Si, es consistente.
- 43. Si, están de acuerdo.
- 44. Si ayudan.
- 45. Si, son suficientes.
- 46. Hay los justos y necesarios.

10. Packing Defects (LP)

- 47. Sí, solo contiene una clase línea 129.
- 48. Si, son menores a 60.
- 49. Si, son menores a 2000.

11. Modularity Defects (MO)

- 50. Solo hay un paquete.
- 51. Solo hay un paquete.
- 52. No.
- 53. Si, se usa ArrayList y es correcto su uso.

12. Performance Defects (PE) [Optional]

- 54. Es probable que existan mejores estructuras para realizar el programa.
- 55. Si, ya que no se ha gastado recursos para hacer pruebas.
- 56. Sí, sin duda.
- 57. Si, se van guardando en la agenda.
- 58. No dado que no conocemos la longitud de los bucles.
- 59. No, todas aportan al resultado.
- 60. No, ya que sería muy complicado hacer que tenga la misma funcionalidad.
- 61. No, no hay dos bucles iguales.