<<singleton>> TestGraph - graph : TestGraph - classesUnderTest : Set<TestClass> - testMethods : Map<Method,TestMethod> + getInstance(): TestGraph + addClass(testClass : TestClass) + getDescriptionForClass(testClass : TestClass) + runClass(testClass : TestClass) - detectCycles(method : Collection<TestMethod>) - collectMethods(testClass : TestClass) : Map<Method,TestMethod> validate(methodsUnderTest : Set<Method>, testClass : TestClass) MethodValidator - testMethods : Set<Method> - fErrors: List<Throwable> - testClass : TestClass + validateMethodsForComposedRunner : List<Throwable> validateNoArgConstructor()

CycleDetector

MethodCollector

- notVisited : List<TestMethod>
- visited : List<TestMethod>

- testClass : TestClass

- newMethods : Set<Method>

- methods : Map<Method, TestMethod>

+ collectTestMethods(): Map<Method,TestMethod>

- done : List<TestMethod>
- + hasCycle() : boolean

- validateTestMethods()
- validateDependencies()

DependencyValidator

- fErrors: List<Throwable>

+ dependencyIsValid(method: Method, dependencies: Method[]): List<Throwable>