Conceitos de orientação a objetos

Herança

Um dos conceitos que a programação orientada a objetos nos traz é a herança, como o próprio nome já diz nada mais é do que herdar elementos de alguém muito próximo. Neste caso estamos falando de classe pai e filha.

Como já foi comentado em java, todas as classes criadas são filhas naturalmente de uma classe Object.

Para especificar que uma classe é filha de outra precisamos utilizar uma palavra reservada extends. Vejamos o exemplo e seguimos para o conceito na prática.

```
interface Animal {
   public void andar();
}

class AnimalAquatico implements Animal{
   ...
}


class AnimalTerreste extends Animal {
   Integer quantidadePatas;
   ...
}

class Gato extends AnimalTerreste {
   ...
}

class Cachorro extends AnimalTerreste {
}

class Peixe extends AnimalAquativo {
}
```

Abstração

O próprio nome já define seu conceito. Abstração é a capacidade de extrair algo, de simplificar deixar apenas o elementar. Para tal aqui precisamos pensar no objeto como na vida real, ter claramente cada característica e ação (atributo e método) e definir tudo isto a nível de classes.

```
public abstract class Animal {
   private Integer quantidadePatas;
   private String nome;
   private String som;
```

```
   public void setQuantidadePatas(Integer qtd) {
      quantidadePatas = qtd;
   }

   public abstract void emitirSom(String som);
}

class Cachorro extends Animal{
  public void emitirSom(String som) {
    ...
  }
}
```

Polimorfismo

Polimorfismo é um outro conceito de programação orientado a objetos que existe uma relação direta com herança uma vez que estabelecemos uma classe como filha (herança de outra) podemos trabalhar o comportamento de se comportar de um tipo ou de outro. Alguns padrões de projeto utilizam-se muito desta característica como por exemplo o Factory.

Para simplificar tal definição vamos para a prática seguindo e considerando as classes definidas no conceito de herança exposto anteriormente:

```
Animal a1 = new Cachorro();
Animal a2 = new Gato();

Map<String, String> map = new HashMap<String, String>();
List<String> list = new ArrayList<String>();
```

Encapsulamento

O conceito de encapsulamento é um dos mais interessantes na programação orientada a objetos pois trata-se de organizar e proteger o codigo do próprio programador. Estranho? No primeiro momento sim. Mas vejamos isso com outros olhos. Imagina uma sala de aula onde qualquer um pode chegar na frente e explicar algo; O que poderia ocorrer? Toda vez que algo tivesse uma mudança seria necessário chegar em todos os alunos e explicar um por um tal alteração. Bem, sendo mais prático em Java o encapsulamento se dá através da palavra PRIVATE. Vejamos uma aplicação da palavra reservada no código abaixo:

```
class Animal {
   private Integer quantidadePatas;
   private String nome;
```

```
   public void setQuantidadePatasInteger qtd) {
      quantidadePatas = qtd;
(    }
}
```