

**Aluno: Francisco Marden Torres Da Silva**

1)

**R:** Nele podemos utilizar o reaproveitamento de métodos criados na super classe uma vez que seu comportamento irá variar de acordo com a maneira/modo de que se é chamado.

2)

**R:** O software que invoca o comportamento polimórfico é independente dos tipos de objeto para os quais as mensagens são enviadas. Novos tipos de objetos que podem responder a chamadas de método existentes podem ser incorporados a um sistema sem exigir modificações no sistema básico

3)

**R:** <https://github.com/mardentorres/POO/tree/main/Exerc%C3%ADcio%20-%20Polimorfismo>

4)

**R:** JAVA 134.34

5)

**R: Falso:** A Sobrecarga nos permite utilizar o mesmo nome em mais de um método, porém, com argumentos diferentes, para que se possa diferencia-los. Caso isso não ocorrer, o programa principal não reconhecerá.

7)

`t = new Professor()` ; Compila. Está sendo instanciado um objeto da classe Professor.

`p = t`. Compila; O objeto `t` herda da superclasse Pessoa.

`s = (Aluno) t`; Não compila. `s` e `t` são subclasses (especializações da superclasse Pessoa) ocupando a mesma hierarquia na árvore de herança.

`s = (Aluno) p`; Compila. Quando se faz `(Aluno) p`, estamos especializando uma superclasse, no caso a superclasse Pessoa de forma criar a subclasse do tipo Aluno.

`p = new Aluno()`; Compila. Qualquer objeto de uma classe que herde de uma superclasse pode ser atribuído a qualquer parente hierárquico superior a ele e, inclusive a ele mesmo.

t = Pessoa(); Não compila, Nesse caso, não existe em um nível hierárquico “maior” que Pessoa.

t = p; Não compila. pois é como o objeto da classe pessoa herdasse de Professor, mas professor não é superclasse e sim subclasse e herda de Pessoa