SERVIÇO PÚBLICO FEDERAL

MEC / SETEC

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SUL-RIO-GRANDENSE

DES Departamento de Seleção

CIDADE DE VENÂNCIO AIRES

INSTRUÇÕES GERAIS

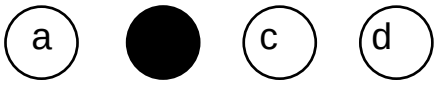
1 - Este caderno de prova é constituído por 40 (quarenta) questões objetivas.

2 - A prova terá duração máxima de 04 (quatro) horas.

3 - Para cada questão, são apresentadas 04 (quatro) alternativas (a – b – c – d). **APENAS UMA delas** responde de maneira correta ao enunciado.

4 - Após conferir os dados, contidos no campo Identificação do Candidato no Cartão de Resposta, assine no espaço indicado.

5 - Marque, com caneta esferográfica azul ou preta de ponta grossa, conforme exemplo abaixo, no Cartão de Resposta – único documento válido para correção eletrônica.



6 - Em hipótese alguma, haverá substituição do Cartão de Resposta.

7 - Não deixe nenhuma questão sem resposta.

8 - O preenchimento do Cartão de Resposta deverá ser feito dentro do tempo previsto para esta prova, ou seja, 04 (quatro) horas.

9 - Serão anuladas as questões que tiverem mais de uma alternativa marcada, emendas e/ou rasuras.

10 - O candidato só poderá retirar-se da sala de prova após transcorrida 01 (uma) hora do seu início.

BOA PROVA!

CONHECIMENTOS ESPECÍFICOS

As questões 1 e 2 referem-se ao código a seguir:

```
1   #include<stdio.h>
2
3   int main(int argc, char *argv) {
4           int a[5], i , j;
5
6           for (i = 0, j = 1; i < 5; i++, j = 5) {
7               a[i] = i + j;
8               j = i+1;
9           }
10
11          for (i = 0; i < 5; i++) {
12              printf("%d ", a[i]++);
13          }
14          printf("\n");
15
16          printf("%d, %d, %d", argc, argv[0], argv[1][0]);
17
18          return 0;
19  }
```

1. O código mostra um programa escrito na linguagem de programação C. Nessa figura, os números à esquerda representam cada linha de código fonte. Eles são meramente ilustrativos e não fazem parte do programa.

Ao executar o código, o valor impresso na saída padrão do usuário pelo comando *printf* da linha 12 será

a) 1 3 5 7 9
b) 1 6 7 8 9
c) 5 6 7 8 9
d) 2 7 8 9 10

2. Suponha que a chamada do programa correspondente ao código seja:
/program hello world

Que alternativa representa o valor impresso na saída padrão quando o código da linha 16 for executado?

a) 2, ./program, h
b) 2, hello, w
c) 3, ./program, h
d) 3, .hello, w

3. O código a seguir mostra um programa escrito na linguagem de programação C. Nesse código, os números à esquerda representam cada linha de código fonte. Eles são meramente ilustrativos e não fazem parte do programa.

```
1   #include<stdio.h>
2   void funcao1(int a, int b){
3           int temp;
4           temp=a;
5           a=b;
6           b=temp;
7   }
8
9   void funcao2(int *a, int *b){
10          int temp;
11          temp=*a;
12          *a=*b;
13          *b=temp;
14  }
15
16  int main(){
17          int a=2,b=3, c=4, d=5;
18          int *e = &b, *f = &d;
19          funcao1(a,b);
20          funcao2(&c,&d);
21          funcao2(e,f);
22          printf("%d, %d, %d, %d",a,b,c,d);
23          return 0;
24  }
```

Ao executar o código, o valor impresso na saída padrão do usuário (linha 22 do programa) será

a) 2, 3, 5, 4
b) 3, 2, 4, 5
c) 3, 2, 5, 4
d) 2, 4, 5, 3

4. O código mostra um programa escrito na linguagem de programação C. Neste código, os números à esquerda representam cada linha de código fonte. Eles são meramente ilustrativos e não fazem parte do programa. Suponha que o endereço da variável a na memória seja 1329207168 e que a quantidade de bytes alocada para o *array* a seja igual a 20.

```
1   #include<stdio.h>
2
3   int main() {
4           int a[5] = {0, 1, 2, 3, 4};
5           int *p = a;
6
7           printf("%d, %d, %d, %d", a[1], *p, *(p+1), sizeof(a)/sizeof(a[1]));
8
9           return 0;
10  }
```

Ao executar o código, o valor impresso na saída padrão do usuário (linha 7 do programa) será:

a) 0, 1329207168, 1329207168, 5
b) 1, 0, 1, 5
c) 1, 0, 1, 20
d) 1, 0, 1329207168, 1329207169, 20

5. O código a seguir mostra um programa escrito na linguagem de programação C. Os números à esquerda representam cada linha de código fonte. Eles são meramente ilustrativos e não fazem parte do programa.

```
1   #include <stdio.h>
2
3   main() {
4       int **p;
5       int *q, *r;
6       int a;
7
8       p = (int **)malloc(sizeof(int *));
9       *p = (int *)malloc(sizeof(int));
10
11      a = 10;
12      r = &a;
13      **p = 11;
14      q = *p;
15      *q = 12;
16      **p = a;
17      *r = 13;
18
19      printf("%d\n", *q);
20
21      free(q);
22      free(p);
23  }
```

Ao executar o código, o valor impresso na saída padrão do usuário (linha 19 do programa) será

a) 10
b) 11
c) 12
d) 13

6. Considere as seguintes afirmações sobre códigos válidos em XHTML, e assinale com (V) para as afirmativas verdadeiras e (F) para as falsas.

( ) Nomes de elementos em XHTML devem ser declarados em letras maiúsculas.
( ) Nomes de atributos em XHTML podem ser declarados em letras maiúsculas.
( ) Valores de atributos em XHTML devem ser declarados entre aspas duplas.
( ) Um documento XHTML deve conter obrigatoriamente uma declaração DOCTYPE.
( ) Um documento XHTML não precisa necessariamente ter um elemento do tipo <p>.

A sequência correta, de cima para baixo, é

a) F – F – V – V – V.
b) F – F – V – V – F.
c) V – V – V – V – V.
d) V – F – V – F – V.

7. Considere o trecho de código HTML a seguir:

```
1   <h2 id="elemento_na_pagina">
2  Algum texto
3  </h2>
4  Um link para <a page=”http://www.google.com/”> o página do Google</a>
5  <p>
6  <img src="/imagens/figura1.gif" alt="uma figura">
7  </p>
8  <p>
9  <a href="@elemento_na_pagina ">link para elemento na pagina</a>.
10 </p>
```

Indique a(s) linha(s) que contêm erro(s) de HTML:

a) 1 e 4
b) 4 e 9
c) 1, 4 e 9
d) 4 e 6

8. Com relação às tabelas em HTML, considere o seguinte texto:

O elemento _____ é usado para representar linhas na tabela, enquanto o elemento _____ é usado para representar células na tabela. Já o elemento _____ é usado para representar uma linha de cabeçalho com formatação especial. O código _____ faz com que cada célula da tabela seja separada de suas vizinhas a uma distância de 5 pixels, enquanto o código _____ faz com que cada célula da tabela seja expandida em 5 pixels em todas direções.

Que alternativa completa corretamente as lacunas do texto acima?

a) <tr>, <td>, <th>, cellpading=“5”, cellspacing=“5”
b) <td>, <tr>, <th>, cellspacing=“5”, cellpading=“5”
c) <td>, <th>, <tr>, cellpading=“5”, cellspacing=“5”
d) <tr>, <td>, <th>, cellspacing=“5”, cellpading=“5”

9. O código a seguir representa uma página simples em HTML com código CSS embarcado:

```
1  <html>
2     <head>
3        <style>
4           p { color: green;}
5           body > p { color: blue;}
6           div p { color: red;}
7        </style>
8     </head>
9
10    <body>
11       <p>Paragrafo 1</p>
12
13       <div>
14       <p>Paragrafo 2</p>
15       </div>
16
17       <p>Paragrafo 3</p>
18    </body>
19 </html>
```

As linhas 11, 14 e 17 contêm código para gerar respectivamente os parágrafos 1, 2 e 3. Considerando o código CSS embarcado nas linhas 4 a 6, que alternativa representa a cor dos parágrafos 1, 2 e 3, uma vez que a página seja visualizada em um navegador?

a) verde, vermelho, verde
b) azul, vermelho, azul
c) azul, verde, azul
d) verde, vermelho, verde

10. Com relação às linguagens HTML e CSS, considere que um programador deseja definir como preta a cor de um link no momento em que o mouse se encontra sobre o link.

Qual afirmativa contém o código correto?

a) a:link { color : #000000 }
b) a:visited { color : #000000 }
c) a:hover { color : #000000 }
d) a:active { color : #000000 }

11. O termo RSS é um subconjunto de "dialetos" XML que serve para agregar conteúdo. É bastante utilizado em *sites* de notícias e *blogs*. A abreviatura do RSS 2.0 é usada para se referir aos seguintes padrões

a) *Really Simple Syndication*.
b) *Really Synchonous Style.*
c) *Relation Server Socket.*
d) *Real Style Synchronous.*

12. O elemento principal na criação e definição do RSS é o uso de

a) *<start>*
b) *<item>*
c) *<open>*
d) *<rss>*

13. Sobre Banco de Dados, são feitas as seguintes afirmações:

I. Segundo SILVERSCHATZ (2006), um Sistema Gerenciador de Bancos de Dados (SGBD) é uma coleção de arquivos e programas inter-relacionados que permitem ao usuário o acesso a consultas e alterações de dados.

II. Nível lógico é o nível médio de abstração que descreve quais dados estão armazenados no banco de dados e quais os inter-relacionamentos entre eles.

III. Nível de visão é o nível mais baixo de abstração que descreve como os dados estão de fato armazenados.

Está (ão) correta(s) apenas a(s) afirmativa(s)

a) I.
b) II.
c) I e II.
d) II e III.

14. Que informação está **INCORRETA**?

a) Atributos podem ser representados pelas propriedades de entidades (tabelas). O atributo serve para caracterizar uma tabela.
b) Ocorrência é um conteúdo ou dado inserido em um atributo de uma entidade (uma linha dentro de uma tabela).
c) Relacionamento é uma ligação ou uma associação entre as entidades.
d) Ocorrências podem ser exemplificadas como: 1:1, 1:N , N:N.

15. Sobre Banco de Dados, são feitas as seguintes afirmações:

I. o ER ou DER é uma ferramenta de modelagem, muito utilizada em projetos de banco de dados.

II. um banco de dados representa alguns aspectos do mundo real, sendo chamado às vezes, de mini-mundo.

III. a implementação de um banco corresponde ao modelo lógico.

Estão corretas as afirmativas

a) I e III apenas.
b) I e II apenas.
c) II e III apenas.
d) I, II e III.

16. Marque a alternativa **INCORRETA**.

a) Cardinalidade mínima 1 = "associação obrigatória" e Cardinalidade mínima N = "associação opcional".
b) Generalização/Especialização: através desses conceitos é possível atribuir propriedades particulares a um subconjunto de ocorrências (especializadas) de uma entidade genérica.
c) Na transformação de um relacionamento com cardinalidade N:N, a cardinalidade da entidade criada (associativa) fica com cardinalidade reduzida (0,1) ou (1,1) em cada uma de suas relações.
d) Entidade associativa é a redefinição de um relacionamento, que passa a ser tratado como uma entidade.

17. Considere as propriedades das transações (propriedades ACID):

I. Atomicidade – cada transação deve ter uma visão constante do banco de dados, sem levar em consideração qualquer outra transação. Durabilidade - Todas as operações de transações são refletidas corretamente no banco de dados, ou nenhuma delas.

II. Consistência – depois da transação, o banco de dados precisa salvar os dados corretamente e protegê-lo da falta de energia ou outras ameaças. Atomicidade – todas as partes da transação devem ser completadas ou nenhuma delas será completada.

III. Isolamento – cada transação deve ter uma visão constante do banco de dados, sem levar em consideração qualquer outra transação. Durabilidade – depois que uma transação for completada com sucesso, as mudanças realizadas por ela no banco de dados persistem, mesmo que existam falhas no sistema.

Está(ão) correta(s) apenas a(s) afirmativa(s)

a) I.
b) II.
c) III.
d) I e II.

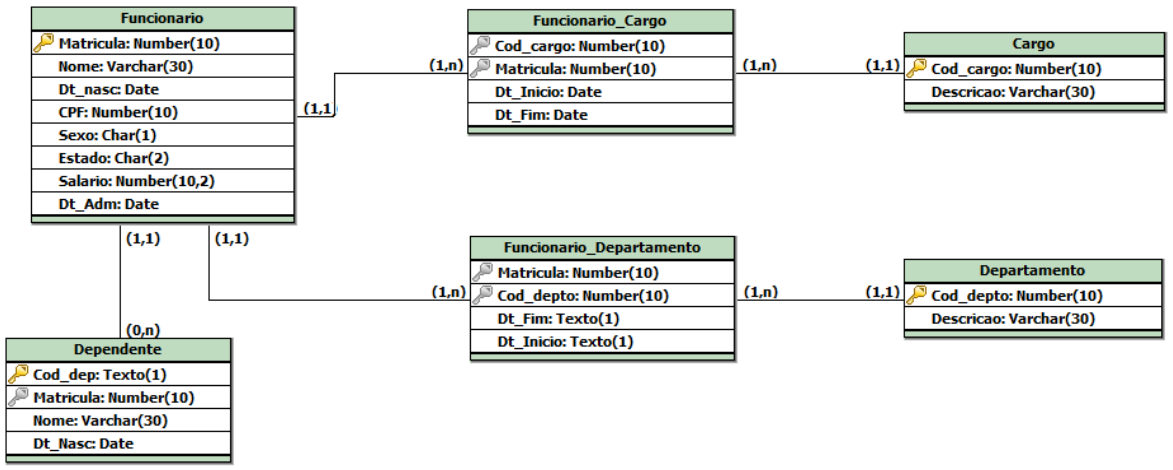
18. Sobre os estados da transação elencados, a sequência correta é:

a) ativa, parcialmente confirmada (em efetivação parcial), falha, confirmada (em efetivação) e abortada.
b) ativa, parcialmente confirmada (em efetivação parcial), falha, abortada.
c) ativa, falha, abortada, parcialmente confirmada (em efetivação parcial) e confirmada (em efetivação).
d) ativa, parcialmente confirmada (em efetivação parcial), falha, abortada e confirmada (em efetivação).

19. Para reverter uma das transações envolvidas no *deadlock*, qual o comando mais adequado?

a) Commit;
b) Rollback to savepoint;
c) Savepoint;
d) Rollback;

Levando em consideração o diagrama, resolva as questões 20, 21 e 22:



20. Que alternativa representa a instrução SQL que resolve a seguinte consulta no Banco de Dados:

"Exiba nome do funcionário e a descrição do cargo para todos os funcionários que tenham o cargo de 'Programador'."

a)
```
SELECT f.nome, c.descricao
FROM funcionario f, cargo c, funcionario_cargo fc
WHERE f.matricula=fc.matricula
AND fc.cod_cargo=c.cod_cargo
OR INITCAP(c.descricao)='Programador';
```

b)
```
SELECT f.nome, c.descricao
FROM funcionario f, cargo c, funcionario_cargo fc
WHERE f.matricula=fc.matricula
AND fc.cod_cargo=c.cod_cargo
AND UPPER(c.descricao)='Programador';
```

c)
```
SELECT f.nome, c.descricao
FROM funcionario f, cargo c, funcionario_cargo fc
WHERE f.matricula=fc.matricula
AND fc.cod_cargo=c.cod_cargo
AND INITCAP(c.descricao)='Programador';
```

d)
```
SELECT f.nome, c.descricao
FROM funcionario f, cargo c, funcionario_cargo fc
WHERE f.matricula=fc.matricula
AND fc.cod_cargo=c.cod_cargo
AND LOWER(c.descricao)='PROGRAMADOR';
```

21. Considerando o SQL abaixo, assinale (V) para as alternativas verdadeiras e (F) para as falsas.

```
SELECT nome, CPF, estado
FROM funcionario
WHERE nome LIKE '%A%'
AND estado NOT IN ('RS','SC')
AND (CPF LIKE '%1%'
OR CPF LIKE '%2%'
OR CPF LIKE '%3%')
ORDER BY estado DESC, nome ASC;
```

( ) A consulta exibirá nome, CPF e estado.
( ) A consulta retornará informações dos funcionários que têm seu nome iniciado pela letra A.
( ) A consulta mostrará os funcionários que possuem no CPF, simultaneamente, os números 1, 2 e 3.
( ) Os dados serão organizados em ordem descendente de nome e ascendente de estado.
( ) Não serão exibidos os dados dos funcionários do RS e de SC.

A sequência correta, de cima para baixo, é

a) V – F – F – V – V.
b) V – V – F – V – F.
c) F – F – V – V – V.
d) V – F – V – V – V.

22."Considerando somente os funcionários que trabalharam em mais de um departamento, mostre o nome e em quantos departamentos cada um deles trabalhou."

a) SELECT f.nome, COUNT(*)
FROM funcionarios f, departamentos d, funcionario_departamento fd
WHERE f.matricula = fd.matricula
AND fd.cod_departamento = d.cod_departamento
GROUP BY p.nome
HAVING COUNT(*)>=2;

b) SELECT f.nome
FROM funcionarios f, departamentos d, funcionario_departamento fd
WHERE f.matricula = fd.matricula
AND fd.cod_departamento = d.cod_departamento
GROUP BY f.nome
AND COUNT(*)>2;

c) SELECT f.nome, COUNT(*)
FROM funcionarios f, departamentos d, funcionario_departamento fd
WHERE f.matricula = d.cod_departamento
AND HAVING COUNT(*)>=2;
GROUP BY f.nome

d) SELECT f.nome
FROM funcionarios f, departamentos d, funcionario_departamento fd
WHERE f.matricula = fd.matricula
AND fd.cod_departamento = d.cod_departamento
GROUP BY f.nome
AND HAVING COUNT(*)>2;

23.Às vezes, o chamado *modelo cascata*, paradigma do ciclo de vida, requer uma abordagem sistemática, sequencial ao desenvolvimento do *software*, que se inicia no nível do sistema e avança ao longo da análise, do projeto, da codificação, do teste e da manutenção.
A partir disso, numere as colunas de cima para baixo.

1 Análise e engenharia de sistemas
2 Análise de requisitos de *software*
3 Projeto
4 Codificação
5 Testes
6 Manutenção

( ) É um processo de múltiplos passos que se concentra em quatro atributos distintos do programa: estrutura de dados, arquitetura de *software*, detalhes procedimentais e caracterização de interface. É documentado e torna-se parte da configuração do *software*.
( ) Reaplica cada uma das etapas precedentes do ciclo de vida a um programa existente e não a um novo programa.
( ) Inicia-se com o estabelecimento dos requisitos para todos os elementos do sistema e prossegue com a atribuição de certo subconjunto desses requisitos ao *software*.
( ) Seu processo de realização concentra-se nos aspectos lógicos internos do *software* e também nos aspectos funcionais externos, ou seja, descobre erros e garante que a entrada definida produza resultados reais que concordem com os resultados exigidos.
( ) É intensificado e concentrado especificamente no *software*.
( ) O projeto deve ser traduzido numa forma legível por máquina.

A sequência correta, de cima para baixo, é

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

24. A medição é comum no mundo da engenharia. Medimos o consumo de energia, o peso, as dimensões físicas, a temperatura, a voltagem..., a lista é quase interminável. Infelizmente, a medição está longe de ser um lugar-comum no mundo da engenharia de *software*. Temos dificuldade em concordar sobre o que medir e dificuldades para avaliar as medidas que são obtidas. O *software* é medido por muitas razões:

I. Indicar a qualidade do produto.

II. Avaliar a produtividade das pessoas que produzem o produto.

III. Avaliar os benefícios derivados de novos métodos e ferramentas de software.

IV. Formar uma linha básica para estimativas.

V. Ajudar a justificar os pedidos de novas ferramentas ou treinamento adicional.

Estão corretas as afirmativas

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

25. A engenharia de *software* é uma área para o desenvolvimento de *software* de alta qualidade para sistemas baseados em computador. Na engenharia de *software* existem quatro paradigmas – o ciclo de vida clássico, o uso de protótipos, o modelo espiral e as técnicas de quarta geração. Cada um dos paradigmas é distinto entre si; contudo, todos eles têm três fases em comum.

Como se caracteriza a Fase de Desenvolvimento?

a) Tão logo o teste de *software* esteja concluído, o *software* estará quase pronto para ser liberado aos usuários finais.
b) Traduz um conjunto de requisitos num elemento de sistema operacional que chamamos de *software*. A primeira etapa concentra-se no projeto.
c) Durante essa etapa, uma descrição limitada do escopo do esforço de *software* é desenvolvida; uma análise de riscos é realizada; os recursos exigidos para se desenvolver o *software* são previstos; estimativas de custo e de prazo são estabelecidas.
d) Fornece uma indicação preliminar da viabilidade do projeto em relação às restrições de custo e de prazo que possam já ter sido estabelecidas.

26. Para manipular interrupções e atender às restrições de tempo do sistema, muitos sistemas operacionais de tempo real fazem cálculos dinâmicos para determinar se as metas do sistema podem ser cumpridas.

Esses cálculos dinâmicos baseiam-se em:

I. Frequência média de ocorrência dos eventos.

II. Natureza dos eventos.

III. Quantidade de tempo necessária para atender os eventos.

Está(ão) correta(s) apenas a(s) afirmativa(s).

a) II.
b) III.
c) I e III.
d) II e III.

27. Sobre as linguagens de programação e a engenharia de *software*, é **INCORRETO** afirmar que

a) as linguagens oferecem os meios para a tradução ser humano/máquina; porém, a qualidade do resultado final está mais estreitamente ligada às atividades de engenharia de *software* que precedem e que se seguem à codificação.
b) o projeto de dados não pode ser influenciado pelas características da linguagem.
c) a qualidade de um projeto de *software* é estabelecida independentemente das características da linguagem de programação (uma notável exceção é o projeto orientado a objeto).
d) durante o planejamento do projeto, uma consideração das características técnicas de uma linguagem de programação raramente é levada a efeito.

28. Centenas de linguagens de programação têm sido usadas uma vez ou outra em esforços sérios de desenvolvimento de *software*. Quatro linguagens de programação são descritas – linguagens de primeira, segunda, terceira e quarta geração.

De acordo com o afirmativas, assinale (V) para as verdadeiras e (F) para as falsas.

( ) A primeira geração de linguagens remonta aos dias da codificação em nível de máquina.
( ) O código de máquina e seu equivalente mais legível por seres humanos - a linguagem *assembler* – representam a segunda geração de linguagens de programação.
( ) As linguagens de primeira geração foram desenvolvidas no final da década de 1950 e no começo da década de 1960 e servem de base para todas as linguagens de programação modernas (terceira geração).
( ) A linguagem ALGOL é a precursora de muitas linguagens de terceira geração e oferece um repertório extremamente rico de construções procedimentais e de tipologia de dados.
( ) As linguagens de quarta geração podem ser divididas em três amplas categorias: linguagens de alto nível de uso geral, linguagens de alto nível orientadas a objeto e linguagens especializadas.
( ) As linguagens de quarta geração combinam características procedimentais e não-procedimentais. Ou seja, a linguagem possibilita que o usuário especifique condições e as correspondentes ações, encorajando, ao mesmo tempo, o usuário a indicar o resultado desejado e então aplicar seu conhecimento específico do domínio para preencher os detalhes procedimentais.

A sequência correta, de cima para baixo, é

a) V – F – F – V – F – V.
b) V – V – F – V – F – V.
c) F – F – F – V – F – F.
d) F – V – F – F – V – V.

29. Sobre a qualidade de *software* e garantia de qualidade, podemos afirmar que os _____ de *software* são a base, a partir da qual a qualidade é medida, os _____ definem um conjunto de critérios de desenvolvimento que orientam a maneira segundo a qual o *software* passa pelo trabalho de engenharia e geralmente existem os _____ que não são mencionados, sendo importante fator da qualidade do *software*.

Os termos que preenchem corretamente as lacunas são

a) padrões – requisitos – requisitos implícitos
b) padrões – requisitos implícitos – requisitos
c) requisitos – padrões – requisitos implícitos
d) requisitos implícitos – padrões – requisitos

30. A manutenibilidade pode ser definida qualitativamente como a facilidade com que um *software* pode ser entendido, corrigido, adaptado e/ou aumentado. Existem diversas medidas quantitativas que podem ser aplicadas para medir a manutenibilidade de *software*. Existem diversas métricas da manutenibilidade que se relacionam ao esforço despendido durante a manutenção que são citadas a seguir, **EXCETO**,

a) tempo de reconhecimento do problema, tempo de revisão de manutenção.
b) tempo de testes globais, tempo de retardo administrativo.
c) tempo de análise do problema, tempo de recuperação parcial.
d) tempo de recuperação total, tempo de correção ativa.

31. O código a seguir mostra um trecho de um programa escrito na linguagem de programação Java e que contém uma estrutura de seleção do tipo *switch*. Os números à esquerda representam cada linha de código fonte. Eles são meramente ilustrativos e não fazem parte do programa.

```
1     Scanner entrada = new Scanner(System.in);
2     System.out.print("Informe o valor desejado: ");
3     int valor = entrada.nextInt();
4     switch (valor) {
5       case 1:
6           System.out.print(" 1 ");
7       case 2:
8           System.out.print(" 2 ");
9       case 3:
10           System.out.print(" 3 ");
11      case 4:
12           System.out.print(" 4 ");
13      case 5:
14           System.out.print(" 5 ");
15      default:
16           System.out.print(" ### ");
17      }
```

Em relação ao código apresentado, são feitas as seguintes afirmações::

I. Será impressa na saída padrão do usuário a mensagem **4**, caso o usuário execute o código e informe o valor 4 na linha 3 do programa.

II. A mensagem **###** (contida na cláusula *default*) será impressa na saída padrão do usuário para qualquer valor informado na linha 3 do programa.

III. Será impressa na saída padrão do usuário a mensagem **5 ###**, caso o usuário execute o código e informe o valor 5 na linha 3 do programa.

Está (ão) correta(s) apenas a (s) afirmativa(s)

a) I.
b) II.
c) III.
d) II e III.

32. Ao desenvolver sistemas utilizando a linguagem de programação Java é comum a utilização de modificadores de acesso *public*, *private* e *protected*.

Em relação a eles, o que é **INCORRETO** afirmar?

a) Atributos declarados como *protected* em uma superclasse podem ser acessados apenas por membros dessa superclasse e por membros de suas subclasses, não sendo acessíveis a outras classes no mesmo pacote.
b) O uso do modificador *public* é opcional, pois todos os atributos e métodos são considerados como públicos quando não é definido nenhum modificador de acesso no momento da declaração.
c) Atributos e métodos declarados como públicos em uma superclasse poderão ser acessados e utilizados livremente em qualquer uma de sua(s) subclasse(s).
d) Os atributos e métodos declarados como *private* são acessíveis apenas aos métodos da classe em que são declarados.

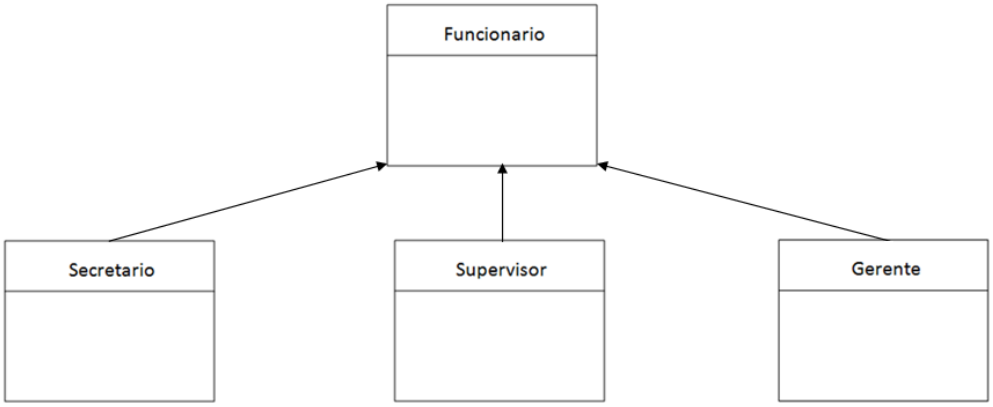
33. O código apresenta um trecho escrito na linguagem de programação Java e contém uma possível modelagem para a classe Pessoa.

```
1    public class Pessoa {
2        private String nome;
3        private int idade;
4        private float altura;
5
6        public Pessoa(String nome) {
7            this.nome = nome;
8        }
9        public Pessoa(int idade) {
10           this.idade = idade;
11       }
12       public Pessoa(float altura) {
13           this.altura = altura;
14       }
15       public Pessoa(String nome, int idade, float altura) {
16           this.nome = nome;
17           this.idade = idade;
18           this.altura = altura;
19       }
20   }
```

O código representa

a) herança.
b) encapsulamento.
c) sobrecarga de construtores.
d) polimorfismo.

34. Considere o diagrama de classes que será utilizado para a implementação de uma determinada aplicação.



A aplicação deverá permitir que sejam instanciados apenas objetos das classes Secretario, Supervisor e Gerente.

```
1  public _______ class Funcionario {
2
3  }
```

Utilizando a linguagem de programação Java e parte da declaração demostrada acima, qual é a palavra-chave correta para preenchimento da lacuna?

a) interface
b) abstract
c) final
d) static

35. O código mostra um trecho de um programa escrito na linguagem de programação Java e que contém uma estrutura de tratamento de exceções. Nesse código, os números à esquerda representam cada linha de código fonte. Eles são meramente ilustrativos e não fazem parte do programa.

```
1       Scanner entrada = new Scanner(System.in);
2       try {
3           System.out.println("Informe um numerador inteiro: ");
4           int numerador = entrada.nextInt();
5           System.out.println("Informe um denominador inteiro");
6           int denominador = entrada.nextInt();
7           float resultado = numerador/denominador;
8           System.out.println("Resultado = " + resultado);
9       } catch (ArithmeticException aritExcp) {
10          System.err.printf("Exceção: %s", aritExcp);
11      } catch (InputMismatchException inptMism) {
12          System.err.printf("Exceção: %s", inptMism);
13      }
```

Com relação ao código apresentado, é correto afirmar?

a) O código contido nas linhas 4 e 6 poderá lançar uma exceção do tipo InputMismatchException, e caso isso ocorra, ela será capturada e tratada nas linhas 11 e 12 do código.
b) O bloco de código contido no bloco try poderá lançar alguma exceção, e caso isso ocorra, sempre será executado tanto o catch referente ao ArithmeticException (linhas 9 e 10) e o catch referente ao InputMismatchException (linhas 11 e 12).
c) O bloco try/catch do programa será capaz de tratar apenas exceções relacionadas à divisão por zero que poderá ocorrer caso o usuário informe um valor nulo na linha de código 6.
d) O código contido na linha 7 poderá lançar exceções do tipo InputMismatchException e ArithmeticException, e caso isso ocorra, elas serão capturadas e tratadas nas linhas 9 a 13 do código.

36. Em relação ao tratamento de exceções na Linguagem de Programação Java, são feitas as seguintes afirmações:

I. Uma exceção não capturada (sem um bloco *catch* correspondente) sempre interromperá a execução do programa, mesmo quando implementado um programa *Multithreading* e a exceção tenha ocorrido em uma determinada *Thread*.

II. Ao implementar um bloco *try-catch*, é possível utilizar um bloco *finally*. Como esse bloco será executado tendo ocorrido ou não uma exceção, é comum sua utilização para que seja realizada a liberação de recursos.

III. Java permite capturar exceções e, além disso, permite que o desenvolvedor possa criar e lançar exceções. Para isso, deve ser criada uma nova exceção e lançá-la utilizando a instrução *throw***.**

Estão corretas as afirmativas

a) I e II apenas.
b) I e III apenas.
c) II e III apenas.
d) I, II e III.

37. Em relação à interface de classes na linguagem de programação Java, são feitas as seguintes afirmações:

I. A declaração de uma interface é feita através do uso da palavra-chave *interface* juntamente com o nome que será dado a ela, sendo que a interface irá conter a declaração de métodos que serão sempre *public* e *abstract*.

II. As classes que desejarem implementar uma determinada interface deverão ser declaradas com o acréscimo da palavra chave *implements* e poderão ou não implementar os métodos definidos na interface.

III. Ao ser criada uma determinada interface, podemos definir métodos. Além disso, é possível que as interfaces também contenham campos que serão implicitamente *final* e *static*.

Estão corretas as afirmativas

a) I e II apenas.
b) I e III apenas.
c) II e III apenas.
d) I, II e III.

38. Leia as sentenças abaixo, das quais foram excluídas algumas palavras.

_____ envolve a utilização de uma variável de superclasse para invocar métodos nos objetos de superclasse e de subclasse.

_____ é uma forma de reutilização de software em que novas classes adquirem os membros de classes existentes e aprimoram essas classes com novas capacidades.

Os construtores de subclasse podem chamar construtores de superclasse via palavra-chave *super.* No entanto, essa chamada deverá ser a _____ linha de comando no construtor da subclasse.

As palavras que completam, correta e respectivamente as lacunas acima, são:

a) Polimorfismo, Herança, primeira
b) Herança, Polimorfismo, última
c) Polimorfismo, Herança, última
d) Herança, Polimorfismo, primeira

39. Em relação a comunicações baseadas em *Socket* na linguagem de programação Java, são feitas as seguintes afirmações

I. Caso aconteça algum erro de entrada/saída ao fechar o socket, uma exceção do tipo *SocketException* irá ocorrer.

II. Um objeto da classe *InetAddress* contém um endereço IP.

III. A classe *DatagramSocket* vincula o aplicativo a uma porta para transmissão de datagrama.

Estão corretas as afirmativas

a) I e II apenas.
b) I e III apenas.
c) II e III apenas.
d) I, II e III.

40. Em relação à programação de *threads* utilizando a linguagem de programação Java, leia a sentença a seguir e preencha as lacunas com a opção correta.

A forma preferida de criar aplicativos Java de múltiplas threads é implementando a interface _____. Essa interface define e exige que seja implementado um método _____ único, que contém o código que definirá o que a thread irá realizar.

a) Runnable, start
b) Serializable, run
c) Serializable, start
d) Runnable, run