Child.java

*Overriding Rules*

class A {}

class B extends A {}

class C extends B {}

class EA extends Exception {}

class EB extends EA {}

class EC extends EB {}

abstract class Parent {

protected abstract B method(B b) throws EB;

protected abstract B exceptionMethod(B b) throws IllegalArgumentException;

protected abstract int primitiveMethod(int i);

//protected static B staticMethod(B b); //Compilation fails. This method requires a body instead of a semicolon

}

public class Child extends Parent {

**//Method parameter must be exactly same**

/\*@Override

protected B method(A a) throws EB { //Compilation fails. The method method(A) of type Child must override or implement a supertype method

return new B();

}\*/

/\*@Override

protected B method(C c) throws EB { //Compilation fails. The method method(C) of type Child must override or implement a supertype method

return new B();

}\*/

/\*@Override

protected int primitiveMethod(long i) { //Compilation fails. The method primitiveMethod(long) of type Child must override or implement a supertype method

return 0;

}\*/

**//Access modifier must be more accessible than parent**

/\*@Override

private B method(B b) throws EB { //Compilation fails. Cannot reduce the visibility of the inherited method from Parent

return new B();

}\*/

/\*@Override

B method(B b) throws EB { //Compilation fails. Cannot reduce the visibility of the inherited method from Parent

return new B();

}\*/

/\*@Override

protected B method(B b) throws EB { //OK

return new B();

}\*/

/\*@Override

public B method(B b) throws EB { //OK

return new B();

}\*/

**//return type must be same or subclass of return type of parent**

/\*@Override

protected B method(B b) throws EB { //OK

return new B();

}\*/

/\*@Override

protected C method(B b) throws EB { //OK

return new C();

}\*/

/\*@Override

protected A method(B b) throws EB { //Compilation fails. The return type is incompatible with Parent.method(B)

return new A;

}\*/

/\*@Override

protected void method(B b) throws EB { //Compilation fails. The return type is incompatible with Parent.method(B)

return ;

}\*/

/\*@Override

protected String method(B b) throws EB {//Compilation fails. The return type is incompatible with Parent.method(B)

return "abc";

}\*/

/\*@Override

protected int primitiveMethod(int i) {//OK

return 0;

}\*/

/\*@Override

protected long primitiveMethod(int i) {//Compilation fails. The return type is incompatible with Parent.primitiveMethod(int)

return 0;

}\*/

/\*@Override

protected short primitiveMethod(int i) {//Compilation fails. The return type is incompatible with Parent.primitiveMethod(int)

return 0;

}\*/

**//For checked exception, only the same or subclass of exception should be thrown**

/\*@Override

protected B method(B b) throws EB { //OK

return new B();

}\*/

/\*@Override

protected B method(B b) throws EC { //OK

return new B();

}\*/

/\*@Override

protected B method(B b) throws EA { //Compilation fails. Exception EA is not compatible with throws clause in Parent.method(B)

return new B();

}\*/

/\*@Override

protected B method(B b) throws Exception { //Compilation fails. Exception Exception is not compatible with throws clause in Parent.method(B)

return new B();

}\*/

/\*@Override

protected B method(B b) { //OK (Throwing no exception is allowed)

return new B();

}\*/

/\*@Override

protected B method(B b) throws IllegalArgumentException { //OK (Throwing any new Runtime Exception is allowed)

return new B();

}\*/

/\*@Override

protected B method(B b) throws EB, IllegalArgumentException { //OK (Throwing any new Runtime Exception is allowed)

return new B();

}\*/

/\*@Override

protected B method(B b) throws EC, IllegalArgumentException { //OK (Throwing any new Runtime Exception is allowed)

return new B();

}\*/

/\*@Override

protected B method(B b) throws EA, IllegalArgumentException { //Compilation fails. Exception EA is not compatible with throws clause in Parent.method(B)

return new B();

}\*/

**//For unchecked exception, previous rule doesn’t apply**

/\*@Override

protected B exceptionMethod(B b) throws IllegalArgumentException { //OK

return new B();

}\*/

/\*@Override

protected B exceptionMethod(B b) throws Exception{ //Compilation fails. Exception Exception is not compatible with throws clause in Parent.exceptionMethod(B)

return new B();

}\*/

/\*@Override

protected B exceptionMethod(B b) { //OK (Throwing no exception is allowed)

return new B();

}\*/

/\*@Override

protected B exceptionMethod(B b) throws NumberFormatException{ //OK (Throwing any new Runtime Exception is allowed)

return new B();

}\*/

/\*@Override

protected B exceptionMethod(B b) throws IllegalArgumentException, NumberFormatException{ //OK (Throwing any new Runtime Exception is allowed)

return new B();

}\*/

**//Static methods cannot be overriden**

/\*@Override

protected B staticMethod(B b) { //Compilation Fails. The method staticMethod(B) of type Child must override or implement a supertype method

return new B();

}\*/

}