Main:

Controller.runAlgorithm

Controller:

void **runAlgorithm**

* Weist jedem Job eine ID zu
* placeJobsFirstPlate aufrufen für Platte 1
* showBenchmarkResults aufrufen für Platte 1
* generateFollowUpPlates aufrufen für Platten 2, 3 etc.
* showSummaryOfJobSets aufrufen für die Ausgabe der besten Lösungenvorschläge

PlateRunResult **placeJobsFirstPlate**

* Schleife: Ruft für jeden einzelnen Job Algorithm.placeJob auf
* Job-Sets für die failed Jobs erstellen mit buildUnplacedJobGroups. In BenchmarkResult mit buildBenchmarkResults speichern.
* Labelt jeden Path mit ihrem ihrem Job-Set etc.

void **generateFollowUpPlates**

* Schleife solange Area & Edge Job-Sets leer sind: Erstellt für jedes Job-Set eine neue Platte und ruft processJobSetGroups auf.

void **processJobSetGroups**

* Schleife für jedes Job-Set:
  + placeLeftoverJobs aufrufen
  + Variablen für BenchmarkVisualizer setzen und Benchmark öffnen mit showBenchmarkResults
  + Benchmark für Platte 2 durchgehen und die besten Ergebnisse speichern (TODO: auslagern)
  + Job-Sets für die failedJobs erstellen mit buildUnplacedJobGroups.
  + failedJobs, rootSetId und parentSetid für die nächste iteration (nächte Platte) von generateFollowUpPlates speichern.

JobSetRunResult **placeLeftoverJobs**

* Schleife: Ruft für jeden einzelnen Job Algorithm.placeJob auf
* In BenchmarkResult mit buildBenchmarkResults speichern.