

# Paser

## 1. Variable and Constant Declaration Defects (VC)

1. No, ya que los nombres convencionales serían en inglés (adress, city, name, etc.).
2. Sí, por ejemplo POB (población) y PRO (provincia). Quedaría más claro si PRO se llamara PROV.
3. Sí, están todas correctamente inicializadas.
4. En esta clase no hay variables no locales.
5. Sí, los atributos estáticos declarados al principio de la clase podrían ser constantes.
6. No, no hay macros en Java.
7. No, ya que los diferentes campos de la agenda, así como las entradas o las líneas pueden ir cambiando..

## 2. Function Definition Defects (FD)

8. Sí, los nombres de las funciones lo cumplen.
9. No, no hay comprobaciones previas.
10. Sí, todas las funciones devuelven los valores correctos.

## 3. Class Definition Defects (CD)

11. Sí, la clase tiene su constructor.
12. No, ya que la clase es pública.
13. En este caso, no.
14. No, ya que no hay herencia en este ejemplo.

## 4. Computation/Numeric Defects (CN)

15. No se produce overflow porque la cantidad de caracteres que puede almacenar un String es muy grande, y en este caso, sólo almacenamos unos cuantos datos personales. No se produce underflow, ya que no hay un procesamiento mayor a la carga.
16. No, ya que el método insertLine requeriría de estructuras de control que alterarían el orden del mismo. .
17. Hay comentarios que evitan la ambigüedad pero no paréntesis con ese fin.

## 5. Comparision/Relational Defects (CR)

18. Si, ejemplo línea 76.
19. No hay bucles.

20. No hay efectos secundarios de las comparaciones.

## **6. Control Flow Defects (CF)**

- 21. No hay bucles.
- 22. No hay bucles.
- 23. No hay bucles.
- 24. No hay ningún switch.
- 25. No hay ningún switch.
- 26. No hay bucles.
- 27. No, no sería necesario.
- 28. No tenemos null bodied control structures.
- 29. Si, terminan todas.
- 30. Si, se evitan porque no se encuentran en el código.

## **7. Input-Output Defects (IO)**

- 31. No se utilizan ficheros.
- 32. No se utilizan ficheros.
- 33. No se utilizan ficheros.
- 34. No se utilizan ficheros.
- 35. No se utilizan ficheros.
- 36. No se utilizan ficheros.

## **8. Module Interface Defects (MI)**

- 37. Si, son correctas. Ejemplo línea 77.
- 38. Si, ya que no tenemos este tipo de valores.

## **9. Comment Defects (CM)**

- 39. Si, tiene un comentario al comienzo de la clase.
- 40. No, no tienen.
- 41. No, no tiene comentarios cada función.
- 42. No tienen todas.
- 43. No hay comentarios.
- 44. No ayudan porque no hay comentarios.
- 45. No, no hay suficientes.
- 46. No, de hecho hacen falta más.

## **10. Packing Defects (LP)**

- 47. No hay archivos.
- 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, no hay código duplicado.
- 53. No hay framework classes.

## **12. Performance Defects (PE) [Optional]**

- 54. Es posible que existan mejores algoritmos para realizar las funciones del programa.
- 55. Si, ya que no se han gastado recursos para hacer pruebas.
- 56. Sí, sin duda.
- 57. Si, si se guardan.
- 58. No hay bucles.
- 59. No, hay bucles.
- 60. No, hay bucles.
- 61. No, hay bucles.