@FunctionalInterface Annotation with respect to Inheritance

|  |  |  |  |
| --- | --- | --- | --- |
| Case 1 | Case 2 | Case 3 | Case 4 |
| If an interface extends functional interface and child interface does not contain any abstract method, the child interface is always function interface | In the child interface we can define exactly same parent interface abstract method | In the child interface, we can’t define any new abstract methods. If we define then we get Unexpected @FunctionalInterface annotation  @FunctionalInterface  ^  MyFunctionalInterface is not a functional interface  multiple non-overriding abstract methods found in interface MyFunctionalInterface  1 error | In the child interface we can define exactly same parent interface abstract method  (we have not annotated child interface as @FunctionalInterface)  This valid. |
| @FunctionalInterface  **public** **interface** ParentInterface {  **public** **void** test();  }  @FunctionalInterface  **public** **interface** ChildInterface **extends** ParentInterface{    } | @FunctionalInterface  **public** **interface** ParentInterface {  **public** **void** test();  }  @FunctionalInterface  **public** **interface** ChildInterface **extends** ParentInterface{  **public** **void** test();  } | @FunctionalInterface  **public** **interface** ParentInterface {  **public** **void** test();  }  @FunctionalInterface  **public** **interface** ChildInterface **extends** ParentInterface{  **public** **void** test2();  } | @FunctionalInterface  **public** **interface** ParentInterface {  **public** **void** test();  }  **public** **interface** ChildInterface **extends** ParentInterface{  **public** **void** test2();  } |

There is no restriction on static and default methods