

01. [Sebesta, 2000] Quais são os três recursos característicos das linguagens orientadas a objeto?
02. [Sebesta, 2000] Qual é a diferença entre uma variável de classe e uma variável de instância?
03. [Sebesta, 2000] O que é um método de sobreposição?
04. [Sebesta, 2000] Descreva uma situação em que a vinculação dinâmica é uma grande vantagem em comparação com sua ausência.
05. [Sebesta, 2000] O que é um método virtual?
06. [Sebesta, 2000] Descreva brevemente as sete questões de projeto usadas, neste capítulo, para as linguagens orientadas a objeto.
07. [Sebesta, 2000] Qual é o protocolo de mensagem de um método?
08. [Sebesta, 2000] Por que as classes da Smaltalk podem responder a mensagens?
09. [Sebesta, 2000] Explique as ações da instrução Smaltalk `result <- first * second`
10. [Sebesta, 2000] Quais são as quatro partes de uma definição de classes da Smaltalk?
11. [Sebesta, 2000] Explique como as mensagens Smaltalk são vinculadas a métodos. Quando isso se realiza?
12. [Sebesta, 2000] Qual verificação de tipo é feita na Smaltalk? Quando ela se realiza?
13. [Sebesta, 2000] Qual tipo de herança a Smaltalk suporta?
14. [Sebesta, 2000] Quais são os dois principais efeitos que a Smaltalk teve sobre a computação?
15. [Sebesta, 2000] Para qual propósito a pseudo-variável `super` da Smaltalk serve?
16. [Sebesta, 2000] Essencialmente, todas as variáveis Smaltalk são de um único tipo. Qual é esse tipo?
17. [Sebesta, 2000] Quantos parâmetros existem em uma mensagem binária Smaltalk?
18. [Sebesta, 2000] Explique as regras de precedência das expressões Smaltalk.
19. [Sebesta, 2000] Como se pode forçar um bloco Smaltalk a ser executado?
20. [Sebesta, 2000] Para qual propósito a pseudo-variável `self` da Smaltalk serve?
21. [Sebesta, 2000] De onde os objetos do C++ são alocados?
22. [Sebesta, 2000] Como os objetos alocados no *heap* em C++ são desalocados?
23. [Sebesta, 2000] Todas as subclasses do C++ são subtipos?
24. [Sebesta, 2000] Sob quais circunstâncias uma chamada a método do C++ é vinculada estaticamente a um método?
25. [Sebesta, 2000] Qual inconveniente há em permitir que os projetistas especifiquem quais métodos podem ser vinculados estaticamente?
26. [Sebesta, 2000] Explique a diferença entre os dois usos de `private` no C++.
27. [Sebesta, 2000] O que é uma função amiga (`friend`) no C++?
28. [Sebesta, 2000] Como o sistema de tipos do Java é diferente do sistema de tipos do C++?

29. [Sebesta, 2000] De onde os objetos Java podem ser alocados?
30. [Sebesta, 2000] Como os objetos Java são alocados?
31. [Sebesta, 2000] Todas as subclasses Java são subtipos?
32. [Sebesta, 2000] Sob quais circunstâncias um método Java é vinculado estaticamente a um método?
33. [Sebesta, 2000] Todas as subclasses Ada 95 são subtipos?
34. [Sebesta, 2000] Como a chamada a um subprograma na Ada 95 é especificada para ser vinculada dinamicamente a uma definição de subprograma? Quando essa decisão é tomada?
35. [Sebesta, 2000] Qual propósito tem uma cláusula `creation` na Eiffel?
36. [Sebesta, 2000] Como um recurso Eiffel é definido para ser visível em subclasses, mas não em clientes?
37. [Sebesta, 2000] O que o qualificador `{nome}` faz quando ele aparece em um recurso Eiffel?
38. [Sebesta, 2000] Como objetos Eiffel alocados no *heap* são desalocados?
39. [Sebesta, 2000] Sob quais circunstâncias uma chamada a método Eiffel é vinculada estaticamente a um método?
40. [Sebesta, 2000] Como uma subclasse Eiffel pode ser definida para não ser um subtipo?
41. [Sebesta, 2000] Escreva a seguinte estrutura de laço Pascal em Smaltalk:

```
while cont < 100 do
begin
  soma := soma div (2 * cont - 1);
  cont := cont + 1;
end;
```

42. [Sebesta, 2000] Escreva o seguinte laço `for` Pascal em Smaltalk:

```
for indice := 10 downto 1 do
  soma := soma + indice;
```

43. [Sebesta, 2000] Escreva a seguinte construção de seleção Pascal em Smaltalk:

```
if cont < 10 then
  resposta := 1;
else
begin
  resposta := 0;
  cont := 0;
end;
```

44. [Sebesta, 2000] Escreva a seguinte estrutura de laço `while` C em Smaltalk:

```
while (cont < 100) {
  soma /= (2 * cont - 1);
  cont++;
}
```

45. [Sebesta, 2000] Escreva a seguinte estrutura de laço `for` C em Smaltalk:

```
for (indice = 10; indice > 0; indice--)
  soma += indice;
```

46. [Sebesta, 2000] Escreva a seguinte construção de seleção C em Smaltalk:

```
if (cont < 10)
  resposta = 1;
else
  resposta = cont = 0;
```

47. [Sebesta, 2000] Escreva um método de instância Smaltalk que aceite quatro valores inteiros, cujos dois primeiros são o numerador e o denominador de uma fração e os dois últimos, de maneira semelhante, representam outra fração. Seu método deve retornar um objeto *array* de dois elementos que representa o numerador e o denominador do produto das duas frações dadas.

48. [Sebesta, 2000] Compare a vinculação dinâmica da Eiffel, do C++, da Smaltalk, da Ada 95 e do Java.

49. [Sebesta, 2000] Compare os controles de acesso de entidades de classe do Eiffel, do C++, da Smaltalk, da Ada 95 e do Java.

50. [Sebesta, 2000] Compare a herança simples da Eiffel, do C++, da Smaltalk, da Ada 95 e do Java.

51. [Sebesta, 2000] Compare a herança múltipla da Eiffel e do C++.

52. [Sebesta, 2000] Compare a herança múltipla do C++ com aquela oferecida pelas interfaces em Java.

53. Dado o código a seguir

```
class B extends A {
  int getID() {
    return id;
  }
}

class C {
  public int name;
}

class A {
  C c = new C();
  public int id;
}
```

Duas opções são verdadeiras com relação às instâncias das classes listadas acima. Quais?

- () A é um B
- () C é um A
- () A tem um C
- () B tem um A
- () B tem um C