# Portierungsbedingte Änderungen in EventsExample

## Importmappings in allen betroffenen Klassen

* Java.util.concurrent.TimeUnit -> def.TimeUnit (auch static imports)
* Java.util.Observable -> def.Observable
* Java.util.Observer -> def.Observer
* Java.lang.System -> def.System (für nanoTime(), currentTimeMillis(), getProperty(“line.separator”))
* Java.util.Random -> def.Random
* Java.io.File -> def.File (für separator)
* Java.awt.Color -> def.Color [COLOR]
* Java.lang.Integer -> def.Integer (für toHexString(int))

## Änderungs-Tags

* Abgeändert, da nur für prozessbasierte Simulation **[PROCESS]**
* Abgeändert, um Abhängigkeiten zu Swing/AWT zu vermeiden **[GUI]**
* Abgeändert um Abhängigkeiten zu Apache Commons Collections zu vermeiden **[AC\_COLLECTION]**
* Abgeändert um Abhängigkeiten zu Elementen außerhalb der Teilfunktionalität zu vermeiden **[FRAGMENT]**
* Abgeändert um JSweet/TypeScript/JavaScript-Eigenheiten umzusetzen **[JSWEET]**
* Abgeändert um Java Reflection zu vermeiden **[REFLECT]**

## Refactorings

* Simulator.EventTreeList: TreeList von Apache Commons durch ArrayList ersetzt [AC\_COLLECTION]
* Simulator.Model: ReferenceMap durch HashMap ersetzt [AC\_COLLECTION]
* Abhängigkeiten zu Threads und Quasar entfernt [PROCESS] in:
  + Experiment, Scheduler, Event, ExternalEvent, EventOf2Entities, EventOf3Entities, Entity, Schedulable, Model, ExternalEventStop, ModelComponent, QueueListStandard
* Simulator.Experiment: Erzeugung der ResourceDB aus setupExperiment() entfernt + Attribut, Getter [PROCESS]
* Report.ReportMultRowsFileOut: Abhängigkeiten zu nicht im Teilfragment vorhandenen Reportern auskommentiert [FRAGMENT]
* Simulator.Model: Methode description() zu modelDescription() umbenannt [JSWEET]
* Dist.Distribution: newInstance()-Aufruf in Konstruktor und reset() refactored [JSWEET]
* Simulator.Condition: *primitive*.class statt *PrimitiveWrapper*.TYPE in Konstruktor und getType() [JSWEET]
* Simulator.ModelCondition: *primitive*.class statt *PrimitiveWrapper*.TYPE in Konstruktor und getType() [JSWEET]
* Simulator.Scheduler: System.nanoTime() über Candy-Adapter
* Simulator.RealTimeEventWrapper: System.nanoTime() über Candy-Adapter
* Simulator.Experiment: System.nanoTime() über Candy-Adapter [JSWEET]
* Dist.DistributionManager: MersenneTwisterRandomGeneratorJS verwendet [RANDOM]
* Dist.NumericalDist: NumberValueHelper statt Number.double/longValue() [JSWEET]
* Simulator.QueueListRandom: Random mit autoseed() erstellen statt Java Random() [RANDOM]
* Simulator.QueueListRandom: insert(): nextIntWithBound(int) statt Java nextInt(int) [RANDOM]
* Simulator.RandomizingEventTreeList: Random mit autoseed() erstellen statt Java Random() [RANDOM]
* Simulator. RandomizingEventTreeList: insert(): nextIntWithBound(int) statt Java nextInt(int) [RANDOM]
* Simulator.ModelComponent und NamedObject: clone()-Methode hinzugefügt, durchreichen von Schedulable zu Object [JSWEET]
* Simulator.Experiment: ExpProgressBar-Code in proceed() in Methode showProgressBar() extrahiert und durch not yet implemented ersetzt um Abhängigkeiten zu Swing und AWT zu vermeiden [GUI]
* Simulator.Experiment: Klassen-Validitätcheck in setRandomNumberGenerator(Class) und setEventList(Class) in Methoden checkRandomGeneratorClassValidity(Class) und checkEventListClassValidity(Class) extrahiert [REFLECT]
  + Reimplementation der Validität in JS nötig
* Simulator.TimeInstant: getTimeAsCalendar auskommentiert, stattdessen getTimeAsDate nutzen [TIME]
* Simulator.TimeInstand: getBeginOf(TimeUnit) reimplementiert mit def.js.Date [Time]
* Simulator.EventTreeList: Casten über Object zu EventNote in removeFirst() [JSWEET]
* Report.TableOutput: File.separator über def.File, hardgecoded auf “/” [JSWEET]
* Report.TableOutput: System.property Aufruf in createFileName() durch Standardwert ersetzt [JSWEET]
* Report.HtmlTableFormatter: Import von Date statt fully qualified name in close() [JSWEET]
* Report.SimulationRunReporter: DecimalFormat entfernt [JSWEET]
* Simulator.Reportable: Erzeugungsversuch von custom reporters in getReporter() in Methode tryInstatiatingCustomReporter() extrahiert und mit Kommentar ersetzt [REFLECT]
  + Versuch wird übersprungen, nutzt immer default reporter
* Simulator.Schedulable.DummyReporter: createDefaultReporter()-Methodenkörper auskommentiert, immer Standard-Reporter zurückgeben um Reflection zu vermeiden [REFLECT]
* Simulator.Experiment: Deprecated Methode randomizeConcurrentEvents() auskommentiert [JSWEET], SortedMapEventList entfernt
* Simulator.QueueListStandard: ArrayList statt LinkedList [JSWEET]
  + Methodenanpassungen auch in QueueListFifo, QueueListLifo [JSWEET]
* Simulator.Entity: equals-Methode die zu Object weiterreicht hinzugefügt, da JSweet sonst Probleme hat in QueueListStandard equals an <E extends Entity> [JSWEET]
* Simulator.SingleUnitTimeFormatter: EnumMaps timeConstants auskommentiert, da nicht genutzt und JSweet keine EnumMaps unterstützt [JSWEET]
* Simulator.SingleUnitTimeFormatter: Da StringBuffer.insert(int, String) nicht unterstützt wird, wird in buildSingleUnitTimeString(long) append() genutzt [JSWEET]
* Simulator.Scheduler: ReentrantLock und Condition entfernt, LinkedBlockingQueue durch Queue/LinkedList ersetzt, da Nebenläufigkeit nicht auftritt und die Klassen nicht von JSweet unterstützt werden [CONCURRENT]
* Simulator.QueueListStandard: implements Interface PropertyChangeListener und dazugehörige Methode propertyChange entfernt [PROCESS]
  + Implements auch entfernt in QueueList, QueueListRandom, QueueListLifo, QueueListFifo
* Simulator.Experiment: Rückgabewert von getOutputPath() geändert (ohne File) [JSWEET]
* Report.JavaScriptFormatter: def.Integer für toHexString() in intToHexString() [JSWEET]
* Report.HtmlTableChartFormatter: def.Integer für toHexString() in intToHexString() [JSWEET]
* Report.HtmlTableChartFormatter: Import von Date statt fully qualified name in close() [JSWEET]
* Report.html5.AbstractChartDataTable: Hardgecoded (1d/0d) verwendet statt Double.POSITIVE\_INFINITY in getHighestDataValue() [JSWEET]
* Report.html5.AbstractChartDataTable: NumberValueHelper statt Number.double in getHighestDataValue() [JSWEET]
* Report.FileOutput: def.System für getProperty(„line.separator“) [JSWEET]
* Report.FileOutput: BufferedWriter entfernt, Writer durch def.OutputWriter ersetzt [FILE]
  + Aufrufe zu JDK-Writern müssen über OutputWriter weitergeleitet werden, Unterklasse von OutputWriter implementieren
  + Zu Testzwecken ConsoleOutputWriter
  + Änderung des Standard-OutputWriters zu ConsoleOutputWriter in NORMAL\_FILE\_ACCESS
* Report.FileSystemAccess: OutputWriter als Rückgabewert [FILE]
* Def.ConsoleOutputWriter: Neue Klasse zum Schreiben des Outputs in die Konsole