

Técnicas de Programação

Lista 4

1º)

a)Classe: As classes podem ser exemplificados como objetos reais, em um programa normalmente as classes devem ter suas próprias estruturas abstratas. Todas as classes devem ter estados e comportamentos, que devem ser encontrados através da observação de como a classe funciona e o que ele deve possuir, podemos tomar um carro como exemplo, um carro tem estados e comportamentos e posso observar eles através da sua abstração, ou seja, um carro tem placa, chassi e etc, e possui comportamentos como o de acelerar e parar.

b)Objeto: De forma simplificada os Objetos são as instâncias de classes, os objetos terão estados e comportamentos, onde seus estados devem ser definidos pelos atributos da classe, e os seus comportamentos definidos pelos métodos das classes. Pegando o exemplo do item A, quando o objeto é instanciado em algum lugar do código, é possível observar que ele irá possuir atributos e métodos para serem utilizados, como no exemplo do carro, ao criar a classe carro (considerando que tenha sido criado o método de 'acelerar'), e instanciar o objeto dentro do código, pode ser usado o método de 'acelerar', onde vai executar e realizar o que foi determinado para 'acelerar' dentro da classe carro.

c)Métodos e Atributos:Os métodos são os comportamentos da classe, normalmente nomeados com verbos, como no exemplo do carro, o método 'acelerar' é comportamento que a classe carro pode exercer. Os métodos são compostos de:

- Identificador: nome do método, normalmente um verbo.
- Visibilidade do método.
- Lista de argumentos.

A estrutura do método é análoga a função vista anteriormente.

Os Atributos serão as propriedades de um objeto, ou seja, as características do objeto, usando o exemplo do carro podemos observar que possui atributos como chassi, placa, velocidade máxima, preço e etc.

d)Encapsulamento: O encapsulamento é uma forma de proteger/esconder as variáveis criadas, sendo possível 3 formas de encapsulamento.

e)Visibilidade de atributos e métodos:

A visibilidade de atributos e métodos podem ser definidas e alteradas por:

- 1) Public: Onde a variável será público para todos os lugares que for utilizada;
- 2) Private: Onde a variável somente será utilizada dentro da própria classe;
- 3) Protected: Onde a variável poderá ser utilizada somente dentro do mesmo pacote.

f)Herança: Herança é uma funcionalidade da Programação Orientada a Objetos que consiste em uma classe filha herdar atributos e métodos da classe pai, e para ser uma classe filha a classe deve ter no início do código a palavra “extends” e em seguida a classe que deseja que seja o pai para herdar seus atributos e métodos.

Obs: quando uma classe não herda explicitamente de ninguém, automaticamente ela já herdará da classe Objects.

g)Polimorfismo: O termo polimorfismo se refere à capacidade de um objeto assumir diferentes formas, ou seja, uma mesma operação pode ser realizada por diferentes objetos de classes diferentes, o que corresponde exatamente ao nome Polimorfismo(várias formas).

h) Classes abstratas: São classes que servirão somente para serem herdadas por classes filhas, ao colocar abstract ao lado do nome da classe dentro do código, essas classes não poderão mais ser instanciadas, só servirão para serem herdadas por outras classes.

i) Construtores: Os construtores são criados com o mesmo nome da classe, e quando iniciado a classe eles já são iniciados juntamente de forma implícita, eles não podem ter indicação de retorno e cada vez que o objeto for instanciado no código ele deverá conter os parâmetros do construtor junto a instância.

Obs: Usando o mecanismo de sobrecarga, mais de um construtor pode ser definido para uma classe, porém, um deles tem de ser obedecidos após instanciar o objeto.

j)Get e Set: o get é uma estrutura utilizada normalmente para pegar algum dado, costumeiramente é utilizado quando algum dado em uma classe é privado e o

desenvolvedor quer pegar esse dado para algo, então usa-se o get seguido do nome do atributo e em seguida retorna este atributo, desta forma, é “burlado” a privacidade do atributo.

Já o set tem a função de atribuir valor ao atributo normalmente.

k) Sobrecarga de métodos: a sobrecarga de métodos faz com que possa existir vários métodos com o mesmo nome em uma classe, sendo que eles devem ter tipos de argumentos diferentes, valor de retorno ou variáveis diferentes para que possa ter mesmos nomes.

l) Sobrescrita de métodos: é a capacidade de uma classe filha herdar de outra classe pai, e quando precisar, acabar sobrescrevendo algum método, e ao instanciar o objeto da classe filha e precisar usar o método, o método a ser utilizado será o sobrescrito do filho.

m) Palavras reservadas(super, this, final): A palavra super é quando estamos trabalhando com herança e queremos referenciar a classe pai, então usamos o super.

A palavra this é utilizada para referenciar atributos da classe ou da própria classe.

A palavra reservada final faz com que uma classe não possa mais herdar para outras classes, ou seja, não pode ter filhos.

n) Relação de objetos: “ter”, “usar” e “ser”: o verbo “ter” acontece quando eu somente tenho um objeto declarado na minha classe.

Já o verbo “usar” é quando eu somente instancio um objeto para usá-lo diretamente no objeto.

Já o verbo “ser” é quando