Na construção de estruturas de herança em uma linguagem de programação, quando uma classe herda de outra, membros da classe base são incorporados como membros da classe derivada. Como as restrições de acessos são gerenciadas em classes diferentes, principalmente o acesso aos membros da classe base a partir das derivadas, deve-se aplicar os especificadores de acesso para obter o controle de acesso à classe base.

Podem ser exemplificados como especificadores de acesso

I. priority, no qual é oferecida a garantia que estes atributos terão prioridade durante a execução do código.

II. protected, que possui atributos visíveis pelas classes derivadas, permitindo flexibilidade ao programador.

III. public, que garante que tudo o que se aplica a objetos da classe base será aplicado aos objetos da classe derivada.

IV. private, no qual membros private só podem ser acessados por funções da classe base que se utilizam das informações contidas nos mesmos.

É correto o que se afirma em

Alternativas
A)
I e II, apenas.

B)
II e III, apenas.

C)
I, II e III, apenas.

D)
II, III e IV apenas.

E)
I, III e IV, apenas.

Em relação à orientação a objetos, considere as informações abaixo.

Um dos mecanismos fundamentais na programação orientada a objetos é o conceito de redefinição, que ocorre quando um método, cuja assinatura já tenha sido especificada, recebe uma nova definição em uma classe derivada.

A linguagem de programação orientada a objetos deve suportar o conceito de ligação tardia (late binding), visto que a definição do método que é candidato a ser efetivamente invocado só ocorre durante a execução do programa.

O mecanismo de redefinição, juntamente com o conceito de ligação tardia, é a chave para a utilização adequada de

Alternativas

A)
polimorfismo.

B)
objeto de classes derivadas.

C)
restrições de acesso.

D)
polimorfismo e herança.

E)
dois métodos de uma mesma classe.

Os métodos, também chamados de funções ou procedimentos em algumas linguagens, ajudam a modularizar um programa, separando suas tarefas em unidades independentes. Você declarou métodos em todos os programas que escreveu. As instruções que ficam no corpo dos métodos são escritas apenas uma vez, ficam ocultas dos outros métodos e podem ser reutilizadas em vários lugares em um programa. Embora a maioria dos métodos seja executada em resposta a chamadas de método em objetos específicos, nem sempre esse é o caso. Às vezes, um método executa uma tarefa que não depende do conteúdo de nenhum objeto. Tal método se aplica à classe na qual é declarado como um todo e é conhecido como método estático ou método de classe.

DEITEL, Paul; DEITEL, Harvey; DEITEL, Abbey. **Android**: como programar. 2 ed. Porto Alegre: Bookman, 2015.

Com base no texto supracitado, analise as asserções a seguir.

I. Dividir um programa em métodos torna-o mais difícil de ser depurado e mantido.

II. Um motivo para modularizar um programa com métodos é evitar repetição de código.

III. Um motivo para modularizar um programa com métodos é a reutilização de software.

IV. Um motivo para modularizar um programa com métodos é a facilidade de manejar o desenvolvimento de programas.

V. É possível criar programas principalmente a partir de métodos padronizados, em vez de desenvolver código personalizado.

É correto o que se afirma em

Alternativas

A)
I e II, apenas.

B)
II e III, apenas.

C)
II, IV e V, apenas.

D)
II, III, IV e V, apenas.

E)
I, II, III, IV e V.