KnapsackProblem-OOP-UML

Benjamin Kataliko | January 14, 2021

Knapsack

- capacity : int
- items : List<items>

+ addItem(o : Item) : void
+ removeItem(o : Item) : Item
+ getCurrentWeight(): int
+ getCurrentValue(): int
+ getStringRepr(): String

Item

- value : int
- weight : int
- repr : String

+ getRepresentation(): String
+ getValue(): int
+ getWeight(): int

1..*

1

*

1

BruteForce

- capacity : int
- overflow : boolean
- currentValue : int
- maxValue : int
- currentWeight : int

+ bruteForce(capacity : int, availableItems : List<Item>, file: File) : Knapsack
- writeSolutionInFile(file: File, sac: Knapsack) : void

1

1..*

1

DynamicProgramming

- KTable : Knapsack[ ][ ]

+ dynamicProgramming(capacity : int, availableItems : List<Item>, file: File) : Knapsack
- writeSolutionInFile(file: File, sac: Knapsack) : void

1

1

KnapsackProblem

+ main(args: String[]) : void

1

1