

Revisão Java

Métodos

* Estrutura: <modificador de acesso> **<opcional static>** <tipo de retorno> <nome> (<tipo param> <param>) {
ex:

```
//sem retorno
public void imprimir(String texto) {
    System.out.println(texto);
}
//com retorno
public String getNome() {
    return this.nome;
}
```

* Escopo da declaração de um método: Sempre deve estar dentro da classe

```
public class Pessoa {
    //sempre a declaração deve estar nesse escopo!
}
```

Declaração de métodos e variáveis

Sempre verificar se estamos utilizando modificadores de acesso, ou seja, não utilizar o default

```
private String nome;
```

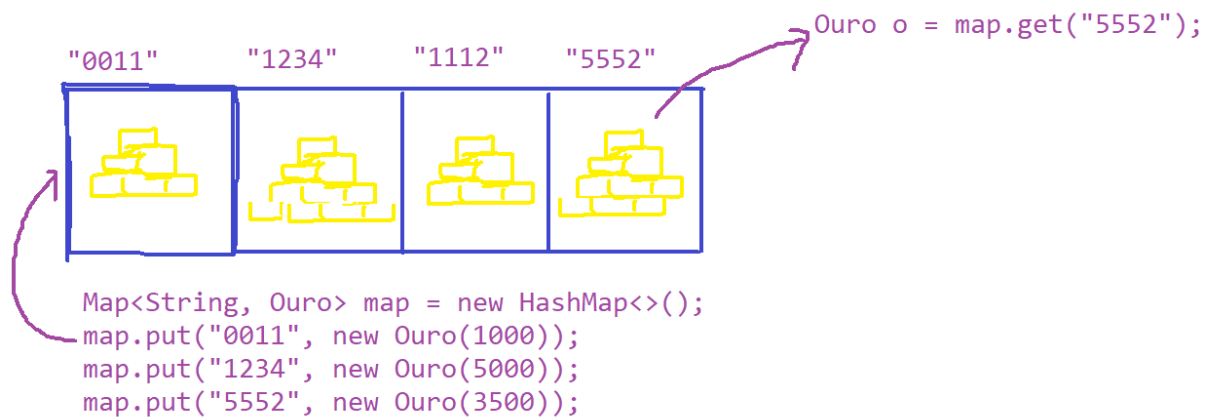
Map

O mapa sempre tem uma chave e um valor vinculado a essa chave. Na declaração do mapa, devemos informar o tipo da chave e o tipo do valor.

Sempre utilizar a interface mapa para se referir a um HashMap

Se a chave do mapa for um objeto criado por nós (nossas classes), devemos sobrescrever o equals e o hashCode, para informar quando um objeto é igual ao outro.

```
Map<tipo da chave, tipo do valor> mapa = new HashMap<>();
ex:
Map<String, Pessoa> mapaDePessoas = new HashMap<>();
```



ArrayList

permite armazenar elementos do tipo definido (entenda como generics <>).
importante prestar atenção no import, que deve ser **java.util**

```
List<String> lista = new ArrayList<>();
lista.add("Maça");

Animal gato = new Animal();
List<Animal> animais = new ArrayList<>();
animais.add(gato);
```

Se utilizarmos o **contains**, será verificada a igualdade utilizando o método equals do objeto.

Polimorfismo

O polimorfismo pode ser obtido através de, **Herança** (classes concretas ou abstratas) e **Interfaces**.

É a capacidade de referenciar um mesmo objeto de várias formas.

Onde o polimorfismo brilha? R: ao receber como parâmetro de um método. Em um tipo de uma lista ou mapa.

