Documentation

Class Algoritmo

heuristicasimple.Algoritmo

Summary

Nested Class Summary

Field Summary

|  |  |
| --- | --- |
| **Modifier and type** | **Field and Description** |
| private int[][] | **Requirements** |
| private int[][] | **ProductionPlan** |
| private int[][] | **Inventory** |
| private int[] | **LeftoverAfterPeriod** |
| private double | **AccumRealCost** |
| private final double[] | **RelSCk** |
| private final double[] | **RelOCk** |
| private final int | **lengthPeriod** |
| private final int[][] | **Nik** |
| private final int[][] | **Mik** |
| private final int[][] | **Dmt** |
| private final double[] | **AccumSCk** |
| private final double[] | **AccumOCk** |
| private final int[] | **MainProducts** |
| private final int[] | **LTk** |
| private final int[] | **invIn** |

Constructor Summary

|  |
| --- |
| Constructor and Description |
| **Algoritmo**(int[][] Nik, int[][] Mik, int[][] Dmt, double[] SCk, double[] OCk, int[] MainProducts, int[] LTk, int[] invIn)  Constructs an Algoritmo that is initialized with:  Nik as the matrix of stroke input with i products and k strokes  Mik as the matrix of stroke output with i products and k strokes  Dmt as the matrix of demand of m **MainProducts** in the t time  SCk as the vector of relative setup costs of every k stroke  Ock as the vector of relative operation costs of every k stroke  MainProducts as the vector of names of the final products  Ltk as the vector of lead times of every k stroke  invIn as the vector of initial inventory of every product. |

Method Summary

|  |  |
| --- | --- |
| Modifier and type | Field and Description |
| public void | **ProductionRequirement**(int numProdFin) |
| public int[][] | **ApplyLotification** (int[][] req, int PeriodLength) |
| public int[] | **ApplyLotification** (int[] req, int PeriodLength) |
| public void | **setProductionPlan**( int lot) |
| public int[] | **getRelProductionPlan**( int lot\_mod, int[] requirement) |
| int[][] | **calcStrokeStroke**() |
| int[][][] | **calcAnd\_OrMatrix**() |
| public ArrayList<Node> | **getRelMainStrokes2Node**(int MPN, int MPA) |
| int[] | **getRelMainStrokes** (int MPN) |
| int[][] | **getMainStrokes** () |
| double | **getRelSCk**(int stroke) |
| double | **getRelOCk**(int stroke) |
| public double | **getAccumSC**(**Node** relative) |
| public double[] | **getAccumSCTable**() |
| public double | **getAccumOC**(**Node** relative) |
| public double[] | **getAccumOCTable**() |
| public int | **chooseLotificationModel**() |
| public int[] | **getRelLLeadTimes**(int MainProduct) |
| public int[][] | **getUnitsTable**(int MainProduct, int[] requirement) |
| public int | **getBestStroke**(double[] SC, double[] OC, int[][] UT, int MSK) |
| public int | **getBestStroke1T**(double[] SC, double[] OC, int[][] UT, int MPN |
|  |  |
| public Map | **productANDMaps**(Map<Integer,int[]> child) |
| public int[] | **updateInventory**(int[] currentRequirements, int MainProductName) |
| public int | **getProdfromStrk**(int strkKey) |
| public Map<Integer,int[]> | **productORMaps**(Map<Integer,int[]> child) |
| public Map<Integer,Double> | **AverageORamountsSC**(Map<Integer,int[]> child) |
| public Map<Integer,int[]> | **SeparateORbyProduct**(Map<Integer,int[]> child) |
| public Map<Integer,Double> | **AverageORamountsOC**(Map<Integer,int[]> child) |
| public void | **accumulateRealCost**(int strokeKey, int amount) |
| public int[] | **getRelRequirements**(int MainProductName, int[] previousRequirements, int MainNodeKey) |
| public List | **Heuristic**(**Node** current, List<Integer> usedStrokes, int MainNodeKey, int MainProductAmount, int[] previousRequirements) |
| public List | **InitialHeuristic**(int MainProductName, int MainProductAmount) |
| public List<List> | **ApplyHeuristic**() |
| public void | **PrintParticularResults**(int name, int amount) |
| public void | **PrintGeneralResults**() |
| public int[][] | **getProductionPlan**() |

Field Detail

|  |
| --- |
|  |
|  |

Constructor Detail

|  |
| --- |
|  |
|  |

Method Detail

|  |
| --- |
|  |
|  |