Package com.smartplanner.model

Class Summary

Class	Description
GoToOptimizedActivityDecider	Generator for decision points (that tells us if we should go to work) GoToOptimizedActivityDecider can't be reused, what means that you have to create new object each time you want to get all of combinations.
LessonPicker	Generator for possible picks of terms for each lesson LessonPicker can't be reused, what means that you have to create new object each time you want to get all of combinations.
LessonWithPossibleTerms	Binder for a Lesson object with an array of possible terms for the lesson
OptimalityCalculator	Calculator for time spent on optimized activity.
SmartPlanner	Finder of optimal timetable that allows to spent maximum possible time doing optimized activity.
TimeDistanceManager	Wrapper for commute matrix, which is a two dimensional array where commuteMatrix.get(id1).get(id2) yields amount of time that is needed to travel from activity with id1 to activity with id2
TimetableEntry	Binder of lesson and picked term.
TimetableValidator	Validator of timetables.
TimetableWithDecisionPointsAndScore	Holder of information about timetable.

Package com.smartplanner.model

Class GoToOptimizedActivityDecider

java.lang.Object
    com.smartplanner.model.GoToOptimizedActivityDecider

```
public class GoToOptimizedActivityDecider
extends java.lang.Object
```

Generator for decision points (that tells us if we should go to work) GoToOptimizedActivityDecider can't be reused, what means that you have to create new object each time you want to get all of combinations. After getNext() returns false, the object is useless.

Constructor Summary

Constructors

Constructor	Description
`GoToOptimizedActivityDecider(java.util.ArrayList<TimetableEntry> singleDayTimetable)`	Creates decider based on single day timetable
`GoToOptimizedActivityDecider(java.util.ArrayList<TimetableEntry> completeTimetable, int getDecisionsForCycleNumberDay)`	Creates decider based on complete timetable.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`java.util.ArrayList<java.lang.Boolean>`	`getNext()`	Returns next combination of decisions
`boolean`	`isNext()`	Checks if there is next combination available.

Methods inherited from class java.lang.Object

Constructor Detail

GoToOptimizedActivityDecider

```
public GoToOptimizedActivityDecider(java.util.ArrayList<TimetableEntry> singleDayTimetable)
```

Creates decider based on single day timetable

Parameters:

`singleDayTimetable` - a timetable schedule for particular day

GoToOptimizedActivityDecider

```
public GoToOptimizedActivityDecider(java.util.ArrayList<TimetableEntry> completeTimetable,
                                    int getDecisionsForCycleNumberDay)
```

Creates decider based on complete timetable. Since the decider needs a timetable for single day, one have to provide number of the day in cycle as a second argument

Parameters:

`completeTimetable` - a complete timetable

`getDecisionsForCycleNumberDay` - specifies a particular day for computations, based on this argument getNext() method will return decisions for the day passed in this argument

Method Detail

getNext

```
public java.util.ArrayList<java.lang.Boolean> getNext()
```

Returns next combination of decisions

Returns:

ArrayList of decision points generated using brute force method. returnedTable.get(n) equal to true means that after n-th lesson we should go to optimized activity

```
public boolean isNext()
```

Checks if there is next combination available.

Returns:

true if there is another combination from brute force method, false else

Package com.smartplanner.model

Class LessonPicker

java.lang.Object
    com.smartplanner.model.LessonPicker

```
public class LessonPicker
extends java.lang.Object
```

Generator for possible picks of terms for each lesson LessonPicker can't be reused, what means that you have to create new object each time you want to get all of combinations. After getNext() returns false, the object is useless.

Constructor Summary

Constructors

Constructor	Description
`LessonPicker (java.util.ArrayList<LessonWithPossibleTerms> lessons, int daysInCycle)`	Creates lesson picker and sets it up for specified set of lessons and possible terms.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`java.util.ArrayList<TimetableEntry>`	`getNext()`	Returns next combination of term picks
`boolean`	`isNext()`	Checks if there is next combination available.

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

```
public LessonPicker(java.util.ArrayList<LessonWithPossibleTerms> lessons,
                    int daysInCycle)
```

Creates lesson picker and sets it up for specified set of lessons and possible terms.

Parameters:

`lessons` - array of all lists with possible terms that can be picked

`daysInCycle` - number of days after which the plan will repeat (is equal to max(LessonWithPossibleTerms.repeatingPeriod) rounded up to a number that is multiple of 7 (the amount of days in a week)

Method Detail

isNext

```
public boolean isNext()
```

Checks if there is next combination available.

Returns:

returns true if there is another combination, false else

getNext

```
public java.util.ArrayList<TimetableEntry> getNext()
```

Returns next combination of term picks

Returns:

returns next combination of picked terms

Package com.smartplanner.model

Class LessonWithPossibleTerms

java.lang.Object
    com.smartplanner.model.entity.Lesson
        com.smartplanner.model.LessonWithPossibleTerms

```
public class LessonWithPossibleTerms
extends Lesson
```

Binder for a Lesson object with an array of possible terms for the lesson

Constructor Summary

Constructors

Constructor	Description
`LessonWithPossibleTerms()`	

Method Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

LessonWithPossibleTerms

```
public LessonWithPossibleTerms()
```

Package com.smartplanner.model

Class OptimalityCalculator

java.lang.Object
    com.smartplanner.model.OptimalityCalculator

```
public class OptimalityCalculator
extends java.lang.Object
```

Calculator for time spent on optimized activity.

Constructor Summary

Constructors

Constructor	Description
`OptimalityCalculator(TimeDistanceManager timeDistanceManager, int maxCommutesPerDay, int minTimeSpentOnOptimizedAtOnceInMinutes, int numberOfDaysInCycle, OptimizedActivity optimizedActivity)`	Creates OptimalityCalculator.

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`TimetableWithDecisionPointsAndScore`	`calculate(java.util.ArrayList<TimetableEntry> timetable)`	Calculates amount of time (in minutes) spent on optimized activity for provided timetable.

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

OptimalityCalculator

```
public OptimalityCalculator(TimeDistanceManager timeDistanceManager,
                            int maxCommutesPerDay,
                            int minTimeSpentOnOptimizedAtOnceInMinutes,
                            int numberOfDaysInCycle,
                            OptimizedActivity optimizedActivity)
```

Creates OptimalityCalculator.

Parameters:

`timeDistanceManager` - object that contains commute matrix, which is data about travel time between each lesson and work

`maxCommutesPerDay` - maximal amount of commutes to work per day (specified by a user)

`minTimeSpentOnOptimizedAtOnceInMinutes` - minimal amount of time in session that user wants to spent on optimized activity if he decides to start it

`numberOfDaysInCycle` - number of days after which the plan will repeat

`optimizedActivity` - an object that contains data about optimized activity

Method Detail

calculate

```
public TimetableWithDecisionPointsAndScore calculate
(java.util.ArrayList<TimetableEntry> timetable)
```

Calculates amount of time (in minutes) spent on optimized activity for provided timetable.

Parameters:

`timetable` - complete timetable that first should pass the validation done by TimetableValidator

Returns:

amount of minutes spent on optimized activity based on provided timetable

Package com.smartplanner.model

Class SmartPlanner

java.lang.Object
    com.smartplanner.model.SmartPlanner

```
public class SmartPlanner
extends java.lang.Object
```

Finder of optimal timetable that allows to spent maximum possible time doing optimized activity.

Constructor Summary

Constructors

Constructor	Description
`SmartPlanner(java.util.List<LessonWithPossibleTerms> lessons, int daysInCycle, TimeDistanceManager distanceManager, int maxCommutesPerDay, OptimizedActivity optimizedActivity)`	Creates SmartPlanner that finds the most optimal plan based on passed arguments

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`TimetableWithDecisionPointsAndScore`	`getOptimalPlan()`	Returns the optimal plan that is calculated based on arguments passed in constructor

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

```
public SmartPlanner(java.util.List<LessonWithPossibleTerms> lessons,
                    int daysInCycle,
                    TimeDistanceManager distanceManager,
                    int maxCommutesPerDay,
                    OptimizedActivity optimizedActivity)
```

Creates SmartPlanner that finds the most optimal plan based on passed arguments

Parameters:

`lessons` - list of all lessons that contains among others possible terms for each lesson.

`daysInCycle` - amount of days, after which the whole plan will repeat

`distanceManager` - object that contains commute matrix, which is data about travel time between each lesson and work

`maxCommutesPerDay` - maximal amount of commutes to work per day (specified by a user)

`optimizedActivity` - object containing data about optimized activity (for example work)

Method Detail

getOptimalPlan

```
public TimetableWithDecisionPointsAndScore getOptimalPlan()
```

Returns the optimal plan that is calculated based on arguments passed in constructor

Returns:

if provided data is valid it returns specified terms for each lesson, amount of time spent and decision points(that tells us if we should go to work after each lesson. If data for computations is not valid returns object with amount of minutes spent on optimized activity equal to 0 and every other field set to null

Package com.smartplanner.model

Class TimeDistanceManager

java.lang.Object
    com.smartplanner.model.TimeDistanceManager

```
public class TimeDistanceManager
extends java.lang.Object
```

Wrapper for commute matrix, which is a two dimensional array where commuteMatrix.get(id1).get(id2) yields amount of time that is needed to travel from activity with id1 to activity with id2

Constructor Summary

Constructors

Constructor	Description
`TimeDistanceManager (java.util.List<java.util.List<java.lang.Integer>> timeDistanceInMinutes)`	Creates TimeDistanceManager based on commute matrix which is two dimensional array, where commuteMatrix.get(id1).get(id2) yields amount of time that is needed to travel from activity with id1 to activity with id2

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`int`	`getTimeDistanceInMinutes(Lesson from, Lesson to)`	Returns time needed to travel between points passed as arguments
`int`	`getTimeDistanceInMinutes(Lesson from, OptimizedActivity to)`	Returns time needed to travel between points passed as arguments
`int`	`getTimeDistanceInMinutes (OptimizedActivity from, Lesson to)`	Returns time needed to travel between points passed as arguments

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

TimeDistanceManager

```
public TimeDistanceManager(java.util.List<java.util.List<java.lang.Integer>> timeDistanceInMinutes)
```

Creates TimeDistanceManager based on commute matrix which is two dimensional array, where commuteMatrix.get(id1).get(id2) yields amount of time that is needed to travel from activity with id1 to activity with id2

Parameters:

`timeDistanceInMinutes` - commute matrix where commuteMatrix.get(id1).get(id2) yields amount of time that is needed to travel from activity with id1 to activity with id2

Method Detail

getTimeDistanceInMinutes

```
public int getTimeDistanceInMinutes(Lesson from, Lesson to)
```

Returns time needed to travel between points passed as arguments

Parameters:

`from` - lesson for which you want to get information about travel time (traveling time from the lesson)

`to` - lesson for which you want to get information about travel time (traveling time to the lesson)

Returns:

amount of time that is needed to travel from lesson "from" to lesson "to"

getTimeDistanceInMinutes

```
public int getTimeDistanceInMinutes(OptimizedActivity from, Lesson to)
```

Returns time needed to travel between points passed as arguments

Parameters:

`from` - optimized activity for which you want to get information about travel time (traveling time from the activity)

`to` - lesson for which you want to get information about travel time (traveling time to the lesson)

Returns:

amount of time that is needed to travel from optimized activity "from" to lesson "to"

getTimeDistanceInMinutes

`from` - lesson for which you want to get information about travel time (traveling time from the lesson)

`to` - optimized activity for which you want to get information about travel time (traveling time to the activity)

Returns:

amount of time that is needed to travel from lesson "from" to optimized activity "to"

OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

ALL CLASSES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

Package com.smartplanner.model

Class TimetableEntry

java.lang.Object
    com.smartplanner.model.TimetableEntry

```
public class TimetableEntry
extends java.lang.Object
```

Binder of lesson and picked term. Represents a particular entry in timetable.

Constructor Summary

Constructors

Constructor	Description
`TimetableEntry(Lesson lesson, Term pickedTerm)`	Creates an object that binds lesson with particular term

Method Summary

All Methods   Instance Methods   Concrete Methods

Modifier and Type	Method	Description
`boolean`	`equals(java.lang.Object other)`	
`Lesson`	`getLesson()`	Returns lesson stored in object
`java.lang.String`	`getName()`	Returns name of the lesson stored in object
`Term`	`getTerm()`	Returns term stored in object

Methods inherited from class java.lang.Object

`clone, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

```
public TimetableEntry(Lesson lesson,
                      Term pickedTerm)
```

Creates an object that binds lesson with particular term

Parameters:

`lesson` - the lesson for which one wants to specify a term

`pickedTerm` - particular term

Method Detail

getTerm

```
public Term getTerm()
```

Returns term stored in object

Returns:

term picked for the lesson

getName

```
public java.lang.String getName()
```

Returns name of the lesson stored in object

Returns:

name of the lesson

getLesson

```
public Lesson getLesson()
```

Returns lesson stored in object

equals

```
public boolean equals(java.lang.Object other)
```

Overrides:

`equals` in class `java.lang.Object`

Package com.smartplanner.model

Class TimetableValidator

java.lang.Object
    com.smartplanner.model.TimetableValidator

```
public class TimetableValidator
extends java.lang.Object
```

Validator of timetables. It checks if lessons doesn't overlaps each other.

Constructor Summary

Constructors

Constructor	Description
`TimetableValidator(TimeDistanceManager distanceManager)`	Creates validator for timetables

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`boolean`	`isValid(java.util.ArrayList<TimetableEntry> timetable)`	Checks if the timetable provided as an argument is valid (i.e.

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

Creates validator for timetables

Parameters:

`distanceManager` - object that contains commute matrix, which is data about travel time between each lesson and work

Method Detail

isValid

```
public boolean isValid(java.util.ArrayList<TimetableEntry> timetable)
```

Checks if the timetable provided as an argument is valid (i.e. none of entries overlaps each other)

Parameters:

`timetable` - timetable to be validated

Returns:

true if timetable is valid, false else

Package com.smartplanner.model

Class TimetableWithDecisionPointsAndScore

java.lang.Object
    com.smartplanner.model.TimetableWithDecisionPointsAndScore

```
public class TimetableWithDecisionPointsAndScore
extends java.lang.Object
```

Holder of information about timetable. It stores decision points and amount of minutes spent on optimized activity.

Constructor Summary

Constructors

Constructor	Description
`TimetableWithDecisionPointsAndScore(int minutesSpentAtOptimizedActivity, java.util.ArrayList<TimetableEntry> optimalTimetable, java.util.ArrayList<java.util.ArrayList<java.lang.Boolean>> optimalDecisionPoints)`	

Method Summary

All Methods   Instance Methods   Concrete Methods

Modifier and Type	Method	Description
`int`	`getMinutesSpentAtOptimizedActivity()`	
`java.util.ArrayList<java.util.ArrayList<java.lang.Boolean>>`	`getOptimalDecisionPoints()`	Calculates optimal timetable that will maximize amount of time spent on optimized activity.
`java.util.ArrayList<TimetableEntry>`	`getOptimalTimetable()`	

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

TimetableWithDecisionPointsAndScore

```
public TimetableWithDecisionPointsAndScore(int minutesSpentAtOptimizedActivity,
                                           java.util.ArrayList<TimetableEntry> optimalTimetable,
                                           java.util.ArrayList<java.util.ArrayList<java.lang.Boolean>> optimalDecisionPoints)
```

Method Detail

getOptimalTimetable

```
public java.util.ArrayList<TimetableEntry> getOptimalTimetable()
```

getOptimalDecisionPoints

```
public java.util.ArrayList<java.util.ArrayList<java.lang.Boolean>> getOptimalDecisionPoints()
```

Calculates optimal timetable that will maximize amount of time spent on optimized activity.

Returns:

an array with decision points. First dimension is responsible for the day in cycle. Second dimension is responsible for decision point. e.g.
optimalDecisionPoints.get(0).get(0) answers question:
Should I go to work in day 0 before first activity?
optimalDecisionPoints.get(0).get(1) answers question:
Should I go to work in day 0 after first activity?
...
optimalDecisionPoints.get(0).get(n) [n = last index in array] answers question:
Should I go to work in day 0 after last activity?