



ATENÇÃO: Essa prova contém 25 questões. Verifique se a prova está completa e legível. Qualquer problema, fale com o fiscal de sala. Marque as respostas no cartão usando caneta azul ou preta. A duração da prova é de 3 horas.

01 - De acordo com o texto a seguir e com os conceitos de orientação a objetos, marque a opção adequada ao problema descrito: Um estacionamento possui manobristas que estacionam veículos. Os veículos têm a capacidade de se mover. Um carro é um veículo que possui 4 rodas, marca e modelo. Um Fiat Uno (carro) foi entregue ao manobrista para estacionar. Uma moto Honda Bis que também é um veículo de 2 rodas foi estacionada pelo mesmo manobrista.

- a) Manobrista é uma classe que implementa as interfaces Carro e Moto.
- b) Carro e Moto são classes abstratas que possuem a propriedade mover.
- c) Marca, modelo e quantidade de rodas não podem ser modelados por meio de atributos.
- d) Veiculo é uma interface que define o método mover e é implementada pelas classes Carro e Moto.
- e) Fiat Uno e Honda Bis são objetos da classe abstrata Veiculo.

02 - Sabendo que o tipo ponteiro em uma determinada arquitetura requer 2 bytes de armazenamento, então o espaço ocupado em memória pelos ponteiros de uma lista circular duplamente encadeada com n elementos é:

- a) $2n$
- b) $4n$
- c) $8n$
- d) $16n$
- e) $32n$



03 - Na orientação a objetos, qual o conceito que abrange a possibilidade de enviar uma mesma mensagem a diferentes objetos, de forma que cada um deles responda de modo mais apropriado ao seu contexto?

- a) Herança
- b) Encapsulamento
- c) Polimorfismo
- d) Paralelismo
- e) Passagem por referência

04 - Quanto aos conceitos básicos de Estruturas de Dados, analise as afirmativas abaixo, dê valores Verdadeiro (V) ou Falso (F) e assinale a alternativa que apresenta a sequência correta de cima para baixo.

- () Uma pilha é uma estrutura de dados que usa o princípio de Last In First Out (LIFO).
- () As filas são fundamentalmente compostas por duas operações básicas: push e pop.
- () As filas são estruturas de dados que usa o princípio First In First Out (FIFO).
- () Pode-se inserir elementos de tipos diferentes em um arranjo (array), desde que a capacidade de armazenamento sofra alterações.

- a) V - V - F - V
- b) V - F - V - F
- c) F - F - V - F
- d) F - F - V - V
- e) V - F - F - V

05 - Levando em conta as estruturas de dados, relacione as colunas.

- (1) Pilha
- (2) Fila
- (3) Árvore-B
- (4) Árvore AVL
- (5) Lista Encadeada
- (6) Hash

(3) Árvore binária balanceada pela altura de cada nó.

(4) Árvore de busca multidirecional e balanceada.

(1) Também chamada de lista LIFO (Last-In, First-Out).

(5) Cada item é chamado nó e contém a informação do endereço seguinte.

(2) Também chamada de lista FIFO (First-In, First-Out).

(6) Estrutura que armazena dados através de chave-valor. A ordem CORRETA de associação, de cima para baixo, é:

- a) 4, 3, 6, 1, 5, 2
- b) 3, 4, 1, 5, 2, 6
- c) 4, 3, 2, 6, 5, 1
- d) 3, 4, 2, 5, 1, 6
- e) 4, 3, 6, 5, 2, 1

06 - Considere a seguinte proposição Z: $(p \wedge q) \wedge (\neg r)$. Em seguida, marque a alternativa abaixo que é logicamente equivalente à negação da proposição Z.

- a) $p \rightarrow (q \rightarrow r)$
- b) $p \rightarrow (r \rightarrow q)$
- c) $(p \rightarrow q) \rightarrow r$
- d) $(p \rightarrow r) \rightarrow q$
- e) $(r \rightarrow q) \rightarrow p$

$$(p \wedge q) \wedge (\neg r)$$

07 - Sobre conceitos de Orientação a Objetos (OO) podemos definir:

I – Comportamento que as suas instancias podem adotar. *Método*

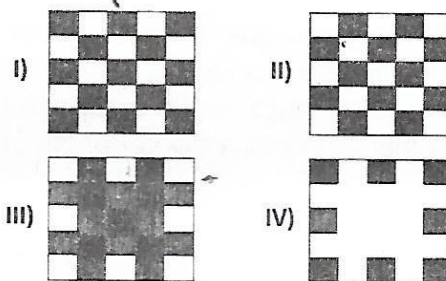
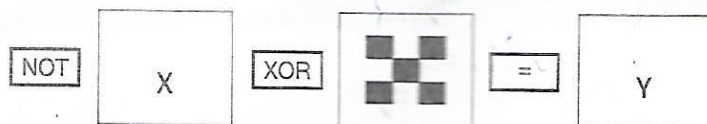
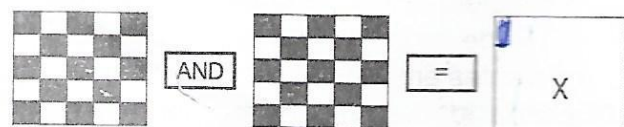
II – Permite a especialização a partir de um tipo mais genérico.

III – Permite que referências de classes mais abstratas executem o comportamento de objetos de suas subclasses.

Os conceitos da OO indicados em I, II e III são, respectivamente:

- a) Objeto, Classe e Herança.
- b) Método, Objeto e Polimorfismo.
- c) Polimorfismo, Herança e Método.
- d) Método, Herança e Polimorfismo.
- e) Atributo, Método e Herança.

08 - Nas figuras abaixo, um pixel preto representa o booleano VERDADEIRO e um pixel branco representa um booleano FALSO. Marque a alternativa correta segundo as operações booleanas indicadas e aplicadas pixel a pixel.



- a) x equivale a imagem I e y equivale a imagem II.
- b) x equivale a imagem II e y equivale a imagem IV.
- c) x equivale a imagem I e y equivale a imagem III.
- d) x equivale a imagem I e y equivale a imagem IV.
- e) x equivale a imagem II e y equivale a imagem III.

09 - Em Swift, structs são Value Types (passadas por valor). Leia o código abaixo e assinale a resposta correta.

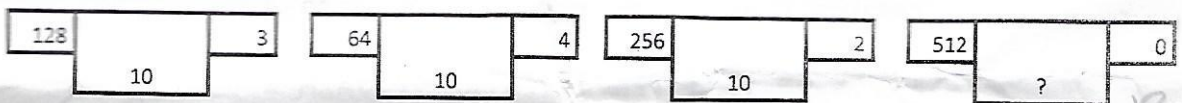
```
01 struct Resolution {
02     var width: Float -
03     var height: Float -
04 }
05
06 var hd = Resolution(width: 1920, height: 1080)
07 var cinema = hd
08 cinema.width = 2048
09 print("\(hd.width)")
```

CÓPIA DA REFERÊNCIA
HD != CINEMA

- a) A instância referenciada pela variável hd receberá o valor 2048 na propriedade width.
 b) As variáveis cinema e hd irão referenciar a mesma instância e essa instância receberá o valor 2048 na propriedade width.
 c) A variável cinema irá referenciar uma instância diferente da referenciada pela variável hd. A instância referenciada por cinema receberá o valor 2048 na propriedade width.
 d) O valor 2048 será exibido no terminal. x
 e) O código não irá compilar. x

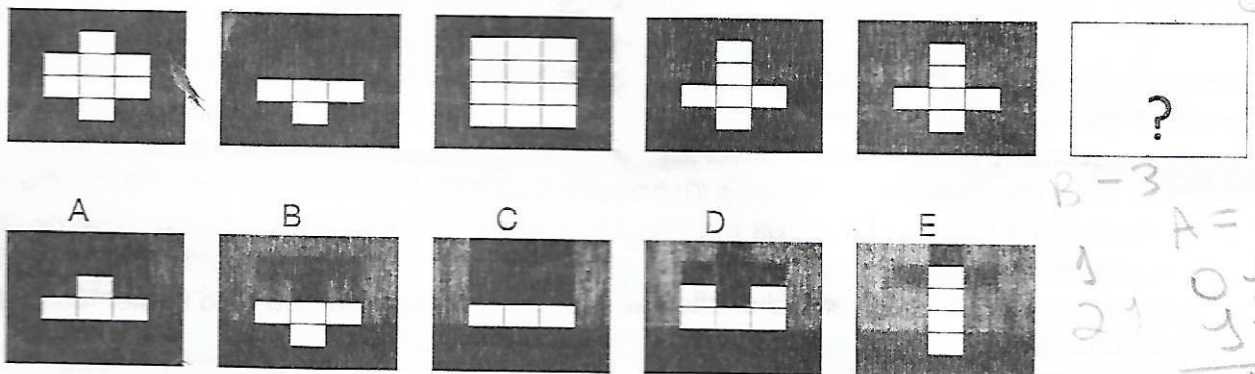
A = 3
B = 2
A = 1
A

10 - Analise a sequência abaixo e, usando o raciocínio lógico, marque a alternativa que indica o número que deve substituir o símbolo de interrogação "?" no último quadro.



- a) 12 b) 11 c) 10 d) 9 e) 8

11 - Analise a sequência abaixo e marque a alternativa que indica a figura que deve ser colocada no lugar do símbolo de interrogação "?" de forma que a lógica se mantenha.



- a) A b) B c) C d) D e) E

DIFERENTES = TRUE
IGUAIS = FALSE

12 - Qual o valor será exibido na terceira iteração do código Swift abaixo?

```
01 let array = [1,2,3,4,5,6,7,8,9,10]
02 func funcao(_ n: Int) -> Bool {
03     return n % 2 == 0
04 }
05 for num in array where funcao(num) {
06     print(num)
07 }
```

a) 1

b) 3

c) 6

d) 8

e) 10

13 - Considere a função em C abaixo.

```
01 int func(int v[], int x, int y) {
02     if (x < y) {
03         if (v[x] == 0) {
04             return 1 + func(v, x+1, y);
05         }else {
06             return func(v, x+1, y);
07         }
08     }else {
09         if (x > y) {
10             return func(v, y, x);
11         }else {
12             if (x == y) {
13                 if (v[x] == 0) {
14                     return 1;
15                 }
16             }
17             return 0;
18         }
19     }
20 }
```

Com relação ao algoritmo da função acima, assinale a opção correta:

- a) A função retorna o número de zeros (0) no intervalo [x,y] do vetor v.
- b) A chamada da função func(vet, 8, 1) resulta numa execução infinita, qualquer que seja o valor de vet passado por parâmetro.
- c) A função retorna um número de zeros (0) no intervalo [x,y-1] do vetor v.
- d) Considerando as características da função, não é possível implementá-la sem o uso de recursividade.
- e) A garantia de que a função recursiva não será infinitamente executada está implementada exclusivamente na linha 17.

14 - Após a execução do código abaixo, qual será a sequência de números impressa?

```

01 class MyInt {
02     public static int num;
03 }
04
05 class MyDouble {
06     double num;
07     public MyDouble(double num) {
08         this.num = num;
09     }
10 }
11
12 public class Mist {
13     void mix (int x, Integer y, MyInt z, MyDouble w) {
14         x = x+1;
15         y = x+2;
16         z.num = y+3;
17         w.num = z.num+4;
18     }
19
20     public static void main (String args[]) {
21         int a = 1;
22         Integer b = 2;
23         MyInt c = new MyInt(); MyInt.num = 3;
24         MyDouble d = new MyDouble (4.0);
25
26         new Mist().mix(a,b,c,d);
27
28         System.out.println (a+", "+b+", "+c.num+", "+d.num);
29     }
30 }

```

Handwritten notes on the code: Above line 13, 'A' is written above 'x' and 'B' above 'y'. Above line 14, 'A' is written above 'x = x+1;'. Above line 15, 'B' is written above 'y = x+2;'. Above line 16, 'z.num = y+3;' is underlined. Above line 17, 'w.num = z.num+4;' is underlined. To the right of the code, a handwritten sequence of numbers is written: 3, 18, 7, 4, 14, 17, 20, 16, 5. A large handwritten '30' is written below the code.

- a) 1, 2, 7, 11.0
- b) 2, 4, 7, 11.0
- c) 1, 2, 3, 4.0
- d) 2, 4, 3, 4.0
- e) 1, 2, 7, 4.0

15 - Para percorrer 250 km, um carro A gasta 60 litros de gasolina. Supondo o uso de um carro B que tem um consumo de gasolina 4 vezes menor, a quantidade necessária de litros de gasolina para percorrer 500 km é:

- a) 10 litros.
- b) 15 litros.
- c) 30 litros.
- d) 45 litros.
- e) 60 litros.

16 - Dado que os números 14, 4, 7, 18, 3, 5, 16, 20 e 17 foram organizados nessa sequência em uma árvore binária de busca. Qual a nova sequência de números quando a referida árvore for percorrida em ordem (também conhecida como ordem simétrica)?

- a) 14, 4, 3, 7, 5, 18, 16, 17, 20
- b) 3, 5, 7, 4, 17, 16, 20, 18, 14
- c) 3, 5, 7, 4, 18, 17, 16, 20, 14
- d) 3, 4, 5, 7, 14, 16, 17, 18, 20
- e) 3, 5, 7, 18, 17, 16, 20, 4, 14

17 - Marque a alternativa correta com base no código em linguagem C apresentado abaixo.

Após a execução do programa a saída é:

```
01  #include <stdio.h>
02  #include <stdlib.h>
03
04  int main(void) {
05      int vetor[] = {1, 1, 2, 3, 5, 8, 13, 21};
06      int *p = vetor;
07      int *p1 = NULL;
08      int *p2 = NULL;
09
10      p1 = ++p;
11      p2 = --p;
12
13      printf("%d %d %d %d \n", vetor[4], *(p+2), *(++p1), *p2);
14
15      return 0;
16  }
```

Handwritten notes: p[2] above line 13; 5 2 3 below line 13; 5 2 2 1 below line 13; a large checkmark below line 13.

a) 5 2 2 1

b) 2 2 2 1

c) 2 5 2 1

d) 5 3 2 1

e) 5 3 3 1

18 - Marque a opção correta com relação a linguagem C.

- a) Na linguagem C ponteiros podem ser usados; porém, endereços não podem ser atribuídos diretamente às variáveis.
- b) A linguagem C possui desde elementos de linguagem de alto nível, tal como a capacidade de permitir a criação de tipos definidos pelo usuário até conceitos de baixo nível, tal como a capacidade de manipular bits.
- c) A linguagem C não permite acesso a posições de memória que excedam os limites de um vetor. Por exemplo, caso se defina `int vetor[10]`, não é possível acessar `vetor[11]`, pois será lançado uma exceção em tempo de execução.
- d) O tipo de dados definido pelo usuário denominado `union` é semelhante ao tipo `struct`, tendo apenas como diferença que o `union` é apropriado para ser manipulado com ponteiros, diferentemente do tipo `struct`.
- e) A linguagem C está em desuso e atualmente Java vem sendo utilizado como alternativa.

19 - Sejam as afirmativas sobre classes que adotam o padrão singletons:

- I - Devem restringir suas instâncias a no máximo um objeto.
- II - É possível instanciar seus objetos a partir de um construtor público.
- III - Podem possuir construtores públicos.
- IV - Mantêm um ponto de acesso global para algum recurso do sistema.
- V - Singletons devem manter seus construtores privados.

Estão corretas as afirmativas:

- a) I, II, IV
- b) I, IV, V
- c) II, III, IV
- d) I, II, V
- e) I, II, IV, V

20 - Selecione a alternativa correta para a saída do programa em linguagem C apresentado a seguir.

```

01  #include <stdio.h>
02  #include <stdlib.h>
03  #include <string.h>
04
05  struct cliente {
06      int codigo; -
07      char nome[80]; -
08      char fone[15]; -
09  };
10
11  int main(void) {
12      struct cliente *cli1, cli2;
13      cli1 = malloc(sizeof(struct cliente));
14
15      cli2.codigo = 10; -
16      strcpy(cli2.nome, "joão"); -
17      strcpy(cli2.fone, "3333-4444"); -
18
19      cli1->codigo = ++cli2.codigo; - 11
20      strcpy(cli1->nome, "maria"); -
21      strcpy(cli1->fone, cli2.fone);
22
23      printf("%d %s %s %d %s %s \n", cli1->codigo, cli1->nome,
24      cli1->fone, cli2.codigo, cli2.nome, cli2.fone);
25      free(cli1);
26
27      return 0;
28  }
29

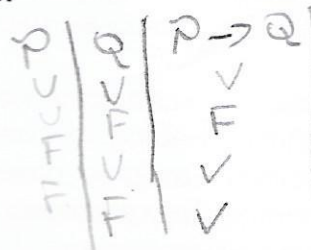
```

Handwritten notes on the code: "11" above line 19, "MARIA" above line 23, "3333-4444" above line 24, and "JOÃO 3333-4444" above line 25.

- a) Esse programa não compila, então um executável não é gerado e, portanto, não há como saber a saída do mesmo.
- b) A saída do programa é: 10 maria 3333-4444 10 joão 3333-4444 X
- c) A saída do programa é: 11 maria 3333-4444 10 joão 3333-4444
- d) A saída do programa é: 11 maria 3333-4444 11 joão 3333-4444
- e) A saída do programa é: 11 maria 3333-4444 11 maria 3333-4444

21 - Se Sandra é carioca, então Renata é cearense ou piauiense. Sabendo que Renata não é cearense e nem piauiense, logo pode-se afirmar que:

- a) Renata é carioca.
- b) Renata não é carioca.
- c) Sandra é cearense.
- d) Sandra é piauiense.
- e) Sandra não é carioca.



22 - Considerando o código abaixo marque a alternativa correta em relação as diferentes execuções do método show().


```
01 public abstract class Figura {
02     int b, h;
03     String tipoFigura = "Figura Abstrata";
04     public abstract int calcArea();
05
06     public void show() {
07         System.out.print(tipoFigura);
08         System.out.print(", base="+b+", altura="+h);
09         System.out.print(", Area="+calcArea()+"\n");
10     }
11
12     public static void main (String [] args) {
13         Figura [] vet = new Figura [3];
14         vet [0] = new Retangulo (3,4);
15         vet [1] = new Triangulo (2,2);
16         vet [2] = new Retangulo (4,3);
17         vet [0].show();
18         vet [1].show();
19         vet [2].show();
20     }
21
22 class Retangulo extends Figura {
23     String tipoFigura;
24     public Retangulo (int b, int h) {
25         this.b = b; this.h = h;
26         this.tipoFigura = "Retangulo";
27     }
28
29     public int calcArea() {
30         return b*h;
31     }
32
33 class Triangulo extends Figura {
34     String tipoFigura;
35     public Triangulo (int b, int h) {
36         this.b = b; this.h = h;
37         super.tipoFigura = "Triangulo";
38     }
39
40     public int calcArea() {
41         return b*h/2;
42     }
43 }
```

Handwritten notes:
B * A
RETANG 3, 4, 12
TRIANG 2, 2, 2

- a) vet[0].show() imprimirá: Retangulo, base=3, altura=4, Area=12 -
- b) vet[1].show() imprimirá: Triangulo, base=2, altura=2, Area=2 -
- c) vet[2].show() imprimirá: Figura Abstrata, base=3, altura=4, Area=12
- d) As alternativas a e b estão corretas.
- e) As alternativas b e c estão corretas.

23 - Marque a alternativa correta com base no código Java abaixo, sabendo que a classe PessoaFisica herda da classe Pessoa.

```
01 public class Pessoa {
02     private String nome;
03     protected String fone;
04
05
06     public String getNome() {
07         return nome;
08     }
09
10     public void setNome(String nome) {
11         this.nome = nome;
12     }
13
14     public String getFone() {
15         return fone;
16     }
17
18     public void setFone(String fone) {
19         this.fone = fone;
20     }
21 }
22
23 public class PessoaFisica EXTENDS Pessoa {
24     private String CPF;
25
26     public PessoaFisica(String nome, String fone, String CPF) {
27         this.setNome(nome);
28         this.fone = fone;
29         this.CPF = CPF;
30     }
31
32     public String getCPF() {
33         return CPF;
34     }
35
36     public void setCPF(String CPF) {
37         CPF = CPF;
38     }
39 }
```



- a) Para que o código esteja correto e compile, deve ser adicionada a palavra reservada ~~implements~~ entre PessoaFisica e Pessoa. ✗
- b) Para que o código esteja correto e compile, deve ser adicionada a palavra reservada extends entre PessoaFisica e Pessoa. Além disso, a palavra-chave protected no atributo fone deve ser substituída por private.
- c) Para que o código esteja correto e compile, deve ser adicionada a palavra reservada extends entre PessoaFisica e Pessoa. ✓
- d) Mesmo com a inclusão da palavra-chave adequada para indicar herança, o código não compila; pois a palavra-chave protected não está sendo usada corretamente.
- e) Desde que seja incluída corretamente a palavra-chave para indicar herança, pode-se substituir a linha de código this.setNome(nome); por this.nome = nome;.

24 - Marque a saída apresentada pelo programa em Java abaixo.

```
01 public class Cliente {
02     private String nome; -
03     private String fone; -
04
05     public Cliente(String nome, String fone) {
06         this.nome = nome;
07         this.fone = fone;
08     }
09
10     public String getNome() {
11         return nome;
12     }
13
14     public void setNome(String nome) {
15         this.nome = nome;
16     }
17
18     public String getFone() {
19         return fone;
20     }
21
22     public void setFone(String fone) {
23         this.fone = fone;
24     }
25
26     public String toString() {
27         return "[" + nome + ", " + fone + "]";
28     }
29 }
30
31 public class App {
32
33     public static void main(String[] args) {
34
35         Cliente c1 = new Cliente("daniel", "3333-4444");
36         Cliente c2 = new Cliente("alice", "3333-4444");
37         Cliente c3 = c1;
38         c3.setFone("4444-5555");
39         Cliente c4 = c2;
40         c4 = new Cliente("marina", "6666-7777");
41
42         System.out.printf("%s %s %s %s", c1, c2, c3, c4);
43     }
44 }
```

- a) [daniel , 3333-4444] [alice , 3333-4444] [daniel , 4444-5555] [marina , 6666-7777]
b) [daniel , 3333-4444] [marina , 6666-7777] [daniel , 3333-4444] [marina , 6666-7777]
c) [daniel , 4444-5555] [alice , 3333-4444] [daniel , 4444-5555] [marina , 6666-7777]
d) [daniel , 4444-5555] [marina , 6666-7777] [daniel , 4444-5555] [marina , 6666-7777]
e) [daniel , 4444-5555] [alice , 3333-4444] [daniel , 4444-5555] [alice , 3333-4444]