Leia o trecho a seguir:
“Um dos principais problemas enfrentados na fase de levantamento de requisitos é o de comunicação. A comunicação constitui-se em um dos maiores desafios da engenharia de software, caracterizando-se pela dificuldade em conseguir compreender um conjunto de conceitos vagos, abstratos e difusos que representam as necessidades de os desejos dos clientes e transformá-los em conceitos concretos e inteligíveis.”Fonte: GUEDES, G. T. A. UML 2: uma abordagem prática. 2 ed. São Paulo: Novatec Editora, 2011. p. 22.
Considerando essas informações e o conteúdo estudado, pode-se afirmar que uma das técnicas mais importantes para obter requisitos é:

D- o de entrevistas, em que o analista deve auxiliar na comunicação com o cliente, definindo as informações que devem ser produzidas, quais devem ser fornecidas e qual o nível de desempenho exigido do software.

Leia o trecho a seguir:
“A capacidade de gerenciamento da indústria da construção civil permite uma razoável precisão na data de entrega das obras, graças à padronização de processos de construção e a uma intensa padronização de componentes. Com exceção talvez apenas da alvenaria, uma edificação é composta de partes já construídas e que são integradas para formar o produto final”. Fonte: DEBONI, J. E. Z. Modelagem orientada a objetos com a UML. São Paulo: Futura, 2003. p. 10.
Considerando essas informações e o conteúdo estudado sobre a clássica comparação da modelagem de software com as engenharias, pode-se afirmar que os modelos de software podem ser comparados em uma construção civil:

B- com as plantas, que contêm uma representação gráfica do produto final: o edifício, permitindo que o cliente avalie o produto e acompanhe seu desenvolvimento.

“Ao fazer a modelagem da arquitetura do sistema, você captura decisões sobre os requisitos do sistema, seus elementos lógicos e seus elementos físicos. Você também fará a modelagem tanto de aspectos estruturais como comportamentais dos sistemas e dos padrões que dão forma a essas visões. Por fim, você desejará focalizar as costuras existentes entre os subsistemas e fazer o acompanhamento desde os requisitos até a entrega.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML, Guia do Usuário. 2. ed. Rio de Janeiro: Editora Campus, 2005. p. 629.
Considerando essas informações e o conteúdo estudado sobre as cinco visões complementares utilizadas na modelagem, analise-as a seguir e associe-as com as suas respectivas funções.
1) Visão de casos de uso.
2) Visão de projeto.
3) Visão de processo.
4) Visão de implementação.
5) Visão de implantação.
( ) Detalha características da distribuição física de um software e seus componentes e conexões.
( ) Modela a distribuição dos processos e das atividades concorrentes do software.
( ) Captura o vocabulário do problema a ser resolvido.
( ) Expõe os requisitos do sistema usando atores e suas ações.

( ) Expõe questões técnicas de engenharia dos componentes do software.
Agora, assinale a alternativa que apresenta a sequência correta:
B- 5, 3, 2, 1, 4.

“A tecnologia sozinha não resolve os problemas, o esforço solitário fica isolado, se não for criativo. O que une a tecnologia com a criatividade e direciona o trabalho é uma ideia comum – uma visão representada em um modelo. Estudando-se as etapas para transforma uma ideia em um produto de software, verifica-se a importância da criação de um modelo.”Fonte: DEBONI, J. E. Z. Modelagem orientada a objetos com a UML. São Paulo: Futura, 2003. p. 14.
Considerando essas informações e o conteúdo estudado sobre benefícios da modelagem, analise as afirmativas a seguir.
I. A modelagem de software reduz as incertezas do produto por registrar, antecipadamente ao desenvolvimento, detalhes que vão reger a implementação.
II. A concepção da modelagem auxilia no processo de desenvolvimento pós-orientação a objetos, chamado de contemporâneo pela dinamicidade de componentes.
III. A padronização e a automação dos projetos conseguida com a modelagem de software aumenta a maturidade nos processos de planejamento e controle nas equipes de desenvolvimento.
IV. A modelagem de software diminui a necessidade de aprendizado da equipe por focar em uma linguagem de programação específica, utilizada mundialmente.
Está correto apenas o que se afirma em:

E- I e III.

“Para entregar um software que satisfaça ao propósito pretendido, será preciso reunir-se e interagir com os usuários de uma maneira disciplinada, com a finalidade de expor os requisitos reais do sistema. Para desenvolver software de qualidade duradoura, será necessário criar uma arquitetura de fundação sólida que aceite modificações.”
Fonte: BOOCH, G., RUMBAUGH, J., JACOBSON, I. UML, Guia do Usuário. 2 ed. Rio de Janeiro: Editora Campus, 2005. p. 31.
Considerando essas informações e o conteúdo estudado sobre tipos de requisitos, analise os requisito a seguir e associe-os com suas respectivas características.
1) Requisitos de segurança.
2) Requisitos de negócio.
3) Requisitos de confiabilidade.
4) Requisitos de aspectos legais.
5) Requisitos de usabilidade.
( ) São específicos de cada software como, por exemplo, um cliente efetuando uma compra on-line em um site de e-commerce. São do tipo funcionais.
( ) Dizem respeito à proteção dos dados e das funções do software, sendo do tipo não funcionais.
( ) Referem-se à facilidade de utilização do software a partir de interfaces amigáveis, sendo do tipo não funcionais.
( ) Estão relacionados à probabilidade de operação livre de falhas de um software, em um ambiente e tempo especificados, sendo do tipo não funcionais.

( ) Definem restrições e condições de legislação do ambiente relacionadas ao campo de atuação do software, sendo do tipo normativos.
Agora, assinale a alternativa que apresenta a sequência correta:
B- 2, 1, 5, 3, 4.

“A maioria dos softwares profissionais é desenvolvida por equipes de projeto que variam em tamanho, desde duas até várias centenas de pessoas. Como é impossível para todos os membros de um grupo grande trabalharem juntos em um único problema, as grandes equipes são geralmente divididas em vários grupos. “Fonte: SOMMERVILLE, I. Engenharia de Software. 9 ed. São Paulo: Pearson Prentice Hall, 2011. p. 423 e 424.
Considerando essas informações e o conteúdo estudado sobre participantes do processo de análise e desenvolvimento de software, analise as funções a seguir e associe-as a seus respectivos detalhes de atuação.
1) Gerente de projeto.
2) Analista.
3) Projetista.
4) Programador.
5) Avaliador de qualidade.
( ) Realiza a codificação das estruturas definidas pelo projetista, realizando a implementação do software. Em alguns vocabulários, esse cargo também é conhecido como desenvolvedor.
( ) Analisa a adequação do processo de desenvolvimento e do produto de software aos padrões e requisitos estabelecidos no projeto.
( ) Define os requisitos do software a partir do conhecimento do domínio de negócio e da comunicação com especialistas. Ele faz a ponte de comunicação entre os profissionais da computação e os profissionais do negócio.
( ) Coordena as atividades de construção do software, incluindo a parte de orçamentação e de acompanhamento do cumprimento do cronograma de trabalho estabelecido.
( ) Integra a equipe de desenvolvimento avaliando alternativas de solução e gerando a especificação de uma solução computacional detalhada.
Agora, assinale a alternativa que apresenta a sequência correta:

A- 4, 5, 2, 1, 3.

“O modelo de casos de uso é importante, pois direciona diversas tarefas posteriores do processo de desenvolvimento de um sistema de software. Além disso, esse modelo força os desenvolvedores a moldarem o sistema de acordo com as necessidades do usuário.”Fonte: BEZERRA, E. Princípios de análise e projeto de sistemas com UML. 2 ed. Rio de Janeiro: Elsevier Editora, 2007. p. 54.
Considerando o trecho apresentado e o conceito estudado sobre casos de uso, analise as afirmativas a seguir.
I. A modelagem de casos de usos foi incorporada como técnica no processo de desenvolvimento de software, denominado *objectory*, no início da década de 1990.
II. O modelo de casos de uso é uma representação das funcionalidades internamente observáveis do sistema e da mecânica de funcionamento de cada componente do sistema.
III. Os casos de uso identificam elementos externos ao sistema que interage com ele.

IV. Um caso de uso pode ser iniciado por um elemento humano, modelado com notação própria, chamado de criador, do termo em inglês *creator*.
Está correto apenas o que se afirma em:

B- I e III.

“Muitos profissionais consideram um tanto complexo o conceito do paradigma de orientação a objetos. No entanto, esse conceito é apenas diferente do enfoque procedural ao qual estão acostumados. Na realidade, o ser humano, no início de sua infância, aprende e pensa de uma maneira orientada a objetos, representando seu conhecimento por meio de abstrações e classificações (na verdade, continuamos fazendo isso mesmo quando adultos, mas desenvolvemos outras técnicas que também utilizamos em paralelo)”.Fonte: GUEDES, G. T. A. UML 2: uma abordagem prática. 2 ed. São Paulo: Novatec Editora, 2011. p. 43.
Considerando essas informações e o conteúdo estudado sobre orientação a objetos, analise os princípios a seguir e associe-os com suas respectivas funções.
1) Abstração.
2) Encapsulamento.
3) Generalização.
4) Polimorfismo.
5) Composição.
( ) Ou ocultamento de informações, pode ser visto como algo que agrupa e protege algo de modo sistêmico.
( ) Permite que as pessoas possam concentrar a atenção nas características essenciais de um objeto.
( ) Em um método de uma classe filha, que tenha a mesma assinatura de um método da classe mãe, faz com que o primeiro prevaleça em relação à operação da mãe.
( ) Se refere a quando um objeto contém outros objetos. Como, por exemplo, um objeto carro que tem quatro objetos rodas.
( ) Ou herança, rege o relacionamento entre elementos gerais e elementos mais específicos.
Agora, assinale a alternativa que apresenta a sequência correta:

E- 2, 1, 4, 5, 3.

“Modelagem de sistema é o processo de desenvolvimento de modelos abstratos de um sistema, em que cada modelo apresenta uma visão ou perspectiva, diferente do sistema. A modelagem de sistema geralmente representa o sistema com algum tipo de notação gráfica.”Fonte: SOMMERVILLE, I. Engenharia de Software. 9 ed. São Paulo: Pearson Prentice Hall, 2011. p. 82.
Considerando essas informações e o conteúdo estudado sobre modelagem de software, pode-se afirmar que as notações:

B- representam o modo sistêmico de escrita de modelos, com a definição de elementos próprios, como ícones padrões em modelos gráficos.

“Somente quando os conceitos inerentes da aplicação são identificados, organizados e entendidos é que os detalhes das estruturas de dados e funções podem ser resolvidos de forma eficaz. O desenvolvimento OO é um processo conceitual independente de uma linguagem de programação até os estágios finais.”
Fonte: BLAHA, M. e Rumbaugh, J. Modelagem e projetos baseados em objetos. 2. ed. Rio de Janeiro: Elsevier Editora, 2006. Pág. 4.
Considerando essas informações e o conceito estudado sobre análise e projeto de software, analise as afirmativas a seguir.
I. Na fase de análise, é verificado o domínio da solução, com investigação e posterior codificação das estruturas que devem ser utilizadas no software a ser construído.
II. Na fase de projeto, é verificado o domínio do problema, com seus detalhes e características inerentes de requisitos e demandas que devem ser resolvidas no software.
III. Na fase de projeto, a chave da questão está na palavra “como”, sendo projetada e modelada a maior parte do software, incluindo seus componentes e sua arquitetura, que contempla a divisão de estruturas.
IV. É necessário verificar o domínio e contexto de necessidades reais de um software em determinado ambiente, devendo isso ser feito na fase de análise.
Está correto apenas o que se afirma em:

A- III e IV.

“[...] a modelagem de caso de uso é amplamente usada para apoiar a elicitação de requisitos. Um caso de uso pode ser tomado como um cenário simples que descreve o que o usuário espera de um sistema. Cada caso de uso representa uma tarefa discreta [...]”.
Fonte: Sommerville, I. Engenharia de Software. 9 ed. São Paulo: Pearson Prentice Hall, 2011. Pág. 86.
Considerando essas informações e o conceito estudado sobre casos de uso, analise as afirmativas a seguir.
I. Um caso de uso deve compreender um fluxo de eventos completo e consistente, com colaborações entre elementos.
II. O caso de uso apresenta relações de quatro componentes essenciais: elementos internos, elementos externos, um relógio para disparar eventos e uma trilha de conexão, para passagem de dados.
III. Devem ser feitos e modelados casos de uso irreais de estouro de limites previstos dadas memórias limitantes, chamados de casos de saturação, para que os projetistas possam programar situações futuras.
IV. O conjunto de casos de uso representa as situações possíveis de utilização do software modelado.
Está correto apenas o que se afirma em:
C- I e IV.

“A tecnologia OO enfatiza o que um objeto é, e não como ele é usado. Os usos de um objeto dependem dos detalhes da aplicação e normalmente mudam durante o desenvolvimento. À medida que os requisitos evoluem, os recursos fornecidos por um objeto são muito mais estáveis do que as formas como ele é usado [...].”Fonte: BLAHA, M. e RUMBAUGH, J. Modelagem e projetos baseados em objetos. 2 ed. Rio de Janeiro: Elsevier Editora, 2006. Pág. 8.
Considerando essas informações e o conteúdo estudado, pode-se afirmar que, na análise e projeto orientado a objeto, a definição de métodos:

E- detalha os comportamentos dos objetos e como será o acesso dos seus atributos, para preenchimento e obtenção dos mesmos.

“A programação deve ser direta, pois todas as decisões difíceis já devem ter sido tomadas. Durante a implementação, é importante seguir uma boa prática de engenharia de software, para que a rastreabilidade do projeto seja aparente e de modo que o sistema permaneça flexível e extensível.”Fonte: BLAHA, M. e RUMBAUGH, J. Modelagem e projetos baseados em objetos. 2 ed. Rio de Janeiro: Elsevier Editora, 2006. p. 5.
Considerando essas informações e o conteúdo estudado sobre a fase de projeto de desenvolvimento de software, analise as afirmativas a seguir.
I. Na fase de projeto, no paradigma orientado a objetos, a partir da identificação dos requisitos, são feitos detalhamentos técnicos das classes identificadas.
II. Na fase de projeto são descartados os modelos de análise, pela inviabilidade de implementá-los. Ou seja, eles foram importantes especialmente para o entendimento da problemática.
III. Na fase do projeto, começa a ficar importante, para a modelagem, a inclusão das caraterísticas algorítmicas internas das soluções.
IV. Mesmo considerando aspectos tecnológicos, a fase de projeto ainda deve conter a identificação de requisitos de negócio, com análise abstrata e unidirecional de associações de classes.
Está correto apenas o que se afirma em:

C- I e III.

“Em virtude de seu tamanho e sua complexidade, o desenvolvimento de sistemas de software é um empreendimento em equipe. Como o próprio nome diz, o gerente de projetos é o profissional responsável pela gerência ou coordenação das atividades necessárias à construção do sistema.”Fonte: BEZERRA, E. Princípios de análise e projeto de sistemas com UML. 2 ed. Rio de Janeiro: Elsevier Editora, 2007. p. 31.
Considerando essas informações e o conteúdo estudado, pode-se afirmar que o gerente de projetos em desenvolvimento de software realiza:

C- a gestão das atividades, da equipe e o orçamento do projeto, estimando o valor e tempo de cada atividade e acompanhando o cumprimento do cronograma estipulado.

“Um sistema de software orientado a objetos consiste em objetos em colaboração com o objetivo de realizar as funcionalidades desse sistema. Cada objeto é responsável por tarefas específicas. É graças à cooperação entre objetos que a computação do sistema se desenvolve.”Fonte: BEZERRA, E. Princípios de análise e projeto de sistemas com UML. 2. ed. Rio de Janeiro: Elsevier Editora, 2007. p. 7.
Considerando essas informações e o conteúdo estudado sobre análise e projeto orientado a objeto, analise as afirmativas a seguir.
I. O objetivo básico da Análise Orientada a Objetos é identificar classes a partir das quais objetos serão representados como instâncias e uma de suas tarefas é a identificação de atores.
II. Para cada classe devem ser anotadas, em modelos, as suas dependências, indicando o nível que cada uma estará na arquitetura, na escala bottom, middle e top.
III. No momento da definição de requisitos no texto obtido, verbos são potenciais candidatos a classes, e nomes (substantivos) são potenciais candidatos a métodos.
IV. Na especificação de atributos, será feita uma lista de todos os detalhes da classe, que serão as variáveis dos objetos.
Está correto apenas o que se afirma em:

A- I e IV.

“Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento. Esses requisitos refletem as necessidades dos clientes para um sistema que serve a uma finalidade determinada, como controlar um dispositivo, colocar um pedido ou encontrar informações.”Fonte: SOMMERVILLE, I. Engenharia de Software. 9 ed. São Paulo: Pearson Prentice Hall, 2011. p. 57.
Considerando essas informações e o conteúdo estudado sobre identificação de requisitos, analise as afirmativas a seguir.
I. O documento de requisitos pode ser usado como um instrumento de registro de consenso entre os desenvolvedores e o cliente.
II. O documento de requisitos, na maioria dos processos de desenvolvimento de software, é estático, pois alterar e incluir novas funções com a codificação ocorrendo é complexo e caro.
III. A definição dos requisitos precisa ser feita em detalhes precisos e sem ambiguidades, permitindo o bom desenvolvimento e, ainda, o trabalho adequado de testadores.
IV. Os requisitos são independentes por tratarem de funções específicas que o software deve conter, sendo isso relacionado ao conceito de abstração.
Está correto apenas o que se afirma em:

E- I e III.

“Na realidade, por mais simples que seja, todo e qualquer sistema deve ser modelado antes de se iniciar sua implementação, entre outras coisas, porque os sistemas de informação frequentemente costumam ter a propriedade de crescer: isto é, aumentar em tamanho, complexidade e abrangência.”
Fonte: GUEDES, G. T. A. UML 2: uma abordagem prática. 2 ed. São Paulo: Novatec Editora, 2011. p. 20.
Considerando essas informações e o conteúdo estudado, pode-se afirmar que, em relação aos diferentes tipos de modelagens no desenvolvimento de software:

D- existe uma categoria que se destaca nesse contexto, a dos modelos gráficos, que têm benefícios de facilidade de entendimento pelo ser humano e, ainda, podem ser acompanhados de textos.

“O modelo de casos de uso é importante, pois direciona diversas tarefas posteriores do processo de desenvolvimento de um sistema de software. Além disso, esse modelo força os desenvolvedores a moldarem o sistema de acordo com as necessidades do usuário.”Fonte: BEZERRA, E. Princípios de análise e projeto de sistemas com UML. 2 ed. Rio de Janeiro: Elsevier Editora, 2007. p. 54.
Considerando o trecho apresentado e o conceito estudado sobre casos de uso, analise as afirmativas a seguir.
I. A modelagem de casos de usos foi incorporada como técnica no processo de desenvolvimento de software, denominado *objectory*, no início da década de 1990.
II. O modelo de casos de uso é uma representação das funcionalidades internamente observáveis do sistema e da mecânica de funcionamento de cada componente do sistema.
III. Os casos de uso identificam elementos externos ao sistema que interage com ele.
IV. Um caso de uso pode ser iniciado por um elemento humano, modelado com notação própria, chamado de criador, do termo em inglês *creator*.
Está correto apenas o que se afirma em:

C- I e III.

“O processo de planejamento se inicia com uma avaliação das restrições (a data de entrega estabelecida, o pessoal disponível, o orçamento total, entre outras) que afetam o projeto. Essa avaliação é realizada em conjunto com uma estimativa dos parâmetros para o projeto, como sua estrutura, seu tamanho e sua distribuição de funções”.Fonte: SOMMERVILLE, I. Engenharia de Software. 6 ed. São Paulo: Pearson Prentice Hall, 2003. p. 63.
Considerando essas informações e o conteúdo estudado, pode-se afirmar que a modelagem de software:

A- é um meio de planejamento que se utiliza de modelos, normalmente gráficos, para a construção eficiente dos códigos das aplicações, sendo que o modelo é uma representação idealizada a ser construída.

“Um esquema é a especificação de um modelo usando uma determinada linguagem, a qual pode ser formal ou informal (por exemplo, linguagem natural), textual ou gráfica. Quando a representação do esquema é gráfica, designa-se usualmente por diagrama.”Fonte: DA SILVA, A. M. R. e VIDEIRA, C. A. E. UML, Metodologias e Ferramentas CASE. Lisboa: Editora Centro Atlântico. 2001. p. 34.
Considerando essas informações e o conteúdo estudado, analise as afirmações a seguir sobre a taxonomia de diagramas da UML e assinale V para a(s) verdadeira(s) e F para a(s) falsa(s).
I. ( ) O Diagrama de Perfil, ou Profile Diagram, está dentro da subcategoria de diagramas de interação, permitindo a definição de novos elementos UML e suas relações dinâmicas.
II. ( ) A subcategoria de diagramas chamada de Interação possui 4 tipos de diagramas que descrevem como grupo de elementos de modelos colaboram em um determinado comportamento.
III. ( ) A categoria de diagramas de Estrutura possui 7 diagramas que apresentam detalhes do sistema de modo estático, incluindo o Diagrama de Classes, representando as classes de objetos e suas ligações.
IV. ( ) O Diagrama de Casos de Uso é um dos mais conhecidos da UML, sendo da categoria de Comportamento e estando na subcategoria de Interação, por modelar os contatos interativos de atores com itens do sistema de modo externo.

Agora, assinale a alternativa que apresenta a sequência correta:

D- F, V, V, F.

“Centenas de ferramentas de engenharia de software de nível industrial são introduzidas a cada ano. A maioria é fornecida por empresas que afirmam que aquela ferramenta irá melhorar o gerenciamento de projeto, ou a análise de requisitos, ou a modelagem do projeto, [...].”Fonte: PRESSMAN, R. S. Engenharia de Software: Uma Abordagem Profissional. 7. ed. Porto Alegre: AMGH, 2011. p. 716.
Considerando o trecho apresentado e o conceito estudado sobre atuação e categorização das ferramentas CASE, analise as afirmativas a seguir.
I. As ferramentas CASE de desenho de bases de dados são específicas para a definição de componentes que tratam argumentos sob ação de algoritmos armazenados em classes orgânicas.
II. A categorização do tipo Vertical é relacionada ao tipo das ferramentas CASE e indica aquelas que, atuando em funções específicas nas fases de desenvolvimento de software, como a identificação de requisitos, por exemplo.
III. Ferramentas CASE podem atuar na fase de codificação, gerando partes de código com base em padrões pré-estabelecidos com fidelidade aos modelos de origem.
IV. Umas das funções possíveis e interessantes de Ferramentas CASE é o auxílio à comunicação da equipe, facilitando a troca organizada e profissional de mensagens entre membros.
Está correto apenas o que se afirma em:
B- III e IV.

“Para compreender a UML, você precisará formar um modelo conceitual da linguagem e isso pressupõe aprender três elementos principais: os blocos de construção básicos da UML, as regras que determinam como esses blocos poderão ser combinados e alguns mecanismos comuns aplicados na UML.”Fonte: BOOCH, G., RUMBAUGH, J., JACOBSON, I. UML, Guia do Usuário. 2. ed. Rio de Janeiro: Editora Campus, 2005. Versão digital. p. 52.
Considerando o conceito estudado sobre a arquitetura de modelos da UML, analise as afirmativas a seguir.
I. Em sua versão 2.5.1, a UML contém 16 tipos de diagramas, divididos em duas categorias: de estrutura, com 7 diagramas, e de comportamento, com 8 diagramas, e 1 transversal, que está nas duas categorias.
II. Em um diagrama de estrutura, pode ser apresentada a composição de classes de objetos e seus relacionamentos em um sistema.
III. A UML prevê a modelagem do software em camadas, com diferentes visões, sendo que alguns diagramas são mais gerais, com uma visão externa do sistema.
IV. Os diagramas de estrutura, pelo seu caráter de arquitetura flexível, podem mostrar detalhes do comportamento de um objeto ao longo do tempo.
Está correto apenas o que se afirma em:
C- II e III.

“[...] os diversos diagramas fornecidos pela UML permitem analisar o sistema em diferentes níveis, podendo enfocar a organização estrutural do sistema, o comportamento de um processo específico, a definição de um determinado algoritmo ou até mesmo as necessidades físicas para a implantação do sistema.”Fonte: GUEDES, G. T. A. UML 2: uma abordagem prática. 2. ed. São Paulo: Novatec Editora, 2011. p. 30.

Considerando essas informações e o conteúdo estudado, pode-se afirmar que um diagrama da categoria de estrutura pode:

B- incluir elementos que representam classes e seus relacionamentos pertencentes à lógica de negócio implementada no software.

“Tendo em conta a proliferação de aplicações nesta área, uma taxonomia das ferramentas CASE é particularmente importante, pois facilita a compreensão da abrangência de uma determinada ferramenta e da sua aplicabilidade nas fases e atividades do processo de desenvolvimento de software. Para além dessas questões, a classificação dessas ferramentas facilita ainda a realização de análises comparativas.”
Fonte: Da Silva, A. M. R. e Videira, C. A. E. UML, Metodologias e Ferramentas CASE. Lisboa: Editora Centro Atlântico. 2001. p. 261. Tradução nossa.
Considerando o trecho apresentado e o conceito estudado sobre classificação de ferramentas CASE, analise as afirmativas a seguir.
I. Upper-Case correspondem à categoria de ferramentas CASE utilizadas na fase de implementação (incluindo desenho técnico, de edição e compilação de código e de testes).
II. Lower-Case correspondem à categoria de ferramentas CASE especializadas na fase de concepção do software (ferramentas de análise e especificação e modelagem de requisitos).
III. Integrated CASE correspondem à categoria de ferramentas CASE que cobrem todo o ciclo de vida do desenvolvimento de software, desde a captura dos requisitos do sistema até o controle final da qualidade.
IV. A categorização das ferramentas CASE pode ser feita por diversos critérios, podendo incluir a análise das funcionalidades disponíveis.
Está correto apenas o que se afirma em:

C- III e IV

“A UML disponibiliza uma forma padrão de modelagem de projetos de sistemas, incluindo seus aspectos conceituais, tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, processos de banco de dados e componentes de software reutilizáveis.”
Fonte: PICHILIANI, M. C. Mapeamento de Software para permitir a colaboração síncrona. Dissertação de mestrado. 2006. (Mestrado em Engenharia Eletrônica e Computação) – Instituto Tecnológico de Aeronáutica, São José dos Campos, 2006. Disponível em: <http://www.comp.ita.br/~pichilia/argo/TeseVersaoFinal.pdf>. Acesso em: 08 nov. de 2019. p. 31.
Considerando essas informações e o conteúdo estudado sobre UML, pode-se afirmar que ela é:

E- uma linguagem com sintaxe bem definida para especificar, visualizar e documentar modelos de software no paradigma orientado a objetos, utilizando uma notação padrão.

“A UML abrange a documentação da arquitetura do sistema e de todos os seus detalhes. A UML também proporciona uma linguagem para a expressão de requisitos e para a realização de testes.”Fonte: BOOCH, G., RUMBAUGH, J., JACOBSON, I. UML, Guia do Usuário. 2. ed. Rio de Janeiro: Editora Campus, 2005. Versão digital. p. 52.

Considerando essas informações e o conteúdo estudado, pode-se afirmar que, para um analista modelar estaticamente a arquitetura de hardware de uma aplicação, ele deve:

A- utilizar um diagrama do tipo estrutural, podendo incluir nós como elementos físicos que representam um recurso computacional, geralmente com memória e, frequentemente, com capacidade de processamento

"Ferramentas CASE (Computer-Aided Software Engineering ou Engenharia de Software Auxiliada por Computador) são softwares que, de alguma maneira, colaboram para a execução de uma ou mais atividades realizadas durante o processo de engenharia de software."Fonte: GUEDES, G. T. A. UML 2: uma abordagem prática. 2. ed. São Paulo: Novatec Editora, 2011. p. 41.
Considerando o trecho apresentado e o conceito estudado sobre Bootstrap, analise as afirmativas a seguir.
I. Escolher uma ferramenta CASE é uma tarefa relativamente fácil e rápida, já que existem poucas opções no mercado com recursos de apoio ao desenvolvimento semelhantes.
II. A maioria das ferramentas CASE disponíveis no mercado provê suporte para a UML, já que essa linguagem proporciona regras de sintaxe clara adequadas para serem apoiadas por ferramentas de software.
III. Ferramentas convencionais de edição gráfica, como o Microsoft Paint, apoiam o desenho de modelos de modo similar ao das ferramentas CASE de modelagem de software, já que possibilitam diagramação de atores, casos de uso e classes com checagem de regras da UML.
IV. Além de auxiliar na diagramação de desenhos, as ferramentas CASE podem emitir relatórios, gerar códigos fonte e controlar de atividades de desenvolvimento.
Está correto apenas o que se afirma em:

C- II e IV.

"Os projetistas da UML não estavam preocupados com os detalhes semânticos da linguagem, mas com sua expressividade. Eles introduziram noções uteis, como diagramas de caso de uso, que ajudam com o projeto, mas são demasiadamente informais para apoiar a execução."Fonte: SOMMERVILLE, I. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011. p. 100.
Considerando o trecho apresentado e o conceito estudado sobre objetivos da UML, analise as afirmativas a seguir.
I. A UML visa encorajar o crescimento do mercado de ferramentas de software orientadas a objeto.
II. A UML objetiva ser um guia simplificado de desenvolvimento, para atrair cada vez mais desenvolvedores que desejam usá-la como um método conciso de desenvolvimento.
III. A UML tem como alvo estar unida a órgãos de checagem automática de código para a comercialização de selos de qualidade e aderência na linguagem.
IV. A UML mira em suportar conceitos de desenvolvimento de alto nível como componentes, colaboração, frameworks e padrões.
Está correto apenas o que se afirma em:
E- I e IV.

"A especificação da linguagem UML 2 é definida por meio da utilização de uma abordagem de metamodelagem que adapta técnicas de especificação formal. Embora essa abordagem necessite de um pouco do rigor de um método de especificação formal, ela oferece as

vantagens de ser mais intuitiva e pragmática.”Fonte: GUEDES, G. T. A. UML 2: uma abordagem prática. 2. ed. São Paulo: Novatec Editora, 2011. p. 466.
Considerando essas informações e o conteúdo estudado sobre especificação da UML a partir de sua versão 2.0, analise os itens de seus componentes a seguir e associe-os a suas respectivas funções:
1) Superestrutura.
2) Infraestrutura.
3) OCL (Object Constraint Language).
4) Relacionamentos.
5) UML Diagram Interchange.
( ) Define o metamodelo da UML com um núcleo de metalinguagem que pode ser reutilizado para definir outras arquiteturas de metamodelos, além de definir mecanismos de personalização e adaptação da UML.
( ) Define as construções da UML a nível de usuário, utilizadas para modelar a estrutura e o comportamento de um sistema.
( ) Ligam itens de modelos para formar relações como as de associação e herança.
( ) Junta informações gráficas com os arquivos XMI, que são um padrão da OMG para intercâmbio de informações. Seu uso mais comum é na persistência (gravação) e troca de metadados entre ferramentas de modelagem.
( ) Permite a escrita de regras e fórmulas para definir comportamentos e restrições em elementos dos modelos, incluindo semânticas próprias.
Agora, assinale a alternativa que apresenta a sequência correta:
D- 2, 1, 4, 5, 3.

“O comportamento de uma sociedade de objetos ou de uma operação individual poderá ser especificado por meio de uma interação. As interações envolvem outros elementos, inclusive mensagens, ações e ligações (as conexões entre os objetos).”Fonte: BOOCH, G., RUMBAUGH, J., JACOBSON, I. UML, Guia do Usuário. 2. ed. Rio de Janeiro: Editora Campus, 2005. Versão digital. p. 58.
Considerando o trecho apresentado e o conceito estudado sobre os diagramas de comportamento da UML, analise as afirmativas a seguir.
I. Os diagramas de comportamento mostram modelos que contêm o comportamento dinâmico dos objetos em um software, incluindo métodos, colaborações e histórico de estados.
II. Os diagramas de comportamento estão bem associados à linguagem de programação utilizada, indicando como o código-fonte e bibliotecas de componentes estão estruturados de modo estático.
III. O comportamento dinâmico de um software pode ser descrito como uma série de alterações no seu ambiente ao longo do tempo, incluindo a sequência de solicitações de serviço feitas pelos objetos.
IV. Nos diagramas de comportamento, tem-se foco na estrutura das classes utilizadas pelo sistema, identificando-as e determinando os atributos e métodos que cada classe tem.
Está correto apenas o que se afirma em:
E- I e III.

Leia o trecho a seguir:
“A Enterprise Architect é uma ferramenta gráfica multiusuário projetada para ajudar suas equipes a construir sistemas robustos e de boa manutenibilidade. E usando relatórios e documentação internos de alta qualidade, você pode oferecer uma visão verdadeiramente compartilhada dos desenvolvimentos com facilidade e precisão!”Fonte: ENTERPRISE

ARCHITECT. Full Lifecycle Modeling for Business, Software and Systems | Sparx Systems. Disponível em: <https://sparxsystems.com/products/ea/>. Acesso em: 05 nov. 2019.
Considerando essas informações e o conteúdo estudado, pode-se afirmar que a Enterprise Artchitect:

D- é uma tradicional ferramenta CASE paga para modelagem, design e gerenciamento de etapas do desenvolvimento de software baseada em UML, com suporte à sua versão 2.0 e padrões similares.

“Modelagem de sistema é o processo de desenvolvimento de modelos abstratos de um sistema, em que cada modelo apresenta uma visão ou perspectiva, diferente do sistema. A modelagem de sistema geralmente representa o sistema com algum tipo de notação gráfica, que, atualmente, quase sempre é baseada em notações de UML.”Fonte: SOMMERVILLE, I. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011. p. 82.
Considerando essas informações e o conteúdo estudado sobre UML, pode-se afirmar que a notação da UML:

B- possui uma vasta gama de símbolos gráficos para a representação de artefatos de software em diferentes diagramas. Para cada símbolo, há sintaxe e semântica bem definidas, permitindo a padronização de modelos.

“A UML é o trabalho de várias pessoas, e as ideias que ali se encontram vêm de muitos trabalhos anteriores. Seria um trabalho importante de pesquisa histórica reconstruir uma lista completa das fontes e ainda mais difícil identificar os muitos precursores que influenciaram a UML, de maneira mais ou menos relevante. Como em qualquer pesquisa científica e prática de engenharia, a UML é uma pequena colina sobre uma grande montanha de experiência anterior.” Fonte: BOOCH, G., RUMBAUGH, J., JACOBSON, I. UML, Guia do Usuário. 2. ed. Rio de Janeiro: Editora Campus, 2005. Versão digital. p. 15.
Considerando essas informações e o conteúdo estudado, analise as afirmações a seguir sobre a evolução da UML e assinale V para a(s) verdadeiras e F para a(s) falsa(s).
I. ( ) Para isentar a UML de estratégias empresariais, há, no OMG, a preferência em aceitar sugestões de melhorias provenientes de pesquisadores de universidades do que de pesquisadores ligados a empresas.
II. ( ) Dada a reconhecida qualidade e boa utilização no mercado, a linguagem UML foi adotada, em 1997, pelo OMG como uma linguagem padrão de modelagem de software feito no paradigma orientado a objetos.
III. ( ) O padrão de data para lançamento de novas versões da UML é em todo mês de setembro de cada ano, com cerimônias feitas em diferentes países. Essa melhoria feita de modo anual permite atualização constante da linguagem.
IV. ( ) Antes de existir, a UML era intensa a problemática de escolha de padrão de notação dentre várias técnicas existentes, que podiam possuíam notações distintas para modelar as mesmas perspectivas de um software.
Agora, assinale a alternativa que apresenta a sequência correta:

B- F, V, F, V.

“Centenas de ferramentas de engenharia de software de nível industrial são introduzidas a cada ano. A maioria é fornecida por empresas que afirmam que aquela ferramenta irá melhorar o gerenciamento de projeto, ou a análise de requisitos, ou a modelagem do projeto, [...].”Fonte: PRESSMAN, R. S. Engenharia de Software: Uma Abordagem Profissional. 7. ed. Porto Alegre: AMGH, 2011. p. 716.
Considerando o trecho apresentado e o conceito estudado sobre atuação e categorização das ferramentas CASE, analise as afirmativas a seguir.
I. As ferramentas CASE de desenho de bases de dados são específicas para a definição de componentes que tratam argumentos sob ação de algoritmos armazenados em classes orgânicas.
II. A categorização do tipo Vertical é relacionada ao tipo das ferramentas CASE e indica aquelas que, atuando em funções específicas nas fases de desenvolvimento de software, como a identificação de requisitos, por exemplo.
III. Ferramentas CASE podem atuar na fase de codificação, gerando partes de código com base em padrões pré-estabelecidos com fidelidade aos modelos de origem.
IV. Umas das funções possíveis e interessantes de Ferramentas CASE é o auxílio à comunicação da equipe, facilitando a troca organizada e profissional de mensagens entre membros.
Está correto apenas o que se afirma em:
A- III e IV.

“Os diagramas de classes do UML são uma integração de diferentes diagramas de classes existentes, nomeadamente no OMT, Booch e outros métodos OO. Extensões específicas de determinados processos (por exemplo: recorrendo a estereótipos e correspondentes ícones) podem ser definidos em vários diagramas para suportarem diferentes estilos de modelagem.” Fonte: Da Silva, A. M. R. e Videira, C. A. E. UML, Metodologias e Ferramentas CASE. Lisboa: Editora Centro Atlântico. 2001. p. 261. Tradução nossa.
Considerando essas informações e o conteúdo estudado sobre a história e evolução da UML, analise os autores a seguir e associe-os a suas respectivas contribuições para a especificação inicial da UML:
1) James Rumbaugh.
2) Ivar Jacobson.
3) David Harel.
4) Rebecca Wirfs-Brock.
5) Sally Shlaer e Stephen Mellor.
( ) Posição de gráficos.
( ) Ciclos de vida de objetos.
( ) Responsabilidades.
( ) Com a técnica OMT (Object Modeling Technique) e o método Objectory.
( ) Com o método OOSE (Object-Oriented Software Engineering).
Agora, assinale a alternativa que apresenta a sequência correta:

E- 3, 5, 4, 2, 1.

“Pode-se fazer uma analogia da UML como uma caixa de ferramentas. Um pedreiro usa sua caixa de ferramentas para realizar suas tarefas. Da mesma forma, a UML pode ser vista como uma caixa de ferramentas utilizada pelos desenvolvedores de sistemas para realizar a construção de modelos.”
Fonte: BEZERRA, E. Princípios de análise e projeto de sistemas com UML. 2. ed. Rio de Janeiro: Elsevier Editora, 2007. p. 16.

Considerando o conceito estudado sobre utilizações da UML, analise as afirmativas a seguir.
I. A UML é um auxílio ao projeto de software, permitindo criar e visualizar as estruturas de algoritmos em modelos que contribuem para a boa comunicação entre os membros da equipe.
II. A aceitação da UML é comprovadamente forte na área acadêmica por ter base teórica formal, o que a deixa complexa, enfraquecendo sua aceitação na indústria, que tem priorizado linguagens simples de modelagem.
III. A parte gráfica da UML propicia um ágil e facilitado entendimento por humanos, porém ocasiona complexidades de interpretações computacionais, trazendo ambiguidades que devem ser tratadas com expressões textuais.
IV. A UML se associa a processos, sendo utilizada como instrumental robusto em atividades para a escrita e visualização de modelos de software.
Está correto apenas o que se afirma em:

E- I e IV.

“Como toda ferramenta CASE de modelagem, a ArgoUML permite a edição dos principais diagramas UML, contém mecanismos de geração automática de código, a partir do modelo, auxilia a tarefa de engenharia reversa [...]”Fonte: PICHILIANI, M. C. Mapeamento de Software para permitir a colaboração síncrona. Dissertação de mestrado. 2006. (Mestrado em Engenharia Eletrônica e Computação) – Instituto Tecnológico de Aeronáutica, São José dos Campos, 2006. Disponível em: <http://www.comp.ita.br/~pichilia/argo/TeseVersaoFinal.pdf>. Acesso em: 08 nov. de 2019. p. 36.
Considerando essas informações e o conteúdo estudado, pode-se afirmar que a ArgoUML:

C- é escrita em Java e tem os diferenciais de possuir código aberto e mecanismos de realização de críticas automáticas nos modelos para orientar e auxiliar o desenvolvedor.

“A UML disponibiliza uma forma padrão de modelagem de projetos de sistemas, incluindo seus aspectos conceituais, tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, processos de banco de dados e componentes de software reutilizáveis.”
Fonte: PICHILIANI, M. C. Mapeamento de Software para permitir a colaboração síncrona. Dissertação de mestrado. 2006. (Mestrado em Engenharia Eletrônica e Computação) – Instituto Tecnológico de Aeronáutica, São José dos Campos, 2006. Disponível em: <http://www.comp.ita.br/~pichilia/argo/TeseVersaoFinal.pdf>. Acesso em: 08 nov. de 2019. p. 31.
Considerando essas informações e o conteúdo estudado sobre UML, pode-se afirmar que ela é:

A- uma linguagem com sintaxe bem definida para especificar, visualizar e documentar modelos de software no paradigma orientado a objetos, utilizando uma notação padrão.

“Tendo em conta a proliferação de aplicações nesta área, uma taxonomia das ferramentas CASE é particularmente importante, pois facilita a compreensão da abrangência de uma determinada ferramenta e da sua aplicabilidade nas fases e atividades do processo de desenvolvimento de software. Para além dessas questões, a classificação dessas ferramentas facilita ainda a realização de análises comparativas.”
Fonte: Da Silva, A. M. R. e Videira, C. A. E. UML, Metodologias e Ferramentas CASE. Lisboa: Editora Centro Atlântico. 2001. p. 261. Tradução nossa.

Considerando o trecho apresentado e o conceito estudado sobre classificação de ferramentas CASE, analise as afirmativas a seguir.
I. Upper-Case correspondem à categoria de ferramentas CASE utilizadas na fase de implementação (incluindo desenho técnico, de edição e compilação de código e de testes).
II. Lower-Case correspondem à categoria de ferramentas CASE especializadas na fase de concepção do software (ferramentas de análise e especificação e modelagem de requisitos).
III. Integrated CASE correspondem à categoria de ferramentas CASE que cobrem todo o ciclo de vida do desenvolvimento de software, desde a captura dos requisitos do sistema até o controle final da qualidade.
IV. A categorização das ferramentas CASE pode ser feita por diversos critérios, podendo incluir a análise das funcionalidades disponíveis.
Está correto apenas o que se afirma em:

A- III e IV.

“O UML providencia um número elevado de conceitos e notações particularmente concebidos de forma a satisfazer os requisitos típicos de modelagem de software. Contudo, podem surgir situações em que se torna desejável a introdução de conceitos e/ou de notações adicionais para além dos definidos originalmente no momento da definição do padrão.”
Fonte: Da Silva, A. M. R. e Videira, C. A. E. UML, Metodologias e Ferramentas CASE. Lisboa: Editora Centro Atlântico. 2001. p. 261. Tradução nossa.
Considerando essas informações e o conteúdo estudado sobre mecanismos de uso geral e de extensão da UML, analise os itens da UML a seguir e associe-os a suas respectivas descrições:
1) Estereótipos.
2) Notas.
3) Pacotes.
4) Tagged Values
5) Restrições.
( ) Adorno contendo comentários textuais anexados a um elemento ou a uma coleção de elementos. Permite anexar informações a um modelo, como requisitos, revisões e explicações.
( ) Recurso de separação que organiza elementos de modelagem em conjuntos maiores que possam ser manipulados como grupos. Realiza, então, o agrupamento de itens semanticamente relacionados.
( ) Conjunto de valores pré-definidos para um elemento, formando um par de valores que pode ser usado para adicionar propriedades a elementos de um modelo.
( ) Especificação de regras que delimitam conjunto de valores ou situações possíveis para um determinado elemento. É um recurso, então, utilizado para definir condições que devem ser mantidas como verdadeiras para que o modelo seja bem formado.
( ) Ampliação do vocabulário da UML, permitindo a criação de novos tipos de blocos de construção que são derivados dos já existentes, mas específicos a determinados problemas. Eles personalizam itens por meio de construções específicas para um domínio, plataforma ou método de desenvolvimento.
Agora, assinale a alternativa que apresenta a sequência correta:

A- 2, 3, 4, 5, 1.

“A introdução de ferramentas CASE numa organização pressupõe uma predisposição para a aplicação de regras e princípios a todo o processo de desenvolvimento, sendo esta pré-condição já de si um aspecto positivo no processo de melhoria do desenvolvimento de

software numa organização.”Fonte: DA SILVA, A. M. R. e VIDEIRA, C. A. E. UML, Metodologias e Ferramentas CASE. Lisboa: Editora Centro Atlântico. 2001. p. 409.
Considerando o trecho apresentado e o conceito estudado sobre ferramentas CASE, analise as afirmativas a seguir.
I. A escolha correta da ferramenta CASE em uma organização é essencial para o sucesso de um projeto de desenvolvimento de software.
II. Existe um fato que dificulta a concorrência comercial no universo das ferramentas CASE: as opções gratuitas são completas, abrangendo toda a UML, e tem bom suporte, dominando o mercado.
III. Na especificação da UML, tem-se a recomendação de 10 ferramentas CASE que são homologadas pelo OMG para uma boa modelagem, facilitando a escolha.
IV. O risco de um baixo desempenho de ferramenta CASE deve ser analisado nos projetos de desenvolvimento de software por interferir negativamente no trabalho da equipe.
Está correto apenas o que se afirma em:

C- I e IV.

“[...] os diversos diagramas fornecidos pela UML permitem analisar o sistema em diferentes níveis, podendo enfocar a organização estrutural do sistema, o comportamento de um processo específico, a definição de um determinado algoritmo ou até mesmo as necessidades físicas para a implantação do sistema.”Fonte: GUEDES, G. T. A. UML 2: uma abordagem prática. 2. ed. São Paulo: Novatec Editora, 2011. p. 30.
Considerando essas informações e o conteúdo estudado, pode-se afirmar que um diagrama da categoria de estrutura pode:

B- incluir elementos que representam classes e seus relacionamentos pertencentes à lógica de negócio implementada no software.

“O comportamento de uma sociedade de objetos ou de uma operação individual poderá ser especificado por meio de uma interação. As interações envolvem outros elementos, inclusive mensagens, ações e ligações (as conexões entre os objetos).”Fonte: BOOCH, G., RUMBAUGH, J., JACOBSON, I. UML, Guia do Usuário. 2. ed. Rio de Janeiro: Editora Campus, 2005. Versão digital. p. 58.
Considerando o trecho apresentado e o conceito estudado sobre os diagramas de comportamento da UML, analise as afirmativas a seguir.
I. Os diagramas de comportamento mostram modelos que contêm o comportamento dinâmico dos objetos em um software, incluindo métodos, colaborações e histórico de estados.
II. Os diagramas de comportamento estão bem associados à linguagem de programação utilizada, indicando como o código-fonte e bibliotecas de componentes estão estruturados de modo estático.
III. O comportamento dinâmico de um software pode ser descrito como uma série de alterações no seu ambiente ao longo do tempo, incluindo a sequência de solicitações de serviço feitas pelos objetos.
IV. Nos diagramas de comportamento, tem-se foco na estrutura das classes utilizadas pelo sistema, identificando-as e determinando os atributos e métodos que cada classe tem.
Está correto apenas o que se afirma em:

B- I e III.

“O estado de um objeto é, portanto, dinâmico. Assim, ao visualizar seu estado, você está realmente especificando o valor de seu estado em um determinado momento no tempo e no espaço. É possível exibir o estado de alteração do objeto, mostrando-o várias vezes no mesmo diagrama de interação, mas com cada ocorrência representando um estado diferente.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 183.
Considerando essas informações e o conteúdo estudado sobre o interaction overview diagram, pode-se afirmar que um dos exemplos do uso do diagrama de interação é:

C- a inserção de mensagens enviadas e recebidas entre os objetos.

“Uma das coisas para as quais os objetos serão utilizados será a modelagem de instâncias concretas que existem no mundo real. Por exemplo, para fazer a modelagem da topologia da rede de uma empresa, você usará diagramas de implantação.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 285.
Considerando essas informações e o conteúdo estudado sobre o component diagram, pode-se afirmar que, para utilizar uma visão estática adequada para a implantação de um determinado sistema, é necessário:

A- adotar uma modelagem de componentes físicos que se localizam em um nó.

“Você pode utilizar o diagrama de estrutura composta para mostrar os detalhes internos de um classificador e descrever os objetos e funções que trabalham juntos para executar o comportamento do classificador contido. Um diagrama de estrutura composta é similar a um diagrama de classe, mas ele representa peças individuais em vez de classes inteiras.”
Fonte: IBM. Diagramas de estrutura composta. Disponível em: <https://www.ibm.com/support/knowledgecenter/pt-br/SS5JSH_9.5.0/com.ibm.xtools.modeler.doc/topics/ccompstruc.html>. Acesso em: 05 nov. 2019.
O trecho apresentado trata do uso dos diagramas de estrutura composta. Considerando essas informações e o conteúdo estudado sobre composite structure diagram, pode-se afirmar que as peças são:

B- componentes de diagrama que simbolizam um grupo de diversas instâncias apresentados em um classificador estruturado contido.

“Um diagrama de implantação mostra a configuração dos nós de processamento em tempo de execução e os componentes neles existentes. Além disso, abrange a visão estática de implantação de uma arquitetura. Tipicamente, um nó inclui um ou mais artefatos.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 28. (Adaptado).
Considerando essas informações e o conteúdo estudado sobre o deployment diagram, pode-se afirmar que os diagramas de implementação:

C- se caracterizam por modelarem a arquitetura física de um sistema.

“Na UML, os diagramas de objetos fornecem uma captura instantânea das instâncias em um sistema e os relacionamentos entre as instâncias. Inicializando os elementos de modelos em um diagrama de classe, você pode explorar o comportamento de um sistema em um determinado momento.”

Fonte: IBM. Diagrama de objetos. Disponível em: <https://www.ibm.com/support/knowledgecenter/pt-br/SS5JSH_9.5.0/com.ibm.xtools.modeler.doc/topics/cobjdiags.html>. Acesso em: 05 nov. 2019.
Considerando essas informações e o conteúdo estudado sobre o object diagram, pode-se afirmar que uma das suas características está relacionada às:

D- instâncias exclusivas dos classificadores e os links entre essas instâncias em determinado instante.

“Objetos concretos aparecem em exemplos específicos, tais como diagramas de objetos, diagramas de componentes e diagramas de implantação. Os papéis aparecem em descrições genéricas, como diagramas de interação e diagramas de atividades.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 560.
O trecho apresentado trata de um dos aspectos dos diagramas de componentes. Considerando essas informações e o conteúdo estudado sobre o component diagram, pode-se afirmar que os diagramas de componentes se caracterizam por:

E- formar a modelagem que envolve aspectos físicos de um sistema orientado à objetos.

“Como classificadores, especificações de instância têm atributos que são referidos como slots. Uma especificação de instância pode ter um slot para cada recurso estrutural de seu classificador, incluindo recursos herdados. Você pode especificar valores para cada slot em uma especificação de instância, pois um tipo válido é definido para o slot.”
Fonte: IBM. Especificações em instâncias na UML. Disponível em: <https://www.ibm.com/support/knowledgecenter/pt-br/SS5JSH_9.5.0/com.ibm.xtools.modeler.doc/topics/cinstancespec.html>. Acesso em: 05 nov. 2019.
Considerando essas informações e o conteúdo estudado sobre o object diagram, pode-se afirmar que as especificações de instâncias são:

C- componentes que simbolizam uma instância presente em um sistema moldado.

“Na modelagem UML, é possível usar os diagramas de atividades para modelar a sequência das ações que devem ocorrer em um sistema ou aplicativo ou para descrever o que acontece em um fluxo de trabalho do processo de negócios.”
Fonte: IBM. Diagramas de Atividades. Disponível em:<https://www.ibm.com/support/knowledgecenter/pt-br/SS8PJ7_9.6.0/com.ibm.xtools.modeler.doc/topics/twrkactd.html>.Acesso em: 05 nov. 2019.
O trecho apresentado trata da utilização dos diagramas de atividades. Considerando essas informações e o conteúdo estudado sobre o activity diagram, analise as afirmativas a seguir.
I. Os diagramas de atividade se caracterizam por se manterem isolados a fim de realizar algumas ações ligadas à visualização, especificação e construção.
II. Os diagramas de atividades têm como função evidenciar o fluxo de controle de uma atividade direcionado para a mesma ação.

III. Os diagramas de atividades estão ligados ao desenvolvimento de sistemas executáveis, utilizados, por exemplo, através da Engenharia de Produção.

IV. Em um fluxo de controle criado por um diagrama de atividade, a mensuração de uma expressão é impraticável.

Está correto apenas o que se afirma em:

E- I e III.

"Um diagrama de comunicação mostra um conjunto de papéis, as conexões existentes entre esses papéis e as mensagens enviadas e recebidas pelas instâncias que representam os papéis. Use os diagramas de comunicação para ilustrar a visão dinâmica de um sistema."Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 162.
Considerando essas informações e o conteúdo estudado sobre o communication diagram, pode-se afirmar que o diagrama de comunicação:

C- é desenvolvido ao se inserir, primeiramente, os objetos e posteriormente representado pelos vínculos que interligam esses objetos.

"A modelagem de estruturas dos objetos envolve um retrato dos objetos de um sistema em um determinado momento. Um diagrama de objetos representa um quadro estático no enredo dinâmico representado por um diagrama de interação."Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 282.
O trecho apresentado traz uma das representações de um diagrama de interação. Considerando essas informações e o conteúdo estudado sobre o interaction overview diagram, pode-se afirmar que os diagramas de interação também são úteis:

D- na modelagem do fluxo de controle inserida no caso de uso.

"Como classificadores, especificações de instância têm atributos que são referidos como slots. Uma especificação de instância pode ter um slot para cada recurso estrutural de seu classificador, incluindo recursos herdados. Você pode especificar valores para cada slot em uma especificação de instância, pois um tipo válido é definido para o slot."
Fonte: IBM. Especificações em instâncias na UML. Disponível em: <https://www.ibm.com/support/knowledgecenter/pt-br/SS5JSH_9.5.0/com.ibm.xtools.modeler.doc/topics/cinstancespec.html>. Acesso em: 05 nov. 2019.
Considerando essas informações e o conteúdo estudado sobre o object diagram, pode-se afirmar que as especificações de instâncias são:

A- componentes que simbolizam uma instância presente em um sistema moldado.

"Um diagrama de componentes é um diagrama que mostra a organização e as dependências existentes em um conjunto de componentes. Além disso, os diagramas de componentes abrangem a visão estática de implementação de um sistema."Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 667. (Adaptado).
Considerando essas informações e o conteúdo estudado sobre component diagram, analise as modelagens disponíveis a seguir e associe-as com suas respectivas características.
1) Modelagem do código fonte.

2) Modelagem de versões do tipo executáveis.
3) Modelagem de bancos de dados físicos.
4) Modelagem de sistemas adaptáveis.
( ) Se refere aos componentes e possui uma versão que tem como foco atingir as áreas necessárias.
( ) É utilizada conjuntamente com determinados diagramas UML direcionados para auxiliar na modelagem de performance e simbolizar esses tipos de sistemas.
( ) Disponibiliza uma API com o objetivo de armazenar informações persistentes.
( ) Utiliza áreas de desenvolvimento que se caracterizam pela integração.
Agora, assinale a alternativa que apresenta a sequência correta:

C- 2, 4, 3, 1.

“O diagrama de classes, especificamente em UML 2, considera os diagramas de estrutura como uma classificação; não existe um diagrama em si chamado ‘diagrama de estrutura’. No entanto, o diagrama de classes oferece um ótimo exemplo do tipo de diagrama de estrutura e fornece um conjunto inicial de elementos de notação que todos os outros diagramas de estrutura usam.”
Fonte: BELL, D. O diagrama de classes. 2016. Disponível em: <https://www.ibm.com/developerworks/br/rational/library/content/RationalEdge/sep04/bell/index.html>. Acesso em: 05 nov. 2019.
O trecho apresentado traz aspectos dos diagramas de classes. Considerando essas informações e o conteúdo estudado sobre o class diagram, pode-se afirmar que uma das principais características destas classes é:

A- apresentar pacotes ou subsistemas utilizados para reunir elementos do seu modelo dentro de um grupo maior.

“No Visual Studio, você pode desenhar um diagrama de atividade para descrever um processo de negócios ou um algoritmo de software como um fluxo de trabalho por meio de uma série de ações. Pessoas, componentes de software ou dispositivos podem executar essas ações.”Fonte: MICROSOFT. Diagramas de atividade UML: diretrizes. 2016. Disponível em: <https://docs.microsoft.com/pt-br/visualstudio/modeling/uml-activity-diagrams-guidelines?view=vs-2015>. Acesso em: 05 nov. 2019.
O trecho apresentado trata de uma das funcionalidades dos diagramas de atividades com o uso do Visual Studio. Considerando essas informações e o conteúdo estudado sobre o activity diagram, analise as afirmativas a seguir e assinale V para a(s) verdadeira(s) e F para a(s) falsa(s).
I. ( ) Os estados de atividades se caracterizam por sua flexibilidade, em que suas atividades podem ser representadas pelos mesmos diagramas de atividade.
II. ( ) No momento em que a ação ou atividade pertencente a um estado se encontra suprida, o fluxo do controle se mantém no estado presente, determinado por meio de mutações.
III. ( ) A ramificação é um caminho alternativo, baseado em expressões booleanas e que poderá apresentar uma transição de entrada e várias saídas.
IV. ( ) As raias de natação são utilizadas nos fluxos de trabalho relacionados aos processos de negócios.
Agora, assinale a alternativa que apresenta a sequência correta:

C- F, F, V, V.

“Os diagramas de implantação são um dos dois tipos de diagramas empregados para a modelagem dos aspectos físicos de um sistema orientado a objetos. O diagrama de implantação mostra a configuração dos nós de processamento em tempo de execução e os artefatos que nele existem.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 606.
Considerando essas informações e o conteúdo estudado sobre o deployment diagram, pode-se afirmar que um nó representa:

B- os dispositivos de hardware e outros dispositivos responsáveis pelo suporte ao ambiente de tempo de execução dentro de um sistema.

“Em seus diagramas, os objetos ativos poderão aparecer sempre que houver objetos passivos. Você pode fazer a modelagem da colaboração de objetos ativos e passivos, incluindo diagramas de colaboração e de sequências. Um objeto ativo poderá aparecer como o destino de um evento em uma máquina de estados.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 95.
Considerando essas informações e o conteúdo estudado sobre communication diagram, pode-se afirmar que o diagrama de colaboração:

A- faz parte dos diagramas de interação.

“Um diagrama de sequência é um diagrama de interação que dá ênfase à ordenação temporal de mensagens. Um diagrama de sequência mostra um conjunto de papéis e as mensagens enviadas e recebidas pelas instâncias que representam os papéis.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 162.
Considerando essas informações e o conteúdo estudado sobre o communication diagram, pode-se afirmar que os diagramas de comunicação se diferenciam dos diagramas de sequência por conta:

A- do caminho e do número de sequência.

“Uma classe é uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica. Uma classe implementa uma ou mais interfaces. Você utiliza classes para capturar o vocabulário do sistema que está em desenvolvimento.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 95.
O trecho apresentado trata do conceito estrito de classes. Considerando essas informações e o conteúdo estudado sobre o class diagram, pode-se afirmar que a modelagem do vocabulário de um sistema:

D- define tanto o número de abstrações que pertencem ao sistema analisado quanto as que se encontram fora do sistema.

“Devido a todas as interações entre o ambiente externo e as peças internas precisarem passar por uma porta, você pode utilizar uma porta para isolar as peças internas de um objeto. Conectores vinculam portas a propriedades e chamam a comunicação entre duas ou mais instâncias.”
Fonte: IBM. Portas. Disponível em: <https://www.ibm.com/support/knowledgecenter/pt-br/SS5JSH_9.5.0/com.ibm.xtools.modeler.doc/topics/cports.html>. Acesso em: 05 nov. 2019. (Adaptado).

O trecho apresentado trata do uso das portas dentro dos diagramas de estrutura. Considerando essas informações e o conteúdo estudado sobre composite structure diagram, pode-se afirmar que uma porta:

E- é uma forma de interação estabelecida entre uma instância do classificador com o seu ambiente.

“A UML não apenas se destina a permitir que modelos sejam transformados em código, como também a aplicar uma engenharia reversa para converter código em modelos. Alguns itens podem ser escritos mais facilmente na sintaxe de uma linguagem de programação textual (por exemplo, as expressões matemáticas), enquanto outros itens são visualizados graficamente com maior clareza na UML (por exemplo, as hierarquias de classes).”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 60. (Adaptado).Considerando essas informações e o conteúdo estudado sobre a engenharia reversa com UML, pode-se afirmar que uma das funções desse tipo de engenharia é:

C- reverter um código fonte de software por meio das suas determinações com elevado nível de abstração.

“Os diagramas de atividades são um caso especial de diagramas de estados, em que todos ou a maioria dos estados são estados de atividades e todas ou a maioria das transições são ativadas pela conclusão de atividades no estado de origem.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 507.
Considerando essas informações e o conteúdo estudado sobre o State Machine Diagram, pode-se afirmar que o diagrama de máquinas de estado e de atividades é essencial para:

A- definir uma modelagem que estabeleça o tempo de vida referente a um determinado objeto.

“A engenharia reversa requer o suporte de ferramentas em conjunto com a intervenção humana. A combinação dos procedimentos de geração de código e de engenharia reversa permite uma engenharia de ciclo completo, o que significa a capacidade de trabalhar em modos de visualização gráfica ou textual.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 50.
Considerando essas informações e o conteúdo estudado sobre a engenharia reversa com UML, pode-se afirmar que esse tipo de engenharia é responsável por:

A- auxiliar na busca de informações, melhorar as operações existentes e diminuir o risco e o custo relacionados ao desenvolvimento de um software

“Um caso de uso é a descrição de sequências de ações realizadas pelo sistema que proporciona resultados observáveis de valor para um determinado ator. Um caso de uso é utilizado para estruturar o comportamento de itens em um modelo.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 55.
Considerando essas informações e o conteúdo estudado sobre o Use Case Diagram, pode-se afirmar que uma de suas características é:

C- fornecer ao conjunto formado por sistemas, subsistemas e classes um elevado nível de acessibilidade e compreensão.

“Um perfil é uma UML com um conjunto de estereótipos predefinidos, valores atribuídos, restrições e classes de base. Ele também seleciona um subconjunto dos tipos de elementos da UML para uso, de maneira que um modelador não fique confuso pelos tipos de elementos que não são necessários para a área de aplicação particular.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 144.
O trecho apresentado trata do conceito básico referente ao perfil dentro de uma UML.
Considerando essas informações e o conteúdo estudado sobre o Profile Diagram, analise as afirmativas a seguir e assinale V para a (s) verdadeira (s) e F para a (s) falsa (s).
I. ( ) Originalmente, a linguagem UML foi criada para as plataformas de perfil.
II. ( ) Ao desenvolvermos um perfil, cria-se uma extensão da UML em um nível mais expansivo.
III. ( ) Um modelo se caracteriza por capturar uma visão pertencente a um sistema físico.
IV. ( ) Um metamodelo pode ser entendido como um modelo que estabelece uma linguagem para apresentar outros modelos.
Agora, assinale a alternativa que apresenta a sequência correta:

A- F, F, V, V.

“Um diagrama de sequências é um diagrama de interação cuja ênfase está na ordenação temporal das mensagens; o diagrama de comunicações é um diagrama de interação cuja ênfase está na organização estrutural dos objetos ou papéis que enviam e recebem mensagens.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 560.
Considerando essas informações e o conteúdo estudado sobre Sequence Diagram, analise os tipos de execução disponíveis a seguir e associe-os com suas respectivas características.
1) Execução opcional.
2) Execução condicional.
3) Execução paralela.
4) Execução de loop ou iterativa.
( ) Representado pela tag alt, este tipo de controle se caracteriza pelo fato de o corpo do operador ser dividido em diversas partes.
( ) Caracteriza-se pelo fato de o corpo ser executado de maneira repetitiva quando a condição de guarda for considerada como verdadeira.
( ) Representado pelo uso da tag par, este tipo de execução se caracteriza pelo fato de cada sub-região simbolizar um modelo de computação denominada paralela.
( ) A tag que o compõe é denominada de opt. Neste tipo de controle, o corpo do operador de controle é realizado caso uma condição de guarda seja considerada como verdadeira.
Agora, assinale a alternativa que apresenta a sequência correta:

C- 2, 4, 3, 1.

"Você pode criar e projetar perfis UML usando diagramas de classe, que fornecem uma abordagem visual, de modelagem, para criar perfis que podem ser particularmente úteis quando se cria relacionamentos. Embora você possa criar e modificar perfis na visualização Explorador de Projetos, a modelagem de perfis é um método alternativo que pode ser mais rápido e mais fácil."Fonte: IBM KNOWLEDGE CENTER. Autorizando perfis UML com diagramas de classe. Disponível em: <https://www.ibm.com/support/knowledgecenter/pt-br/SS5JSH_9.1.2/com.ibm.xtools.profiles.doc/topics/t_adding_class_dgms_to_profiles.html>. Acesso em: 27 nov. 2019.
O trecho apresentado trata da criação de modelagem de perfis. Considerando essas informações e o conteúdo estudado sobre o Profile Diagram, pode-se afirmar que, nas metaclasses, as instâncias:

A- são elementos concretos em uma UML.

"É importante deixar claro que um diagrama de sequência mostra um conjunto de papéis e as mensagens enviadas e recebidas pelas instâncias que representam os papéis. Use os diagramas de sequência para ilustrar a visão dinâmica de um sistema."Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 285.
Considerando essas informações e o conteúdo estudado sobre o Sequence Diagram, pode-se afirmar que sua principal função é:

E- criar uma ordenação temporal das mensagens.

"Os diagramas de casos de uso são importantes para visualizar, especificar e documentar o comportamento de um elemento. Esses diagramas fazem com que sistemas, subsistemas e classes fiquem acessíveis e compreensíveis, por apresentarem uma visão externa sobre como esses elementos podem ser utilizados no contexto."Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 363.
Considerando essas informações e o conteúdo estudado sobre o Use Case Diagram, pode-se afirmar que, na modelagem da visão de caso de uso pertencente a um cenário, as duas formas de usar os diagramas de casos de uso são:

D- a modelagem do contexto de um cenário e a modelagem dos requisitos de um sistema.

"Um diagrama de estados mostra uma máquina de estados, dando ênfase ao fluxo de controle de um estado para outro. Uma máquina de estados é um comportamento que especifica as sequências de estados pelos quais um objeto passa durante seu tempo de vida."Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 510-511.
Considerando essas informações e o conteúdo estudado sobre a State Machine Diagram, pode-se afirmar que os diagramas de estados têm como aspecto o fato de serem utilizados:

C- na modelagem referente aos aspectos dinâmicos pertencentes a um sistema.

"Um classificador (classifier) é um supertipo definido no metamodelo UML usando extensivamente ao longo da especialização ou sempre que pretendemos nos referir a um elemento que descreve estrutura de comportamento."Fonte: RAMOS, R. Treinamento prático em UML. São Paulo: Universo dos Livros, 2006. p. 95.
Considerando essas informações e o conteúdo estudado sobre a Profile Diagram, pode-se afirmar que uma metaclasse classifier é uma:

E- metaclasse abstrata que simboliza uma classificação de instâncias.

“Começando com as mensagens que iniciam a interação, distribua cada mensagem subsequente de cima para baixo entre as linhas de vida, mostrando as propriedades de cada mensagem (como seus parâmetros), conforme necessário para explicar a semântica da interação. Se for necessário visualizar o aninhamento das mensagens ou dos pontos no tempo quando a computação real estiver sendo realizada, adorne a linha de vida de cada objeto com seu foco de controle.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 393.
Considerando essas informações e o conteúdo estudado sobre Sequence Diagram, pode-se afirmar que a linha de vida é uma:

A- linha esboçada verticalmente, simbolizando a existência de um objeto em um determinado período.

“Os pacotes bem estruturados agrupam elementos que estão próximos semanticamente e que tendem a se modificar em conjunto. Portanto, os pacotes bem estruturados são fracamente acoplados e muito coesos, com acesso altamente controlado ao conteúdo do pacote.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 259.O trecho apresentado refere-se aos aspectos do conceito de pacotes.
Considerando essas informações e o conteúdo estudado sobre o Package Diagram, pode-se afirmar que uma característica referente aos diagramas de pacotes é o fato de:

A- possibilitar o controle ao acesso dos seus conteúdos e as emendas visualizadas dentro da arquitetura do sistema.

“Um diagrama de casos de uso exibe um conjunto de casos de uso e atores (um tipo especial de classe) em seus relacionamentos. Diagramas de caso de uso abrangem a visão estática de casos de uso do sistema. Esses diagramas são importantes principalmente para a organização e a modelagem de comportamentos do sistema.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 64.
Considerando essas informações e o conteúdo estudado sobre o Use Case Diagram, pode-se afirmar que um dos seus objetivos é:
A- compartilhar propriedades similares a todos os outros diagramas.

“É possível criar diversos diagramas de tempo, em que cada um focalize uma linha de vida ou visualização diferente da interação. Os elementos incluídos em um diagrama de sequência não são incluídos no diagrama de tempo correspondente. No entanto, os elementos incluídos em um diagrama de tempo são incluídos no diagrama de sequência correspondente.”Fonte: IBM KNOWLEDGE CENTER. Diagramas de Tempo UML. Disponível em: <https://www.ibm.com/support/knowledgecenter/pt-br/SS5JSH_9.1.2/com.ibm.xtools.model.timing.diagram.doc/topics/c_timingdiagram.html>. Acesso em: 27 nov. 2019.
Considerando essas informações e o conteúdo estudado sobre Timing Diagram, pode-se afirmar que o formato proporcional se caracteriza por auxiliar:

D- na compreensão da maneira como os eventos são realizados em tempo real.

“Os pacotes ajudam a organizar os elementos em modelos, de maneira que você seja capaz de compreendê-los com maior facilidade. Os pacotes também permitem controlar o acesso a seus conteúdos, de modo que você possa controlar as costuras existentes na arquitetura do sistema.” Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 261.
Considerando essas informações e o conteúdo estudado sobre Package Diagram, analise as afirmativas a seguir.
I. O pacote define um espaço de nome, indicando que os componentes de mesmo modelo precisam ser expostos de uma única maneira dentro do pacote que contém estes componentes.
II. Os elementos de modelos distintos poderão apresentar nomes diversificados dentro de um pacote.
III. É possível controlar a visibilidade dos elementos de um pacote de maneira similar à visibilidade dos atributos e operações que compõem uma classe.
IV. Os pacotes possuem papel essencial, visto que auxiliam na ampliação dos elementos que constituem um diagrama.
Está correto apenas o que se afirma em:

E- I e III.

“Os diagramas de tempo focalizam o tempo ou duração da mensagem ou condições em mudança em uma linha de tempo no diagrama. Você cria diagramas de tempo para representar uma parte do tempo de um sistema. É possível usar diagramas de tempo para examinar e modelar ainda mais as restrições de tempo e duração.” Fonte: IBM KNOWLEDGE CENTER. Diagramas de Tempo UML. Disponível em: <https://www.ibm.com/support/knowledgecenter/pt-br/SS5JSH_9.1.2/com.ibm.xtools.model.timing.diagram.doc/topics/c_timingdiagram.html>. Acesso em: 27 nov. 2019.
Considerando essas informações e o conteúdo estudado sobre Timing Diagram, pode-se afirmar que o formato ordenado é útil para:

E- conseguir um número de dados e aplicar em formato inicial, a fim de analisar os dados de volume elevado e avaliar a exibição e impressão.

“Um diagrama de temporização é um diagrama de interação que mostra os tempos reais em diferentes objetos ou papéis, em vez das sequências de mensagens relativas. Ele mostra os momentos exatos em que as mensagens são trocadas pelos papéis.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 337.
O trecho apresentado trata do conceito de diagramas de tempo. Considerando essas informações e o conteúdo estudado sobre Timing Diagram, pode-se afirmar que a sua principal diferença em relação ao diagrama de máquinas de estado é:

D- o fato de que o diagrama de tempo muda o estado de um objeto ao longo do tempo.

“Um pacote pode conter outros elementos, incluindo classes, interfaces, componentes, nós, colaborações, casos de uso, diagramas e até outros pacotes. A propriedade de elementos é um relacionamento composto, significando que os elementos são declarados no pacote.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 263.
O trecho apresentado traz informações sobre os elementos que compõem um pacote. Considerando essas informações e o conteúdo estudado sobre o Package Diagram, pode-se afirmar que, para determinar a visibilidade de um elemento que constitui um pacote, é preciso:

D- utilizar o nome do elemento como prefixo.

“Um objeto reativo é aquele cujo comportamento é mais bem caracterizado por sua resposta a eventos ativados externamente ao seu contexto. Um objeto reativo tem um claro tempo de vida cujo comportamento atual é afetado pelo seu passado.”Fonte: BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: guia do usuário. Rio de Janeiro: Elsevier, 2012. p. 508.
O trecho apresentado traz o conceito de objeto reativo. Considerando essas informações e o conteúdo estudado sobre os State Machine Diagram, pode-se afirmar que uma das ações para criar modelos de um objeto reativo é:

E- selecionar os estados inicial e final para o objeto.