
1. Qual classe é usada para gravar texto em arquivos de forma simples?

Resposta: A classe `PrintWriter`.

2. Quais métodos do `PrintWriter` são usados para gravar texto?

Resposta: `print()` e `println()`.

3. Para que serve o método `printf()` no `PrintWriter`?

Resposta: Permite formatar a saída de texto conforme especificadores, como `% .2f`.

4. Como abrir um arquivo texto para adicionar conteúdo sem sobrescrever?

Resposta: Usar `new FileOutputStream(arquivo, true)` e passar para o `PrintWriter`.

5. O que são caracteres de escape?

Resposta: Sequências especiais usadas para representar caracteres como tab (`\t`), nova linha (`\n`), aspas, etc.

6. Qual caractere de escape representa nova linha?

Resposta: `\n`.

7. Qual diferença entre separadores de linha em Windows e Linux?

Resposta: Windows usa `\r\n`; Linux usa `\n`.

8. Qual método de PrintWriter produz automaticamente o separador de linha da plataforma?

Resposta: `println()`.

9. Como se cria um PrintWriter a partir de um nome de arquivo?

Resposta: `new PrintWriter("arquivo.txt")`.

10. Qual classe é utilizada para leitura de arquivos texto?

Resposta: A classe Scanner.

11. Cite três métodos de leitura do Scanner.

Resposta: `nextLine()`, `nextInt()`, `nextDouble()`.

12. O que o método hasNext() faz?

Resposta: Retorna true se ainda há conteúdo a ser lido.

13. Como ler uma palavra usando Scanner?

Resposta: Usar o método `next()`.

14. Como se retorna uma linha inteira usando Scanner?

Resposta: Com o método `nextLine()`.

15. Qual é a função do mapa de caracteres (character set)?

Resposta: Representar o conjunto de caracteres disponíveis e seus códigos.

16. Cite três mapas de caracteres.

Resposta: US-ASCII, Latin-x, Unicode.

17. Qual é o objetivo do Unicode?

Resposta: Representar todos os caracteres de todas as linguagens.

18. O que é code point?

Resposta: O número que identifica um caractere em um mapa.

19. UTF-8 utiliza quantos bytes para caracteres ocidentais?

Resposta: 1 byte.

20. Quantos bytes UTF-16 pode usar?

Resposta: 2 ou 4 bytes.

21. Quantos bytes UTF-32 utiliza?

Resposta: Sempre 4 bytes.

22. Qual a representação do caractere € em UTF-8 no exemplo do slide?

Resposta: 0xE2 0x82 0xAC.

23. Qual a representação do € em UTF-16 no exemplo?

Resposta: 0xAC 0x20.

24. Qual classe permite gravar texto com codificação específica?

Resposta: PrintWriter usando um construtor com parâmetro de charset.

25. Como se cria um PrintWriter em UTF-8?

Resposta: new PrintWriter("arq.txt", "UTF-8").

26. Como Scanner lê um arquivo com codificação específica?

Resposta: new Scanner(new File("arquivo.txt"), "UTF-8").

27. Qual método de PrintWriter grava texto sem pular linha?

Resposta: print().

28. Qual método fecha o arquivo no PrintWriter?

Resposta: close().

29. O que Scanner(File) faz?

Resposta: Abre o arquivo para leitura.

30. O que acontece se um arquivo for aberto com FileOutputStream sem "append"?

Resposta: Ele é recriado, sobrescrevendo o conteúdo. (*Informação geral coerente com classes de texto e binários; aplica-se também na composição PrintWriter + FileOutputStream.*)

1 – Sobre gravação de arquivos texto:

- a) () PrintWriter grava texto em arquivos.
- b) () print() sempre pula linha.
- c) () println() usa o separador de linha da plataforma.
- d) () printf() permite escrever texto formatado.

Gabarito: V, F, V, V

2 – Sobre caracteres de escape:

- a) () \n representa nova linha.
- b) () \t representa tabulação.
- c) () \\ representa barra invertida.
- d) () \\r apaga a linha atual.

Gabarito: V, V, V, F
(\\r apenas retorna o cursor ao início da linha)

3 – Sobre PrintWriter:

- a) () Pode ser criado a partir de um FileOutputStream.
- b) () Precisa sempre sobrescrever o arquivo.
- c) () close() libera o arquivo.
- d) () println() gera \\r\\n em todos os SOs.

Gabarito: V, F, V, F

4 – Sobre Scanner:

- a) () Scanner lê arquivos texto.
- b) () nextInt() lê inteiros.
- c) () hasNext() verifica se há conteúdo.
- d) () nextLine() lê apenas uma palavra.

Gabarito: V, V, V, F

5 – Sobre leitura com Scanner:

- a) () next() lê a próxima palavra.
- b) () nextLine() lê a linha inteira.
- c) () nextDouble() lê números decimais.
- d) () Scanner não lê números.

Gabarito: V, V, V, F

6 – Sobre mapas de caracteres:

- a) () ASCII representa apenas caracteres ingleses.
- b) () Unicode representa caracteres de várias linguagens.
- c) () Latin-x inclui caracteres latinos.
- d) () Code point é sempre igual à representação binária no Unicode.

Gabarito: V, V, V, F

7 – Sobre codificações Unicode:

- a) () UTF-8 usa entre 1 e 4 bytes.
- b) () UTF-16 usa sempre 2 bytes.
- c) () UTF-32 usa 4 bytes.
- d) () UTF-8 é eficiente para idiomas ocidentais.

Gabarito: V, F (pode usar 2 ou 4), V, V

8 – Sobre codificações no PrintWriter:

- a) () É possível criar PrintWriter com UTF-8.
- b) () O charset é informado no construtor.
- c) () Sem charset, UTF-32 é o padrão.
- d) () write() escreve caracteres com a codificação definida.

Gabarito: V, V, F, V

9 – Sobre leitura com codificação:

- a) () Scanner pode ler arquivo em UTF-8.
- b) () Basta informar a codificação no construtor.

- c) () Sem charset, a codificação padrão da JVM é usada.
- d) () Scanner falha ao ler UTF-8.

Gabarito: V, V, V, F

10 – Sobre separadores de linha:

- a) () Windows usa \r\n.
- b) () Linux usa \n.
- c) () println() adapta o separador conforme o SO.
- d) () Todos os SOs usam o mesmo separador.

Gabarito: V, V, V, F

11 – Sobre métodos de leitura:

- a) () nextLine() consome o restante da linha atual.
- b) () next() lê até o próximo espaço.
- c) () nextInt() ignora caracteres não numéricos.
- d) () hasNext() retorna false se o arquivo acabou.

Gabarito: V, V, F, V

12 – Sobre saída formatada:

- a) () printf() aceita especificadores como %.2f.
- b) () print() não aceita formatação.
- c) () printf() é equivalente ao format().
- d) () print() remove caracteres especiais.

Gabarito: V, V, V, F

13 – Sobre PrintWriter e FileOutputStream:

- a) () PrintWriter pode receber FileOutputStream no construtor.
- b) () O modo append permite incluir novas linhas.
- c) () Sem append o arquivo é sobreescrito.
- d) () append impede adicionar novas linhas.

Gabarito: V, V, V, F

14 – Sobre UTF-8 e UTF-16:

- a) () UTF-8 usa 1 byte para caracteres latinos.
- b) () UTF-16 representa todos os símbolos em 2 bytes.
- c) () UTF-8 é mais compacto para idiomas latinos.
- d) () UTF-32 é mais compacto que UTF-16.

Gabarito: V, F, V, F

15 – Sobre leitura de arquivos texto:

- a) () Scanner pode ler números e textos do mesmo arquivo.
- b) () nextInt() consome apenas números.
- c) () nextLine() captura tudo até o fim da linha.
- d) () Scanner não fecha arquivos automaticamente.

Gabarito: V, V, V, V