## Klasse Babysteps:

Der Klasse Babysteps wird die Anzahl der Sekunden für den Timer, der Controler des Fensters zum zurücksetzen des Codes und das Label zum Anzeigen des Countdowns, übergeben.

Methoden:

* void Start():  
  Starten des Timer. Der Timer ruft einmal pro Sekunde die geerbte Methode void Run von dem Task-Objekt auf.
* void Stop():  
  Stopen des Timers
* boolean Running():  
  gibt an, ob der Timer noch läuft

Konstruktor:

Der Konstruktor erstellt das Task-Objekt (erbt von TimerTask), dass dem Timer übergeben wird bei void Start(). Dem Task-Objekt, werden die Sekunden, der Controler und das Label übergeben.

## Klasse Task

Der Klasse Babysteps wird die Anzahl der Sekunden für den Timer, der Controler des Fensters zum zurücksetzen des Codes und das Label zum Anzeigen des Countdowns, übergeben.

Die Klasse erbt von TimerTask, damit diese für einen Timer genutzt werden kann.

Die Klasse ist nur für die Klasse Babysteps sichtbar.

Methoden:

* void run():  
  Zählt die Sekunden um eins runter und schreibt die Sekunden für den Nutzer sichtbar in das Label auf dem Fenster oben rechts. Sind die Sekunden runter auf 0 gezählt, wird die Methode void stepBack() des Controlers ausgeführt, damit der Nutzer nicht mehr im Code schrieben kann, sondern die Tests ändern kann.

## Klasse CompilerHelper

Diese Klasse soll helfen mit der Bibliothek Virtual-Kata von Dr. Jens Bendisposto zu arbeiten.  
Dafür stellt diese Klasse mehrere Methoden zur Verfügung, die eine Überprüfung von Kompilierfehlern für die Tests oder den Quellcode und zur Feststellung, von welcher Testklasse Tests nicht bestanden wurden.

Methoden

* void SetFeatureTest(String feautureTestClassName, String featureTestClassSource)  
  setzen der Klasse des Akzepttanztest
* void SetTest(String testClassName, String testClassSource)  
  setzen der Klasse für normale Tests im Ablauf
* void AddSourceClass(String className, String classSource)  
  setzten der Klasse, der den Code zum löse der Tests enthält
* void CompileAndTest()  
  Kompiliert die Klassen (auch die Testklassen) die dem CompilerHelper bereits übergeben wurden
* TestResult GetTestResult()  
  übergibt das Objekt der Ergebnisse für die Tests  
  TestResult ist eine Klasse der Bibliothek „virtual kata“
* CompilerResult GetCompilerResult()  
  übergibt das Objekt der Ergebnisse für das Kompilieren aller Klassen  
  CompilerResult ist eine Klasse der Bibliothek „virtual kata“
* Boolean HasCompilerErrors()  
  prüft nach ob Fehler beim Kompilieren aufgetreten sind und übergibt dann das Ergebnis
* String GetSourceClassCompilerError()  
  übergibt den Fehler der beim Kompilieren des Codes zum Erfüllen der Tests aufgetreten sind   
  Nutzt GetCompilerError(…)
* String GetTestClassCompilerError()  
  Übergibt den Fehler der beim Kompilieren, der Testklasse aufgetreten sind   
  Nutzt GetCompilerError(…)
* String GetFeatureTestClassClassCompilerError()  
  Übergibt den Fehler der beim Kompilieren, der Akzeptanztestklassse aufgetreten sind  
  Nutzt GetCompilerError(…)
* Private String GetCompilerError(CompilationUnit cu)  
  CompilationUnit ist eine Klasse der Bibliothek „virtual kata“  
  Sammelt alle Kompilierfehler in einen String für die genannte Klasse in dem Objekt cu
* String GetCompilerErrors()  
  gibt alle Fehler zurück, die beim Kompilieren aufgetreten sind, von allen Klassen
* String GetTestFaillures()  
  sammelt von allen Tests, die fehlgeschlagen sind, die Message von junit und gibt sie zurück
* int NumberOfSucceddfulTests()  
  gibt die Anzahl der nicht fehlgeschlagenen Tests zurück
* int NumberOfFailedTests()  
  gibt die Anzahl der nicht bestandenen Tests aus der Testklasse zurück  
  Nutzt CountFailedFeatureTest(…)
* int NumberOfFailledFeatureTest()  
  gibt die Anzahl der nicht bestandenen Tests aus der Testklasse zurück   
  Nutzt CountFailedFeatureTest(…)
* private int CountFeiledFeatureTest(CompilationUnit cu)   
  CompilationUnit ist eine Klasse der Bibliothek „virtual kata“  
  zählt alle fehlgeschlagenen Tests für die genannte TestKlasse in dem Objekt cu