* **Classe**

Abstração dos objetos (tem informações genéricas, o que o objeto vai conter)

* **Objeto**

Instancia da classe

* **Atributo**

Características do objeto, ou seja a estrutura de dados que vai representar a classe.

* **Método**

Ações do objeto

* **Método Construtor**
* Mesmo nome da classe ;
* Instanciar o objeto;
* Se não houver um método construtor ele é uma classe abstrata
* Se a classe não for abstrata e não definir um método construtor definido, por padrão é vazio;
* Pode existir o método construtor com o mesmo nome porem com assinatura do método diferente (atributos diferentes);
* Uma classe sem método construtor é uma classe abstrata.
* **Método assessor (Getter e Setter)**

Para dar acesso para os atributos encapsulados, onde para cada instância de variável, um método Getterr retorna seu valor, enquanto um método Setter o define ou atualiza.

* **Método To String**

Converte um objeto para String;

* **Collections.sort ();**

Ordenar listas.

* **Classe Compareble com o método Compare To**

Define o critério de ordenação

* **To String**

Retorna uma representação string de um objeto.

* **This**

Com o This você identifica a variável local da classe, se não usar o This vai chamar sempre a variável do método.

* **Herança**

Classe herda as características de outra classe tanto os atributos quanto os métodos.

Não pode ter heranças múltiplas (uma criança não pode ter dois pais), porem pode ter várias classes herdando de uma (vários filhos de um mesmo pai).

* **Private**

Private é um modificador de acesso que restringe totalmente o acesso aquele recurso da classe de todas as demais classes, sejam elas do mesmo pacote, de outros pacotes ou até subclasses. Este modificador pode ser aplicado a atributos e métodos.

* **Polimorfismo**

São métodos com o mesmo nome porem comportamentos diferentes de acordo com a instancia da classe.