

A herança é um mecanismo da Orientação a Objeto que permite criar novas classes a partir de classes já existentes, aproveitando-se das características existentes na classe a ser estendida. Este mecanismo é muito interessante, pois promove um grande reuso e reaproveitamento de código existente. Com a herança é possível criar classes derivadas, subclasses, a partir de classes bases, superclasses. As subclasses são mais especializadas do que as suas superclasses, mais genéricas. As subclasses herdam todas as características de suas superclasses, como suas variáveis e métodos.

Referencia: <https://www.devmedia.com.br/encapsulamento-polimorfismo-heranca-em-java/12991>

Ferramenta :

1. Verificar se todos os atributos e métodos estão com seus respectivos modificadores de acesso;
2. Verificar se está sendo respeitada a hierarquia entre as classes;
3. Verificar a viabilidade dos atributos e métodos nas classes filhas;
4. Caso a classe pai ter um método abstrato, verificar se na classe filha o mesmo foi implementado.

Inspeção Classes devem ser inspecionadas com a ferramenta Referenciar origem

Exemplo 1 - Java.util.Random()

1- Verdadeiro

2 - Falso

3 - Falso

4 - Falso

Exemplo 2 - Java.util.Calendar()

1 - Verdadeiro 2 - Falso 3 - Falso 4 - Falso

Exemplo 3 - (

<https://github.com/zozfabio/Empregos/blob/master/src/java/br/edu/unidavi/empregabilidade/util/servlets/usuario/UsuarioSalvar.java>)

1 - Verdadeiro

2 - Verdadeiro

3 - Verdadeiro

4 - Falso

Exemplo - 4 (

<https://github.com/eldermoraes/javaoneus2016/blob/master/app/javaonedb/src/main/java/com/javaone/servlet/DataServlet.java>)

1 - Verdadeiro

2 - Verdadeiro

3 - Verdadeiro

4 - Falso

Exemplo - 5 (

<https://github.com/Wrakor/GarbageCollector/blob/master/code/GarbageCollector/src/main/ContainerThread.java>)

1 - Verdadeiro

2 - Verdadeiro

3 - Verdadeiro

4 - Falso