

# MyTasks

Version 1.3

Generated by Doxygen 1.8.3.1

Wed May 1 2013 14:23:58



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Packages . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>Namespace Documentation</b>	<b>7</b>
4.1	Package com.cpsc481.mytasks.chart . . . . .	7
4.1.1	Detailed Description . . . . .	7
<b>5</b>	<b>Class Documentation</b>	<b>9</b>
5.1	com.cpsc481.mytasks.chart.AbstractChart Class Reference . . . . .	9
5.1.1	Detailed Description . . . . .	9
5.1.2	Member Function Documentation . . . . .	10
5.1.2.1	buildBarDataset . . . . .	10
5.1.2.2	buildBarRenderer . . . . .	10
5.1.2.3	buildCategoryDataset . . . . .	10
5.1.2.4	buildCategoryRenderer . . . . .	10
5.1.2.5	buildDataset . . . . .	11
5.1.2.6	buildDateDataset . . . . .	11
5.1.2.7	buildMultipleCategoryDataset . . . . .	11
5.1.2.8	buildRenderer . . . . .	11
5.1.2.9	setChartSettings . . . . .	12
5.2	com.cpsc481.mytasks.AddTask Class Reference . . . . .	12
5.2.1	Detailed Description . . . . .	13
5.2.2	Member Function Documentation . . . . .	13
5.2.2.1	createTask . . . . .	13
5.2.2.2	onCreate . . . . .	13
5.2.2.3	onCreateOptionsMenu . . . . .	13
5.2.2.4	showDatePicker . . . . .	14

5.2.2.5	<a href="#">showTimePicker</a>	14
5.3	<a href="#">com.cpsc481.mytasks.chart.ChartInterface Interface Reference</a>	14
5.3.1	<a href="#">Detailed Description</a>	15
5.3.2	<a href="#">Member Function Documentation</a>	15
5.3.2.1	<a href="#">execute</a>	15
5.3.2.2	<a href="#">getDesc</a>	15
5.3.2.3	<a href="#">getName</a>	15
5.3.3	<a href="#">Member Data Documentation</a>	15
5.3.3.1	<a href="#">DESC</a>	15
5.3.3.2	<a href="#">NAME</a>	16
5.4	<a href="#">com.cpsc481.mytasks.CurrentTaskListFragment.ChooseBeginOrEndFragment Class Reference</a>	16
5.4.1	<a href="#">Detailed Description</a>	16
5.5	<a href="#">com.cpsc481.mytasks.CurrentTaskListFragment Class Reference</a>	16
5.5.1	<a href="#">Detailed Description</a>	17
5.5.2	<a href="#">Member Function Documentation</a>	17
5.5.2.1	<a href="#">addTask</a>	17
5.5.2.2	<a href="#">fillData</a>	18
5.5.2.3	<a href="#">onCreate</a>	18
5.5.2.4	<a href="#">onListItemClick</a>	18
5.5.2.5	<a href="#">removeTask</a>	18
5.5.2.6	<a href="#">showDiffBegin</a>	18
5.5.2.7	<a href="#">startTask</a>	18
5.6	<a href="#">com.cpsc481.mytasks.customclasses.DateAndTime Class Reference</a>	19
5.7	<a href="#">com.cpsc481.mytasks.DueDatePickerFragment Class Reference</a>	19
5.7.1	<a href="#">Detailed Description</a>	19
5.7.2	<a href="#">Member Function Documentation</a>	19
5.7.2.1	<a href="#">onDateSet</a>	19
5.7.2.2	<a href="#">setHandler</a>	20
5.8	<a href="#">com.cpsc481.mytasks.DueTimePickerFragment Class Reference</a>	20
5.8.1	<a href="#">Detailed Description</a>	20
5.8.2	<a href="#">Member Function Documentation</a>	20
5.8.2.1	<a href="#">onTimeSet</a>	20
5.8.2.2	<a href="#">setHandler</a>	21
5.9	<a href="#">com.cpsc481.mytasks.Information Class Reference</a>	21
5.9.1	<a href="#">Detailed Description</a>	21
5.10	<a href="#">com.cpsc481.mytasks.MainActivity Class Reference</a>	21
5.10.1	<a href="#">Detailed Description</a>	22
5.10.2	<a href="#">Member Function Documentation</a>	22
5.10.2.1	<a href="#">addTaskButton</a>	22
5.10.2.2	<a href="#">getInfo</a>	22

5.10.2.3	showDiff	22
5.11	com.cpsc481.mytasks.chart.PieChart Class Reference	23
5.11.1	Detailed Description	23
5.11.2	Member Function Documentation	23
5.11.2.1	execute	23
5.11.2.2	getDesc	23
5.11.2.3	getName	24
5.12	com.cpsc481.mytasks.Reports Class Reference	24
5.12.1	Detailed Description	24
5.12.2	Member Function Documentation	25
5.12.2.1	getAdviceLine	25
5.12.2.2	showCompletionChart	25
5.12.2.3	showDifficultyChart	25
5.12.2.4	showOnTimeChart	25
5.12.2.5	timesTakenPerDifficulty	25
5.12.2.6	underOverOrNearPerDifficulty	26
5.13	com.cpsc481.mytasks.SetDifficultyTimeEstimateFragment Class Reference	26
5.13.1	Detailed Description	26
5.14	com.cpsc481.mytasks.SetTimeFragment Class Reference	27
5.14.1	Detailed Description	27
5.14.2	Member Function Documentation	27
5.14.2.1	onCreateDialog	27
5.15	com.cpsc481.mytasks.TabListener< T extends Fragment > Class Reference	27
5.15.1	Detailed Description	28
5.15.2	Constructor & Destructor Documentation	28
5.15.2.1	TabListener	28
5.16	com.cpsc481.mytasks.customclasses.Task Class Reference	28
5.16.1	Member Function Documentation	29
5.16.1.1	getDifficulty	29
5.16.1.2	getDueDate	29
5.16.1.3	getDueTime	29
5.16.1.4	getName	29
5.16.1.5	getStartDate	29
5.16.1.6	getStartTime	30
5.16.1.7	setDifficulty	30
5.16.1.8	setDueDate	30
5.16.1.9	setDueTime	30
5.16.1.10	setName	30
5.16.1.11	setStartDate	30
5.16.1.12	setStartTime	30

5.17	<a href="#">com.cpsc481.mytasks.databasestorage.TaskDatabaseHelper Class Reference</a>	30
5.17.1	<a href="#">Detailed Description</a>	32
5.17.2	<a href="#">Member Data Documentation</a>	32
5.17.2.1	<a href="#">CREATE_DIFFICULTY_TABLE</a>	32
5.17.2.2	<a href="#">CREATE_DUEDATE_TABLE</a>	32
5.17.2.3	<a href="#">CREATE_DUETIME_TABLE</a>	32
5.17.2.4	<a href="#">CREATE_ENDDATE_TABLE</a>	33
5.17.2.5	<a href="#">CREATE_ENDTIME_TABLE</a>	33
5.17.2.6	<a href="#">CREATE_MASTER_TABLE</a>	33
5.17.2.7	<a href="#">CREATE_STARTDATE_TABLE</a>	33
5.17.2.8	<a href="#">CREATE_STARTTIME_TABLE</a>	33
5.17.2.9	<a href="#">CREATE_TASKNAME_TABLE</a>	33
5.18	<a href="#">com.cpsc481.mytasks.databasestorage.TaskDataStore Class Reference</a>	34
5.18.1	<a href="#">Detailed Description</a>	34
5.18.2	<a href="#">Member Function Documentation</a>	35
5.18.2.1	<a href="#">addNewTask</a>	35
5.18.2.2	<a href="#">completeTask</a>	35
5.18.2.3	<a href="#">findAllTasks</a>	35
5.18.2.4	<a href="#">findSelectForTask</a>	35
5.18.2.5	<a href="#">findTask</a>	35
5.18.2.6	<a href="#">findTasksForDifficulty</a>	36
5.18.2.7	<a href="#">getCompletedOnTimeVsNot</a>	36
5.18.2.8	<a href="#">getDifficultyID</a>	36
5.18.2.9	<a href="#">getDueDateID</a>	36
5.18.2.10	<a href="#">getDueTimeID</a>	37
5.18.2.11	<a href="#">getEndDateID</a>	37
5.18.2.12	<a href="#">getEndTimeID</a>	37
5.18.2.13	<a href="#">getEstimatedTimeForDifficulty</a>	37
5.18.2.14	<a href="#">getStartDateID</a>	37
5.18.2.15	<a href="#">getStartTimeID</a>	38
5.18.2.16	<a href="#">getTaskNameID</a>	38
5.18.2.17	<a href="#">getTotalCompletedVsNot</a>	38
5.18.2.18	<a href="#">hasSetTime</a>	38
5.18.2.19	<a href="#">isComplete</a>	38
5.18.2.20	<a href="#">isTaskStarted</a>	39
5.18.2.21	<a href="#">startTask</a>	39
5.18.2.22	<a href="#">updateTimeForDifficulty</a>	39
5.19	<a href="#">com.cpsc481.mytasks.TaskSimpleAdapter Class Reference</a>	39
5.19.1	<a href="#">Detailed Description</a>	40

**[Index](#)****40**





# Chapter 1

## Namespace Index

### 1.1 Packages

Here are the packages with brief descriptions (if available):

<a href="#">com.cpsc481.mytasks.chart</a> . . . . .	7
---	---



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

com.cpsc481.mytasks.chart.ChartInterface . . . . .	14
com.cpsc481.mytasks.chart.AbstractChart . . . . .	9
com.cpsc481.mytasks.chart.PieChart . . . . .	23
com.cpsc481.mytasks.customclasses.DateAndTime . . . . .	19
OnDateSetListener	
com.cpsc481.mytasks.DueDatePickerFragment . . . . .	19
OnTimeSetListener	
com.cpsc481.mytasks.DueTimePickerFragment . . . . .	20
TabListener	
com.cpsc481.mytasks.TabListener< T extends Fragment > . . . . .	27
com.cpsc481.mytasks.customclasses.Task . . . . .	28
com.cpsc481.mytasks.databasesstorage.TaskDataStore . . . . .	34
Activity	
com.cpsc481.mytasks.Information . . . . .	21
DialogFragment	
com.cpsc481.mytasks.CurrentTaskListFragment.ChooseBeginOrEndFragment . . . . .	16
com.cpsc481.mytasks.DueDatePickerFragment . . . . .	19
com.cpsc481.mytasks.DueTimePickerFragment . . . . .	20
com.cpsc481.mytasks.SetTimeFragment . . . . .	27
Fragment	
com.cpsc481.mytasks.Reports . . . . .	24
com.cpsc481.mytasks.SetDifficultyTimeEstimateFragment . . . . .	26
FragmentActivity	
com.cpsc481.mytasks.AddTask . . . . .	12
ListActivity	
com.cpsc481.mytasks.MainActivity . . . . .	21
ListFragment	
com.cpsc481.mytasks.CurrentTaskListFragment . . . . .	16
SimpleCursorAdapter	
com.cpsc481.mytasks.TaskSimpleAdapter . . . . .	39
SQLiteOpenHelper	
com.cpsc481.mytasks.databasesstorage.TaskDatabaseHelper . . . . .	30



## Chapter 3

# Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">com.cpsc481.mytasks.chart.AbstractChart</a>	9
<a href="#">com.cpsc481.mytasks.AddTask</a>	12
<a href="#">com.cpsc481.mytasks.chart.ChartInterface</a>	14
<a href="#">com.cpsc481.mytasks.CurrentTaskListFragment.ChooseBeginOrEndFragment</a>	16
<a href="#">com.cpsc481.mytasks.CurrentTaskListFragment</a>	16
<a href="#">com.cpsc481.mytasks.customclasses.DateAndTime</a>	19
<a href="#">com.cpsc481.mytasks.DueDatePickerFragment</a>	19
<a href="#">com.cpsc481.mytasks.DueTimePickerFragment</a>	20
<a href="#">com.cpsc481.mytasks.Information</a>	21
<a href="#">com.cpsc481.mytasks.MainActivity</a>	21
<a href="#">com.cpsc481.mytasks.chart.PieChart</a>	23
<a href="#">com.cpsc481.mytasks.Reports</a>	24
<a href="#">com.cpsc481.mytasks.SetDifficultyTimeEstimateFragment</a>	26
<a href="#">com.cpsc481.mytasks.SetTimeFragment</a>	27
<a href="#">com.cpsc481.mytasks.TabListener&lt; T extends Fragment &gt;</a>	27
<a href="#">com.cpsc481.mytasks.customclasses.Task</a>	28
<a href="#">com.cpsc481.mytasks.databasestorage.TaskDatabaseHelper</a>	30
<a href="#">com.cpsc481.mytasks.databasestorage.TaskDataStore</a>	34
<a href="#">com.cpsc481.mytasks.TaskSimpleAdapter</a>	39



## Chapter 4

# Namespace Documentation

### 4.1 Package `com.cpsc481.mytasks.chart`

#### Classes

- class [AbstractChart](#)
- interface [ChartInterface](#)
- class [PieChart](#)

#### 4.1.1 Detailed Description

Copyright (C) 2009, 2010 SC 4ViewSoft SRL

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.



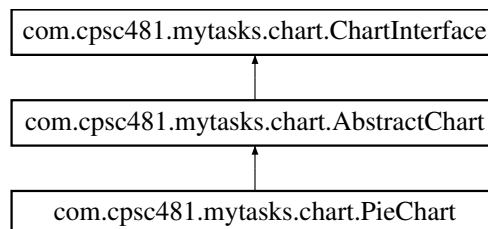


## Chapter 5

# Class Documentation

### 5.1 com.cpsc481.mytasks.chart.AbstractChart Class Reference

Inheritance diagram for com.cpsc481.mytasks.chart.AbstractChart:



#### Public Member Functions

- void **addXYSeries** (XYMultipleSeriesDataset dataset, String[] titles, List< double[]> xValues, List< double[]> yValues, int scale)

#### Protected Member Functions

- XYMultipleSeriesDataset [buildDataset](#) (String[] titles, List< double[]> xValues, List< double[]> yValues)
- XYMultipleSeriesRenderer [buildRenderer](#) (int[] colors, PointStyle[] styles)
- void **setRenderer** (XYMultipleSeriesRenderer renderer, int[] colors, PointStyle[] styles)
- void [setChartSettings](#) (XYMultipleSeriesRenderer renderer, String title, String xTitle, String yTitle, double xMin, double xMax, double yMin, double yMax, int axesColor, int labelsColor)
- XYMultipleSeriesDataset [buildDateDataset](#) (String[] titles, List< Date[]> xValues, List< double[]> yValues)
- CategorySeries [buildCategoryDataset](#) (String title, double[] values, String type)
- MultipleCategorySeries [buildMultipleCategoryDataset](#) (String title, List< String[]> titles, List< double[]> values)
- DefaultRenderer [buildCategoryRenderer](#) (int[] colors)
- XYMultipleSeriesDataset [buildBarDataset](#) (String[] titles, List< double[]> values)
- XYMultipleSeriesRenderer [buildBarRenderer](#) (int[] colors)

#### Additional Inherited Members

##### 5.1.1 Detailed Description

An abstract class for the demo charts to extend. It contains some methods for building datasets and renderers.

## 5.1.2 Member Function Documentation

### 5.1.2.1 XYMultipleSeriesDataset `com.cpsc481.mytasks.chart.AbstractChart.buildBarDataset ( String[] titles, List< double[]> values )` `[protected]`

Builds a bar multiple series dataset using the provided values.

#### Parameters

<i>titles</i>	the series titles
<i>values</i>	the values

#### Returns

the XY multiple bar dataset

### 5.1.2.2 XYMultipleSeriesRenderer `com.cpsc481.mytasks.chart.AbstractChart.buildBarRenderer ( int[] colors )` `[protected]`

Builds a bar multiple series renderer to use the provided colors.

#### Parameters

<i>colors</i>	the series renderers colors
---------------	-----------------------------

#### Returns

the bar multiple series renderer

### 5.1.2.3 CategorySeries `com.cpsc481.mytasks.chart.AbstractChart.buildCategoryDataset ( String title, double[] values, String type )` `[protected]`

Builds a category series using the provided values.

#### Parameters

<i>titles</i>	the series titles
<i>values</i>	the values

#### Returns

the category series

### 5.1.2.4 DefaultRenderer `com.cpsc481.mytasks.chart.AbstractChart.buildCategoryRenderer ( int[] colors )` `[protected]`

Builds a category renderer to use the provided colors.

#### Parameters

<i>colors</i>	the colors
---------------	------------

#### Returns

the category renderer

5.1.2.5 **XYMultipleSeriesDataset** `com.cpsc481.mytasks.chart.AbstractChart.buildDataset ( String[] titles, List< double[]> xValues, List< double[]> yValues )` `[protected]`

Builds an XY multiple dataset using the provided values.

**Parameters**

<i>titles</i>	the series titles
<i>xValues</i>	the values for the X axis
<i>yValues</i>	the values for the Y axis

**Returns**

the XY multiple dataset

5.1.2.6 **XYMultipleSeriesDataset** `com.cpsc481.mytasks.chart.AbstractChart.buildDateDataset ( String[] titles, List< Date[]> xValues, List< double[]> yValues )` `[protected]`

Builds an XY multiple time dataset using the provided values.

**Parameters**

<i>titles</i>	the series titles
<i>xValues</i>	the values for the X axis
<i>yValues</i>	the values for the Y axis

**Returns**

the XY multiple time dataset

5.1.2.7 **MultipleCategorySeries** `com.cpsc481.mytasks.chart.AbstractChart.buildMultipleCategoryDataset ( String title, List< String[]> titles, List< double[]> values )` `[protected]`

Builds a multiple category series using the provided values.

**Parameters**

<i>titles</i>	the series titles
<i>values</i>	the values

**Returns**

the category series

5.1.2.8 **XYMultipleSeriesRenderer** `com.cpsc481.mytasks.chart.AbstractChart.buildRenderer ( int[] colors, PointStyle[] styles )` `[protected]`

Builds an XY multiple series renderer.

**Parameters**

<i>colors</i>	the series rendering colors
<i>styles</i>	the series point styles

**Returns**

the XY multiple series renderers

5.1.2.9 `void com.cpsc481.mytasks.chart.AbstractChart.setChartSettings ( XYMultipleSeriesRenderer renderer, String title, String xTitle, String yTitle, double xMin, double xMax, double yMin, double yMax, int axesColor, int labelsColor )`  
[protected]

Sets a few of the series renderer settings.

**Parameters**

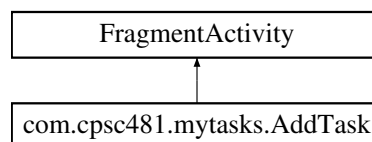
<i>renderer</i>	the renderer to set the properties to
<i>title</i>	the chart title
<i>xTitle</i>	the title for the X axis
<i>yTitle</i>	the title for the Y axis
<i>xMin</i>	the minimum value on the X axis
<i>xMax</i>	the maximum value on the X axis
<i>yMin</i>	the minimum value on the Y axis
<i>yMax</i>	the maximum value on the Y axis
<i>axesColor</i>	the axes color
<i>labelsColor</i>	the labels color

The documentation for this class was generated from the following file:

- `src/com/cpsc481/mytasks/chart/AbstractChart.java`

## 5.2 com.cpsc481.mytasks.AddTask Class Reference

Inheritance diagram for `com.cpsc481.mytasks.AddTask`:

**Public Member Functions**

- void `createTask` (View view)
- boolean `onCreateOptionsMenu` (Menu menu)
- void `showDatePicker` (View view)
- void `showTimePicker` (View view)

**Protected Member Functions**

- void `onCreate` (Bundle savedInstanceState)

**Private Attributes**

- int `dueTime` = 0
- int `dueDate` = 0

- String **difficulty** = "Easy"
- String **taskName** = ""
- String **difficultySettings** [] = {TaskDatabaseHelper.DIFFICULTY\_ONE, TaskDatabaseHelper.DIFFICULTY\_TWO, TaskDatabaseHelper.DIFFICULTY\_THREE}

### 5.2.1 Detailed Description

This class is a fragment that allows the addition of a new task. It instantiates a spinner and the appropriate DatePickerFragments and the TimePickerFragments

#### Author

Ryan Britt

#### Version

1.0

### 5.2.2 Member Function Documentation

#### 5.2.2.1 void com.cpsc481.mytasks.AddTask.createTask ( View view )

This method is used to create a task. It calls a static method (not the best way, I admit) to pass a reference to an object that contains all of the task details. It will not call finish() until all the fields have been instantiated. If all the fields have not been instantiated, it will show a Toast to indicate so.

#### Parameters

<i>view</i>	The view from which createTask is being called, generally a button.
-------------	---

#### Since

1.0

#### 5.2.2.2 void com.cpsc481.mytasks.AddTask.onCreate ( Bundle *savedInstanceState* ) [protected]

This method is called upon the creation of this fragment, per the Android fragment life cycle.

#### Parameters

<i>savedInstanceState</i>	the Bundle that this method is called with, per the android fragment life cycle.
---------------------------	--

#### Since

1.0

#### 5.2.2.3 boolean com.cpsc481.mytasks.AddTask.onCreateOptionsMenu ( Menu *menu* )

This method adds the menu to the action bar if it is present. While included, this method is not used as the action bar menu has not been given any additional function yet.

## Parameters

<i>menu</i>	The menu added to the action bar
-------------	----------------------------------

## Since

1.0

## 5.2.2.4 void com.cpsc481.mytasks.AddTask.showDatePicker ( View view )

\*This method shows the date picker for the due date

## Parameters

<i>view</i>	The view from which this method is called, generally a button.
-------------	--

## Since

1.0

## 5.2.2.5 void com.cpsc481.mytasks.AddTask.showTimePicker ( View view )

This method shows the time picker for the due time

## Parameters

<i>view</i>	The view from which this method is called, generally a button.
-------------	--

## Since

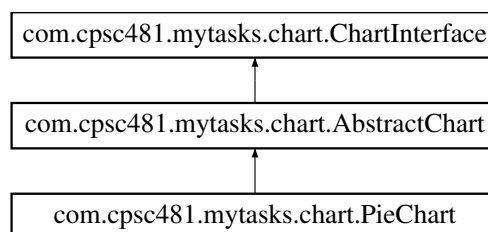
1.0

The documentation for this class was generated from the following file:

- src/com/cpsc481/mytasks/AddTask.java

### 5.3 com.cpsc481.mytasks.chart.ChartInterface Interface Reference

Inheritance diagram for com.cpsc481.mytasks.chart.ChartInterface:



## Public Member Functions

- String [getName](#) ()
- String [getDesc](#) ()
- Intent [execute](#) (Context context, double[] values, String type)

## Public Attributes

- String [NAME](#) = "name"
- String [DESC](#) = "desc"

### 5.3.1 Detailed Description

Defines the charts.

### 5.3.2 Member Function Documentation

#### 5.3.2.1 Intent com.cpsc481.mytasks.chart.ChartInterface.execute ( Context *context*, double[] *values*, String *type* )

Executes the chart.

##### Parameters

<i>context</i>	the context
----------------	-------------

##### Returns

the built intent

Implemented in [com.cpsc481.mytasks.chart.PieChart](#).

#### 5.3.2.2 String com.cpsc481.mytasks.chart.ChartInterface.getDesc ( )

Returns the chart description.

##### Returns

the chart description

Implemented in [com.cpsc481.mytasks.chart.PieChart](#).

#### 5.3.2.3 String com.cpsc481.mytasks.chart.ChartInterface.getName ( )

Returns the chart name.

##### Returns

the chart name

Implemented in [com.cpsc481.mytasks.chart.PieChart](#).

### 5.3.3 Member Data Documentation

#### 5.3.3.1 String com.cpsc481.mytasks.chart.ChartInterface.DESC = "desc"

A constant for the description field in a list activity.

### 5.3.3.2 String com.cpsc481.mytasks.chart.ChartInterface.NAME = "name"

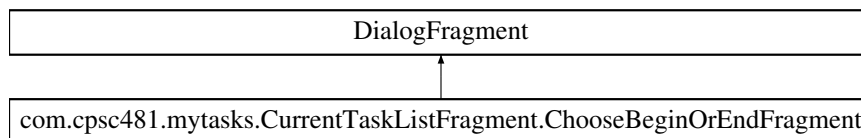
A constant for the name field in a list activity.

The documentation for this interface was generated from the following file:

- src/com/cpsc481/mytasks/chart/ChartInterface.java

## 5.4 com.cpsc481.mytasks.CurrentTaskListFragment.ChooseBeginOrEndFragment Class Reference

Inheritance diagram for com.cpsc481.mytasks.CurrentTaskListFragment.ChooseBeginOrEndFragment:



### Public Member Functions

- Dialog **onCreateDialog** (Bundle savedInstanceState)

### 5.4.1 Detailed Description

This private class is the fragment for choosing to begin or end a task dialog.

#### Author

Ryan Britt

#### Version

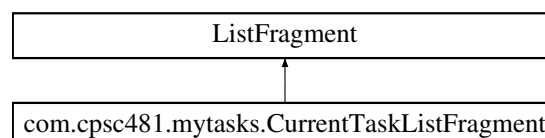
1.0

The documentation for this class was generated from the following file:

- src/com/cpsc481/mytasks/CurrentTaskListFragment.java

## 5.5 com.cpsc481.mytasks.CurrentTaskListFragment Class Reference

Inheritance diagram for com.cpsc481.mytasks.CurrentTaskListFragment:



### Classes

- class [ChooseBeginOrEndFragment](#)



## Public Member Functions

- View **onCreateView** (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void **showDiffBegin** ()
- void **onCreate** (Bundle savedInstanceState)
- void **removeTask** (long id)
- void **startTask** (long id)
- void **onListItemClick** (ListView l, View v, int position, long id)

## Static Public Member Functions

- static void **addTask** (**Task** taskToAdd)

## Private Member Functions

- void **fillData** ()

## Private Attributes

- **TaskSimpleAdapter** **tasks**
- Activity **currentActivity**
- int **clickedItemIndex**

## Static Private Attributes

- static **TaskDataStore** **taskData**
- static ArrayList< **Task** > **taskList**

### 5.5.1 Detailed Description

Fragment that displays the current task list in the tabs.

#### Author

Ryan Britt

#### Version

1.0

### 5.5.2 Member Function Documentation

5.5.2.1 static void com.cpsc481.mytasks.CurrentTaskListFragment.addTask ( **Task** *taskToAdd* ) [static]

Adds a new task to the database which is represented as a Task object on input. Calls **addnewTask**;

#### Parameters

<i>taskToAdd</i>	The Task object which will be decomposed and added to the database.
------------------	---

### 5.5.2.2 void com.cpsc481.mytasks.CurrentTaskListFragment.fillData ( ) [private]

This method fills the data in the List.

This method of getting the content from a database has been deprecated. A future way to make this app better would be to use content providers to get the content from content providers and using loaders as this method loads the information on the UI thread, and not on a separate thread.

[findAllTasks](#)

### 5.5.2.3 void com.cpsc481.mytasks.CurrentTaskListFragment.onCreate ( Bundle savedInstanceState )

This is the overridden onCreate method that checks to see if the initial time per difficulty has been set. It prompts the user to do so, if it has not been set. This method also initializes the database.

#### Parameters

<i>savedInstanceState</i>	The saved state if it is being resumed.
---------------------------	---

### 5.5.2.4 void com.cpsc481.mytasks.CurrentTaskListFragment.onListItemClick ( ListView l, View v, int position, long id )

This is called when a list item is clicked.

#### See Also

android.app.ListFragment::onListItemClick(android.widget.ListView, android.view.View, int, long)

It shows the dialog box to set either begin or end the task.

### 5.5.2.5 void com.cpsc481.mytasks.CurrentTaskListFragment.removeTask ( long id )

Removes the task from the list. Calls [completeTask](#) to complete the task and [fillData](#) to refresh the screen.

#### Parameters

<i>id</i>	The id of the task to be removed
-----------	----------------------------------

### 5.5.2.6 void com.cpsc481.mytasks.CurrentTaskListFragment.showDiffBegin ( )

This method shows the difficulty settings at the first run of the app. It starts out with an Easy difficulty setting and sets the Intro flag to true, which lets the DialogFragment know to show the Medium and Hard settings in sequential order after this dialog.

Calls [SetTimeFragment](#)

### 5.5.2.7 void com.cpsc481.mytasks.CurrentTaskListFragment.startTask ( long id )

Begins the task and calls [startTask](#)

#### Parameters

<i>id</i>	The id of the task to begin
-----------	-----------------------------

The documentation for this class was generated from the following file:

- src/com/cpsc481/mytasks/CurrentTaskListFragment.java

## 5.6 com.cpsc481.mytasks.customclasses.DateAndTime Class Reference

### Static Public Member Functions

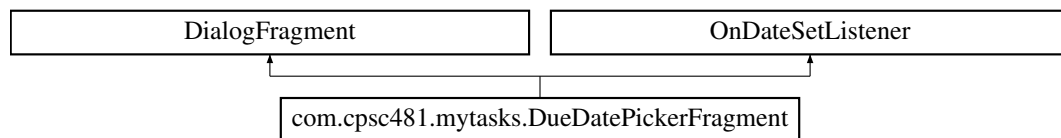
- static String **printDate** (int date)
- static int **getMonth** (int date)
- static int **getDay** (int date)
- static int **getYear** (int date)
- static String **printTime** (int time)
- static int **getHour** (int time)
- static int **getMinute** (int time)
- static String **amOrPm** (int time)
- static int **combineDate** (int month, int day, int year)
- static int **combineTime** (int hour, int minute)

The documentation for this class was generated from the following file:

- src/com/cpsc481/mytasks/customclasses/DateAndTime.java

## 5.7 com.cpsc481.mytasks.DueDatePickerFragment Class Reference

Inheritance diagram for com.cpsc481.mytasks.DueDatePickerFragment:



### Public Member Functions

- void **setHandler** (Handler t)
- Dialog **onCreateDialog** (Bundle savedInstanceState)
- void **onDateSet** (DatePicker view, int year, int month, int day)

#### 5.7.1 Detailed Description

Fragment that shows a DueDatePicker and handles the input by sending a Message through a Handler class instance.

#### 5.7.2 Member Function Documentation

##### 5.7.2.1 void com.cpsc481.mytasks.DueDatePickerFragment.onDateSet ( DatePicker view, int year, int month, int day )

This method tells the datepicker what to do when the date is set.

## Parameters

<i>view</i>	The DatePicker view that the message is being received from
<i>year</i>	The year chosen for the datepicker
<i>month</i>	The month chosen for the datepicker
<i>day</i>	The day chosen for the datepicker.

5.7.2.2 void com.cpsc481.mytasks.DueDatePickerFragment.setHandler ( Handler *t* )

This method sets the data handler in order to make the data visible to other classes. The handler method is only used in [DueDatePickerFragment](#) and [DueTimePickerFragment](#)

## Parameters

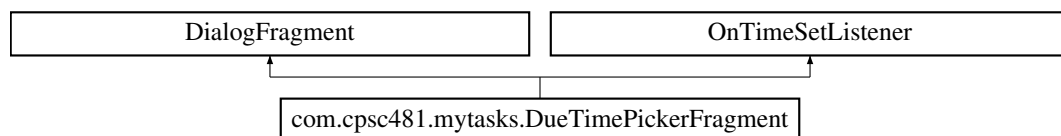
<i>t</i>	The handler that the datepicker is to use.
----------	--

The documentation for this class was generated from the following file:

- src/com/cpsc481/mytasks/DueDatePickerFragment.java

## 5.8 com.cpsc481.mytasks.DueTimePickerFragment Class Reference

Inheritance diagram for com.cpsc481.mytasks.DueTimePickerFragment:



## Public Member Functions

- void [setHandler](#) (Handler *t*)
- Dialog **onCreateDialog** (Bundle savedInstanceState)
- void [onTimeSet](#) (TimePicker view, int hourOfDay, int minute)

## 5.8.1 Detailed Description

This class is the fragment for a time picker dialog and handles the input by sending a Message through a Handler class instance.

## 5.8.2 Member Function Documentation

5.8.2.1 void com.cpsc481.mytasks.DueTimePickerFragment.onTimeSet ( TimePicker *view*, int *hourOfDay*, int *minute* )

This method is called when the time is set in the time picker and tells the picker what to do with the data.

## Parameters

<i>view</i>	The timepicker view from which this method is called
<i>hourOfDay</i>	The hour of the day (in 24 hr notation) that has been chosen
<i>minute</i>	The minute of the day that has been chosen

5.8.2.2 void com.cpsc481.mytasks.DueTimePickerFragment.setHandler ( Handler *t* )

This method sets the data handler in order to make the data visible to other classes. The handler method is only used in [DueDatePickerFragment](#) and [DueTimePickerFragment](#)

## Parameters

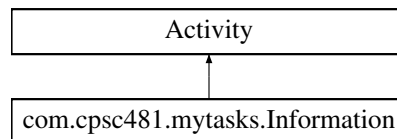
<i>t</i>	The handler that the timepicker is to use.
----------	--

The documentation for this class was generated from the following file:

- `src/com/cpsc481/mytasks/DueTimePickerFragment.java`

## 5.9 com.cpsc481.mytasks.Information Class Reference

Inheritance diagram for com.cpsc481.mytasks.Information:



## Public Member Functions

- boolean **onCreateOptionsMenu** (Menu menu)

## Protected Member Functions

- void **onCreate** (Bundle savedInstanceState)

## 5.9.1 Detailed Description

This is an empty Activity class that simply holds the text to show the [Information](#) view.

## Author

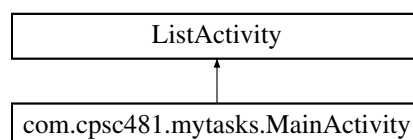
rjbritt

The documentation for this class was generated from the following file:

- `src/com/cpsc481/mytasks/Information.java`

## 5.10 com.cpsc481.mytasks.MainActivity Class Reference

Inheritance diagram for com.cpsc481.mytasks.MainActivity:



## Public Member Functions

- void [showDiff](#) (View view)
- void [getInfo](#) (View view)
- void [addTaskButton](#) (View view)
- boolean **onCreateOptionsMenu** (Menu menu)

## Protected Member Functions

- void **onCreate** (Bundle savedInstanceState)

### 5.10.1 Detailed Description

Main class that loads in the app. This activity holds the tabs, which are the fragments [CurrentTaskListFragment](#), [Reports](#), [SetDifficultyTimeEstimateFragment](#)

#### Author

rjbritt

### 5.10.2 Member Function Documentation

#### 5.10.2.1 void com.cpsc481.mytasks.MainActivity.addTaskButton ( View view )

Begins the process to add a new task. Begins a new Activity

##### Parameters

<i>view</i>	The view this is called from, typically a button.
-------------	---

#### 5.10.2.2 void com.cpsc481.mytasks.MainActivity.getInfo ( View view )

Shows information view. Begins a new activity

##### Parameters

<i>view</i>	The view this is called from, typically a button.
-------------	---

#### 5.10.2.3 void com.cpsc481.mytasks.MainActivity.showDiff ( View view )

Shows the dialog for setting the time based on the button that was pressed. The button is physically located on the Settings tab XML, but the button calls are redirected here.

##### Parameters

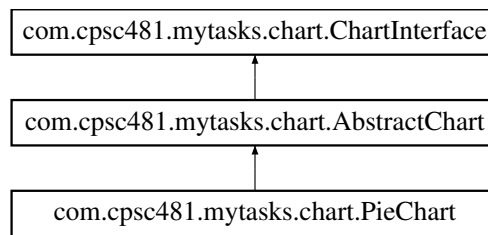
<i>view</i>	The view that called this method, typically a button
-------------	--

The documentation for this class was generated from the following file:

- `src/com/cpsc481/mytasks/MainActivity.java`

## 5.11 com.cpsc481.mytasks.chart.PieChart Class Reference

Inheritance diagram for com.cpsc481.mytasks.chart.PieChart:



### Public Member Functions

- String [getName](#) ()
- String [getDesc](#) ()
- Intent [execute](#) (Context context, double[] values, String type)

### Static Public Attributes

- static String **COMPLETION\_TYPE** = "completion"
- static String **ONTIME\_TYPE** = "onTime"
- static String **EASY\_STAT\_TYPE** = "easyChart"
- static String **MED\_STAT\_TYPE** = "mediumChart"
- static String **HARD\_STAT\_TYPE** = "hardChart"

### Additional Inherited Members

#### 5.11.1 Detailed Description

Budget demo pie chart.

#### 5.11.2 Member Function Documentation

##### 5.11.2.1 Intent com.cpsc481.mytasks.chart.PieChart.execute ( Context *context*, double[] *values*, String *type* )

Executes the chart. Builds the appropriate chart from the values based on the type of chart.

##### Parameters

<i>context</i>	the context
<i>values</i>	The values being passed to build the pie chart.
<i>type</i>	The type of chart to be created. From the public static fields in this class.

##### Returns

the built intent

Implements [com.cpsc481.mytasks.chart.ChartInterface](#).

##### 5.11.2.2 String com.cpsc481.mytasks.chart.PieChart.getDesc ( )

Returns the chart description.

**Returns**

the chart description

Implements [com.cpsc481.mytasks.chart.ChartInterface](#).

### 5.11.2.3 String com.cpsc481.mytasks.chart.PieChart.getName ( )

Returns the chart name.

**Returns**

the chart name

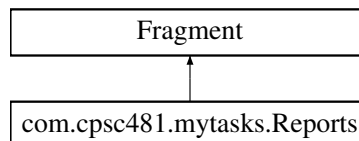
Implements [com.cpsc481.mytasks.chart.ChartInterface](#).

The documentation for this class was generated from the following file:

- src/com/cpsc481/mytasks/chart/PieChart.java

## 5.12 com.cpsc481.mytasks.Reports Class Reference

Inheritance diagram for com.cpsc481.mytasks.Reports:

**Public Member Functions**

- View **onCreateView** (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void **onActivityCreated** (Bundle savedInstanceState)
- void [showCompletionChart](#) (View view)
- void [showOnTimeChart](#) (View view)
- void [showDifficultyChart](#) (String difficulty)
- List< Integer > [timesTakenPerDifficulty](#) (String difficulty)
- double[] [underOverOrNearPerDifficulty](#) (String difficulty, List< Integer > timesPerDifficulty)

**Private Member Functions**

- String [getAdviceLine](#) (double[] statistics, String difficulty)

### 5.12.1 Detailed Description

This is a fragment that computes and shows the reports based on all the information that is available.

**Author**

Ryan Britt



## 5.12.2 Member Function Documentation

### 5.12.2.1 String com.cpsc481.mytasks.Reports.getAdviceLine ( double[] *statistics*, String *difficulty* ) [private]

This method gets the advice line (whether to increase, decrease, or not change the estimated time) for a particular difficulty with a set of statistics.

#### Parameters

<i>statistics</i>	An array of form [under,over,near] where each of the fields are the number of tasks for that particular difficulty that are under the estimated time, over the estimated time, or near the estimated time.
<i>difficulty</i>	This is the difficulty for which you would like to obtain an advice line for.

#### Returns

This is a String that describes what should be done to the estimated time of the difficulty based on the information available.

### 5.12.2.2 void com.cpsc481.mytasks.Reports.showCompletionChart ( View *view* )

This method shows the pie chart for how many tasks have been completed vs not.

#### Parameters

<i>view</i>	The view from which this is called.
-------------	-------------------------------------

### 5.12.2.3 void com.cpsc481.mytasks.Reports.showDifficultyChart ( String *difficulty* )

This method shows the pie chart for each difficulty as to how many are under, over, or near the estimated time

#### Parameters

<i>difficulty</i>	The name of the difficulty chart that is to be displayed.
-------------------	---

### 5.12.2.4 void com.cpsc481.mytasks.Reports.showOnTimeChart ( View *view* )

This method shows the pie chart for how many tasks have been completed on time vs not

#### Parameters

<i>view</i>	The view from which this is called.
-------------	-------------------------------------

### 5.12.2.5 List<Integer> com.cpsc481.mytasks.Reports.timesTakenPerDifficulty ( String *difficulty* )

This method returns the raw value of the amount of time taken per task for a particular difficulty.

#### Parameters

<i>difficulty</i>	The difficulty to analyze
-------------------	---------------------------

**Returns**

a List with all the absolute time values of the amount of time taken per task.

#### 5.12.2.6 `double [] com.cpsc481.mytasks.Reports.underOverOrNearPerDifficulty ( String difficulty, List< Integer > timesPerDifficulty )`

This method analyzes the time taken per task for a particular difficulty. It splits it up into times that were under the estimated time (<- the LEE\_WAY\_PERCENTAGE constant as defined), times that were over the estimated time (>+ the LEE\_WAY\_PERCENTAGE constant as defined), and times that were 'near' the estimated time (+- the LEE\_WAY\_PERCENTAGE constant as defined)

**Parameters**

<i>difficulty</i>	the difficulty to analyze
<i>timesPer-Difficulty</i>	the list of all the absolute values of the time taken per task.

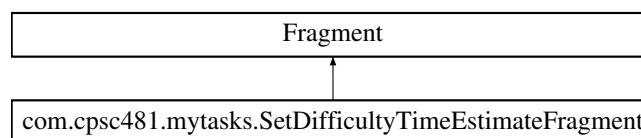
**Returns**

The documentation for this class was generated from the following file:

- `src/com/cpsc481/mytasks/Reports.java`

## 5.13 `com.cpsc481.mytasks.SetDifficultyTimeEstimateFragment` Class Reference

Inheritance diagram for `com.cpsc481.mytasks.SetDifficultyTimeEstimateFragment`:

**Public Member Functions**

- View **onCreateView** (LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
- void **onActivityCreated** (Bundle savedInstanceState)

### 5.13.1 Detailed Description

This fragment is used to show the buttons to change the estimated time per difficulty.

**Author**

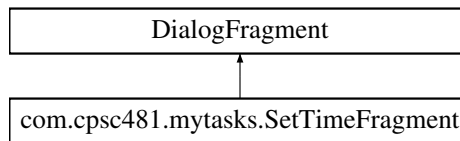
rjbritt

The documentation for this class was generated from the following file:

- `src/com/cpsc481/mytasks/SetDifficultyTimeEstimateFragment.java`

## 5.14 com.cpsc481.mytasks.SetTimeFragment Class Reference

Inheritance diagram for com.cpsc481.mytasks.SetTimeFragment:



### Public Member Functions

- Dialog [onCreateDialog](#) (Bundle savedInstanceState)

### Private Attributes

- NumberPicker **hrPick**
- NumberPicker **minPick**

#### 5.14.1 Detailed Description

This fragment is meant to be shown as a dialog The XML which this references shows 2 NumberPickers This fragment handles the input from the XML and calls more dialogs if it is in "intro" mode.

#### Author

Ryan Britt

#### 5.14.2 Member Function Documentation

##### 5.14.2.1 Dialog com.cpsc481.mytasks.SetTimeFragment.onCreateDialog ( Bundle savedInstanceState )

This onCreateDialog sets up the min and max values for the hour and minute pickers as well as the action performed when the "Enter" button is tapped.

#### See Also

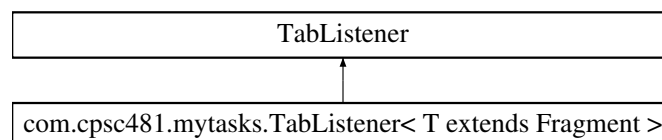
android.app.DialogFragment::onCreateDialog(android.os.Bundle)

The documentation for this class was generated from the following file:

- src/com/cpsc481/mytasks/SetTimeFragment.java

## 5.15 com.cpsc481.mytasks.TabListener< T extends Fragment > Class Reference

Inheritance diagram for com.cpsc481.mytasks.TabListener< T extends Fragment >:



## Public Member Functions

- [TabListener](#) (Activity activity, String tag, Class< T > clz)
- void **onTabSelected** (ActionBar.Tab tab, FragmentTransaction ft)
- void **onTabUnselected** (ActionBar.Tab tab, FragmentTransaction ft)
- void **onTabReselected** (Tab tab, FragmentTransaction ft)

## Private Attributes

- Fragment **mFragment**
- final Activity **mActivity**
- final String **mTag**
- final Class< T > **mClass**

### 5.15.1 Detailed Description

This class is a action bar tab listener to add tabs

#### Author

rjbritt

#### Parameters

<T>	
-----	--

### 5.15.2 Constructor & Destructor Documentation

5.15.2.1 `com.cpsc481.mytasks.TabListener< T extends Fragment >.TabListener ( Activity activity, String tag, Class< T > clz )`

Constructor used each time a new tab is created.

#### Parameters

<i>activity</i>	The host Activity, used to instantiate the fragment
<i>tag</i>	The identifier tag for the fragment
<i>clz</i>	The fragment's Class, used to instantiate the fragment

The documentation for this class was generated from the following file:

- `src/com/cpsc481/mytasks/TabListener.java`

## 5.16 com.cpsc481.mytasks.customclasses.Task Class Reference

### Public Member Functions

- **Task** (String name, String difficulty, int endDate, int endTime)
- **Task** (String name, String difficulty, int startDate, int endDate, int startTime, int endTime)
- String **toString** ()
- String [getName](#) ()
- void [setName](#) (String name)
- String [getDifficulty](#) ()

- void [setDifficulty](#) (String difficulty)
- int [getStartDate](#) ()
- void [setStartDate](#) (int startDate)
- int [getDueDate](#) ()
- void [setDueDate](#) (int endDate)
- int [getStartTime](#) ()
- void [setStartTime](#) (int startTime)
- int [getDueTime](#) ()
- void [setDueTime](#) (int endTime)

### Private Attributes

- String **name**
- String **difficulty**
- int **startDate**
- int **dueDate**
- int **startTime**
- int **dueTime**

## 5.16.1 Member Function Documentation

### 5.16.1.1 String com.cpsc481.mytasks.customclasses.Task.getDifficulty ( )

#### Returns

the difficulty

### 5.16.1.2 int com.cpsc481.mytasks.customclasses.Task.getDueDate ( )

#### Returns

the dueDate

### 5.16.1.3 int com.cpsc481.mytasks.customclasses.Task.getDueTime ( )

#### Returns

the dueTime

### 5.16.1.4 String com.cpsc481.mytasks.customclasses.Task.getName ( )

#### Returns

the name

### 5.16.1.5 int com.cpsc481.mytasks.customclasses.Task.getStartDate ( )

#### Returns

the startDate

5.16.1.6 `int com.cpsc481.mytasks.customclasses.Task.getStartTime ( )`

Returns

the startTime

5.16.1.7 `void com.cpsc481.mytasks.customclasses.Task.setDifficulty ( String difficulty )`

Parameters

<i>difficulty</i>	the difficulty to set
-------------------	-----------------------

5.16.1.8 `void com.cpsc481.mytasks.customclasses.Task.setDueDate ( int endDate )`

Parameters

<i>dueDate</i>	the dueDate to set
----------------	--------------------

5.16.1.9 `void com.cpsc481.mytasks.customclasses.Task.setDueTime ( int endTime )`

Parameters

<i>dueTime</i>	the dueTime to set
----------------	--------------------

5.16.1.10 `void com.cpsc481.mytasks.customclasses.Task.setName ( String name )`

Parameters

<i>name</i>	the name to set
-------------	-----------------

5.16.1.11 `void com.cpsc481.mytasks.customclasses.Task.setStartDate ( int startDate )`

Parameters

<i>startDate</i>	the startDate to set
------------------	----------------------

5.16.1.12 `void com.cpsc481.mytasks.customclasses.Task.setStartTime ( int startTime )`

Parameters

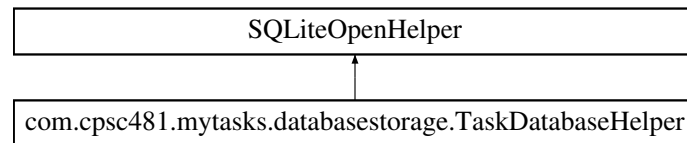
<i>startTime</i>	the startTime to set
------------------	----------------------

The documentation for this class was generated from the following file:

- `src/com/cpsc481/mytasks/customclasses/Task.java`

## 5.17 `com.cpsc481.mytasks.databasesstorage.TaskDatabaseHelper` Class Reference

Inheritance diagram for `com.cpsc481.mytasks.databasesstorage.TaskDatabaseHelper`:



## Public Member Functions

- **TaskDatabaseHelper** (Context context)
- void **onCreate** (SQLiteDatabase db)
- void **onUpgrade** (SQLiteDatabase db, int oldVersion, int newVersion)

## Static Public Attributes

- static final String **UNIQUE\_ID** = "\_id"
- static final int **YES** = 1
- static final int **NO** = 0
- static final String **TASK\_TABLE** = "Task"
- static final String **TASK\_TASKNAME** = "TaskName"
- static final String **TASK\_DIFFICULTY** = "Difficulty"
- static final String **TASK\_STARTDATE** = "StartDate"
- static final String **TASK\_STARTTIME** = "StartTime"
- static final String **TASK\_ENDDATE** = "EndDate"
- static final String **TASK\_ENDTIME** = "EndTime"
- static final String **TASK\_DUEDATE** = "DueDate"
- static final String **TASK\_DUETIME** = "DueTime"
- static final String **TASK\_COMPLETE\_FLAG** = "CompletedFlag"
- static final String **TASK\_COMPLETE\_ON\_TIME\_FLAG** = "CompletedOTFlag"
- static final String **TASKNAME\_TABLE** = "TaskName"
- static final String **TASKNAME\_NAME** = "name"
- static final String **STARTDATE\_TABLE** = "StartDate"
- static final String **STARTDATE\_DATE** = "date"
- static final String **STARTTIME\_TABLE** = "StartTime"
- static final String **STARTTIME\_TIME** = "time"
- static final String **ENDDATE\_TABLE** = "EndDate"
- static final String **ENDDATE\_DATE** = "date"
- static final String **ENDTIME\_TABLE** = "EndTime"
- static final String **ENDTIME\_TIME** = "time"
- static final String **DUEDATE\_TABLE** = "DueDate"
- static final String **DUEDATE\_DATE** = "date"
- static final String **DUETIME\_TABLE** = "DueTime"
- static final String **DUETIME\_TIME** = "time"
- static final String **DIFFICULTY\_TABLE** = "Difficulty"
- static final String **DIFFICULTY\_DIFFICULTY** = "difficulty"
- static final String **DIFFICULTY\_ESTIMATEDTIME** = "estimateCompleteTime"
- static final String **DIFFICULTY\_ONE** = "Easy"
- static final String **DIFFICULTY\_TWO** = "Medium"
- static final String **DIFFICULTY\_THREE** = "Hard"

## Static Private Attributes

- static final String **LOG\_TAG** = "Database Helper Log"
- static final String **DATABASE\_NAME** = "TaskDatabaseHelper"
- static final int **DATABASE\_VERSION** = 1
- static final String **CREATE\_MASTER\_TABLE**
- static final String **CREATE\_TASKNAME\_TABLE**
- static final String **CREATE\_DIFFICULTY\_TABLE**
- static final String **CREATE\_STARTDATE\_TABLE**
- static final String **CREATE\_STARTTIME\_TABLE**
- static final String **CREATE\_ENDDATE\_TABLE**
- static final String **CREATE\_ENDTIME\_TABLE**
- static final String **CREATE\_DUEDATE\_TABLE**
- static final String **CREATE\_DUETIME\_TABLE**

### 5.17.1 Detailed Description

This class is an embodiment of the sqlite database used to house all of the data for this app.

Author

rjbritt

### 5.17.2 Member Data Documentation

5.17.2.1 final String com.cpsc481.mytasks.databasestorage.TaskDatabaseHelper.CREATE\_DIFFICULTY\_TABLE [static], [private]

Initial value:

```
=
    "create table Difficulty(_id integer primary key autoincrement,"
    + "difficulty text unique, estimateCompleteTime integer);"
```

5.17.2.2 final String com.cpsc481.mytasks.databasestorage.TaskDatabaseHelper.CREATE\_DUEDATE\_TABLE [static], [private]

Initial value:

```
=
    "create table DueDate(_id integer primary key autoincrement,"
    + "date integer unique);"
```

5.17.2.3 final String com.cpsc481.mytasks.databasestorage.TaskDatabaseHelper.CREATE\_DUETIME\_TABLE [static], [private]

Initial value:

```
=
    "create table DueTime(_id integer primary key autoincrement,"
    + "time integer unique);"
```



5.17.2.4 `final String com.cpsc481.mytasks.databasesstorage.TaskDatabaseHelper.CREATE_ENDDATE_TABLE` [static], [private]

**Initial value:**

```
=
    "create table EndDate(_id integer primary key autoincrement,"
    + "date integer unique);"
```

5.17.2.5 `final String com.cpsc481.mytasks.databasesstorage.TaskDatabaseHelper.CREATE_ENDTIME_TABLE` [static], [private]

**Initial value:**

```
=
    "create table EndTime(_id integer primary key autoincrement,"
    + "time integer unique);"
```

5.17.2.6 `final String com.cpsc481.mytasks.databasesstorage.TaskDatabaseHelper.CREATE_MASTER_TABLE` [static], [private]

**Initial value:**

```
=
    "create table Task (_id integer primary key autoincrement, "
    + "TaskName integer,Difficulty integer, StartDate integer," +
    "StartTime integer,EndDate integer,EndTime integer," +
    " DueDate integer, DueTime integer, CompletedFlag integer, CompletedOTFlag integer);"
```

5.17.2.7 `final String com.cpsc481.mytasks.databasesstorage.TaskDatabaseHelper.CREATE_STARTDATE_TABLE` [static], [private]

**Initial value:**

```
=
    "create table StartDate(_id integer primary key autoincrement,"
    + "date integer unique);"
```

5.17.2.8 `final String com.cpsc481.mytasks.databasesstorage.TaskDatabaseHelper.CREATE_STARTTIME_TABLE` [static], [private]

**Initial value:**

```
=
    "create table StartTime(_id integer primary key autoincrement,"
    + "time integer unique);"
```

5.17.2.9 `final String com.cpsc481.mytasks.databasesstorage.TaskDatabaseHelper.CREATE_TASKNAME_TABLE` [static], [private]

**Initial value:**

```
=
    "create table TaskName(_id integer primary key autoincrement,"
    + "name text unique );"
```

The documentation for this class was generated from the following file:

- `src/com/cpsc481/mytasks/databasesstorage/TaskDatabaseHelper.java`

## 5.18 com.cpsc481.mytasks.databasestorage.TaskDataStore Class Reference

### Public Member Functions

- **TaskDataStore** (Context ctx)
- **TaskDataStore** **open** () throws SQLException
- void **addNewTask** (String name, int duedate, int duetime, String difficulty)
- void **completeTask** (long id)
- Cursor **findAllTasks** ()
- Cursor **findSelectForTask** (long id)
- Cursor **findTask** (long id)
- long **getTaskNameID** (String name)
- long **getStartDateID** (int startDate)
- long **getStartTimeID** (int startTime)
- long **getEndDateID** (int endDate)
- long **getEndTimeID** (int endTime)
- long **getDifficultyID** (String difficulty)
- long **getDueDateID** (int duedate)
- long **getDueTimeID** (int duetime)
- boolean **hasSetTime** ()
- boolean **isComplete** (int id)
- void **startTask** (long id)
- boolean **isTaskStarted** (long id)

### Static Public Member Functions

- static Cursor **findTasksForDifficulty** (String difficulty)
- static void **updateTimeForDifficulty** (String difficulty, int newTime)
- static int **getEstimatedTimeForDifficulty** (String difficulty)
- static int[] **getTotalCompletedVsNot** ()
- static int[] **getCompletedOnTimeVsNot** ()

### Static Public Attributes

- static **TaskDatabaseHelper** **databaseHelper**

### Private Attributes

- final Context **currentContext**

### Static Private Attributes

- static final String **LOG\_TAG** = "TaskDataStore log"
- static SQLiteDatabase **taskDB**

#### 5.18.1 Detailed Description

This is the datastore class that stores all the methods for manipulating the database, such as adding and removing items from the database.

#### Author

rjbritt

## 5.18.2 Member Function Documentation

5.18.2.1 `void com.cpsc481.mytasks.databasesstorage.TaskDataStore.addNewTask ( String name, int duedate, int duetime, String difficulty )`

Adds a new task to the database.

### Parameters

<i>name</i>	The name of the Task
<i>duedate</i>	The task's duedate
<i>duetime</i>	The task's duetime
<i>difficulty</i>	The task's difficulty

5.18.2.2 `void com.cpsc481.mytasks.databasesstorage.TaskDataStore.completeTask ( long id )`

This sets the endDate/time entry in the main table and creates a new entry in the endDate/Time tables as needed. It also sets the onComplete flag to YES and the completeOnTime flag to whether or not the task was completed on time.

### Parameters

<i>id</i>	
-----------	--

5.18.2.3 `Cursor com.cpsc481.mytasks.databasesstorage.TaskDataStore.findAllTasks ( )`

This method returns a cursor pointing to a joined table that consists of the string or integer entries of the task name, start date, start time, end date, end time, due date, due time, difficulty, and completed flag for only the tasks that have been completed.

### Returns

The cursor mentioned above.

5.18.2.4 `Cursor com.cpsc481.mytasks.databasesstorage.TaskDataStore.findSelectForTask ( long id )`

returns a cursor pointing to select raw data results of the joined table that tells a particular tasks name, due date, due time, difficulty, and completed flag

### Parameters

<i>id</i>	The id of the task for which you want to find the select data
-----------	---

### Returns

the cursor mentioned above

5.18.2.5 `Cursor com.cpsc481.mytasks.databasesstorage.TaskDataStore.findTask ( long id )`

returns a cursor pointing all integer indices from the main table for a particular task

### Parameters

<i>id</i>	The whose information you want to know
-----------	--

**Returns**

the cursor described above

**5.18.2.6** `static Cursor com.cpsc481.mytasks.databasesstorage.TaskDataStore.findTasksForDifficulty ( String difficulty )`  
`[static]`

Returns a cursor that points to the results of a query for finding the task name, start date, start time, end date, end time, for a particular difficulty.

**Parameters**

<i>difficulty</i>	The difficulty that the list of tasks for is desired
-------------------	--

**Returns**

the cursor mentioned above.

**5.18.2.7** `static int [] com.cpsc481.mytasks.databasesstorage.TaskDataStore.getCompletedOnTimeVsNot ( )` `[static]`

This method checks to see how many tasks are completed on time or not and returns a size 2 integer array with the contents of how many tasks are completed on time in index 0 and how many are not completed on time in index 1.

**Returns**

the array as mentioned above

**5.18.2.8** `long com.cpsc481.mytasks.databasesstorage.TaskDataStore.getDifficultyID ( String difficulty )`

Gets the ID for a particular difficulty from its respective table

**Parameters**

<i>difficulty</i>	the string difficulty whose ID you desire
-------------------	---

**Returns**

a long representing the ID of the difficulty

**5.18.2.9** `long com.cpsc481.mytasks.databasesstorage.TaskDataStore.getDueDateID ( int duedate )`

Gets the ID for a particular due date from its respective table

**Parameters**

<i>duedate</i>	the integer due date whose ID you desire
----------------	--

**Returns**

a long representing the ID of the due date

**5.18.2.10** `long com.cpsc481.mytasks.databasesstorage.TaskDataStore.getDueTimeID ( int duetime )`

Gets the ID for a particular due time from its respective table

**Parameters**

<i>duetime</i>	the integer due time whose ID you desire
----------------	--

**Returns**

a long representing the ID of the due date

**5.18.2.11** `long com.cpsc481.mytasks.databasesstorage.TaskDataStore.getEndDateID ( int endDate )`

Gets the ID for an end date from its respective table

**Parameters**

<i>endDate</i>	the integer end date whose ID you desire
----------------	--

**Returns**

a long representing the ID of the end date

**5.18.2.12** `long com.cpsc481.mytasks.databasesstorage.TaskDataStore.getEndTimeID ( int endTime )`

Gets the ID for an end time from its respective table

**Parameters**

<i>endTime</i>	the integer end time whose ID you desire
----------------	--

**Returns**

a long representing the ID of the end time

**5.18.2.13** `static int com.cpsc481.mytasks.databasesstorage.TaskDataStore.getEstimatedTimeForDifficulty ( String difficulty )`  
`[static]`

This returns the estimated time for a particular difficulty as shown in the the difficulty table

**Parameters**

<i>difficulty</i>	the difficulty whose estimated time is desired
-------------------	--

**Returns**

the integer representation of the time for the difficulty

**5.18.2.14** `long com.cpsc481.mytasks.databasesstorage.TaskDataStore.getStartDateID ( int startDate )`

Gets the ID for a start date from its respective table

## Parameters

<i>startDate</i>	The integer startdate whose ID you desire
------------------	---

## Returns

a long representing the ID of the start date

5.18.2.15 `long com.cpsc481.mytasks.databasestorage.TaskDataStore.getStartTimelD ( int startTime )`

Gets the ID for a start time from its respective table

## Parameters

<i>startTime</i>	the integer start time whose ID you desire
------------------	--

## Returns

a long representing the ID of the start time

5.18.2.16 `long com.cpsc481.mytasks.databasestorage.TaskDataStore.getTaskNameID ( String name )`

Gets the ID for a task name from its respective table

## Parameters

<i>name</i>	The task name whose ID you desire
-------------	-----------------------------------

## Returns

a long representing the ID of the task name

5.18.2.17 `static int [] com.cpsc481.mytasks.databasestorage.TaskDataStore.getTotalCompletedVsNot ( ) [static]`

This method checks to see how many tasks are completed or not and returns a size 2 integer array with the contents of how many tasks are completed in index 0 and how many are not completed in index 1.

## Returns

the array as mentioned above

5.18.2.18 `boolean com.cpsc481.mytasks.databasestorage.TaskDataStore.hasSetTime ( )`

This method is used to determine if the estimated time has been set in the internal database.

## Returns

boolean whether or not the time has been set initially.

5.18.2.19 `boolean com.cpsc481.mytasks.databasestorage.TaskDataStore.isComplete ( int id )`

This method checks to see if a particular task has been marked as complete or not.

## Parameters

<i>id</i>	The ID of the task whose completion status is to be investigated
-----------	--

## Returns

a boolean describing whether or not the task is marked as complete or not. True is equivalent to yes, the task is complete.

5.18.2.20 boolean com.cpsc481.mytasks.databasesetorage.TaskDataStore.isTaskStarted ( long *id* )

This method checks to see if the task has started or not. The start time and date is marked as -1 if the task has not been started and this method checks that status on a particular task.

## Parameters

<i>id</i>	The task to check the start status of
-----------	---------------------------------------

## Returns

a boolean describing whether the task has been started or not. True is equivalent to yes, the task has started.

5.18.2.21 void com.cpsc481.mytasks.databasesetorage.TaskDataStore.startTask ( long *id* )

This method marks a task as begun at a particular time. It finds or inserts new start times and dates into the Start-Date and StartTime tables as necessary and then updates the main table with the appropriate ID of the StartDate and StartTime.

## Parameters

<i>id</i>	The id of the task to start.
-----------	------------------------------

5.18.2.22 static void com.cpsc481.mytasks.databasesetorage.TaskDataStore.updateTimeForDifficulty ( String *difficulty*, int *newTime* ) [static]

This updates the estimated time for a particular difficulty. This method does not do error checking to make sure that the difficulty to change is a valid difficulty and the SQL update will fail if it is not a valid difficulty option in the Difficulty table

## Parameters

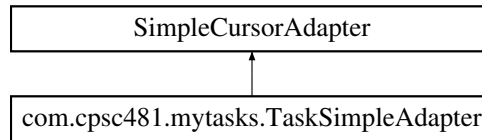
<i>difficulty</i>	The difficulty that you would like to update the estimated time of completion for.
<i>newTime</i>	The new estimated time for that particular difficulty.

The documentation for this class was generated from the following file:

- src/com/cpsc481/mytasks/databasesetorage/TaskDataStore.java

## 5.19 com.cpsc481.mytasks.TaskSimpleAdapter Class Reference

Inheritance diagram for com.cpsc481.mytasks.TaskSimpleAdapter:



## Classes

- class **TaskBinder**

## Public Member Functions

- **TaskSimpleAdapter** (Context context, int layout, Cursor c, String[] from, int[] to)

### 5.19.1 Detailed Description

This is a custom adapter based on SimpleCursorAdapter w/ a custom viewbinder in order to specifically display the integer date and time in a human readable format.

#### Author

rjbritt

The documentation for this class was generated from the following file:

- src/com/cpsc481/mytasks/TaskSimpleAdapter.java



# Index

- addNewTask
  - com::cpssc481::mytasks::databasestorage::Task-DataStore, [35](#)
- addTask
  - com::cpssc481::mytasks::CurrentTaskListFragment, [17](#)
- addTaskButton
  - com::cpssc481::mytasks::MainActivity, [22](#)
- buildBarDataset
  - com::cpssc481::mytasks::chart::AbstractChart, [10](#)
- buildBarRenderer
  - com::cpssc481::mytasks::chart::AbstractChart, [10](#)
- buildCategoryDataset
  - com::cpssc481::mytasks::chart::AbstractChart, [10](#)
- buildCategoryRenderer
  - com::cpssc481::mytasks::chart::AbstractChart, [10](#)
- buildDataset
  - com::cpssc481::mytasks::chart::AbstractChart, [10](#)
- buildDateDataset
  - com::cpssc481::mytasks::chart::AbstractChart, [11](#)
- buildMultipleCategoryDataset
  - com::cpssc481::mytasks::chart::AbstractChart, [11](#)
- buildRenderer
  - com::cpssc481::mytasks::chart::AbstractChart, [11](#)
- com.cpsc481.mytasks.AddTask, [12](#)
- com.cpsc481.mytasks.chart, [7](#)
- com.cpsc481.mytasks.chart.AbstractChart, [9](#)
- com.cpsc481.mytasks.chart.ChartInterface, [14](#)
- com.cpsc481.mytasks.chart.PieChart, [23](#)
- com.cpsc481.mytasks.CurrentTaskListFragment, [16](#)
- com.cpsc481.mytasks.CurrentTaskListFragment.-ChooseBeginOrEndFragment, [16](#)
- com.cpsc481.mytasks.customclasses.DateAndTime, [19](#)
- com.cpsc481.mytasks.customclasses.Task, [28](#)
- com.cpsc481.mytasks.databasestorage.TaskDataStore, [34](#)
- com.cpsc481.mytasks.databasestorage.TaskDatabase-Helper, [30](#)
- com.cpsc481.mytasks.DueDatePickerFragment, [19](#)
- com.cpsc481.mytasks.DueTimePickerFragment, [20](#)
- com.cpsc481.mytasks.Information, [21](#)
- com.cpsc481.mytasks.MainActivity, [21](#)
- com.cpsc481.mytasks.Reports, [24](#)
- com.cpsc481.mytasks.SetDifficultyTimeEstimate-Fragment, [26](#)
- com.cpsc481.mytasks.SetTimeFragment, [27](#)
- com.cpsc481.mytasks.TabListener< T extends Frag-ment >, [27](#)
- com.cpsc481.mytasks.TaskSimpleAdapter, [39](#)
- com::cpssc481::mytasks::AddTask
  - createTask, [13](#)
  - onCreate, [13](#)
  - onCreateOptionsMenu, [13](#)
  - showDatePicker, [14](#)
  - showTimePicker, [14](#)
- com::cpssc481::mytasks::CurrentTaskListFragment
  - addTask, [17](#)
  - fillData, [17](#)
  - onCreate, [18](#)
  - onListItemClick, [18](#)
  - removeTask, [18](#)
  - showDiffBegin, [18](#)
  - startTask, [18](#)
- com::cpssc481::mytasks::DueDatePickerFragment
  - onDateSet, [19](#)
  - setHandler, [20](#)
- com::cpssc481::mytasks::DueTimePickerFragment
  - onTimeSet, [20](#)
  - setHandler, [20](#)
- com::cpssc481::mytasks::MainActivity
  - addTaskButton, [22](#)
  - getInfo, [22](#)
  - showDiff, [22](#)
- com::cpssc481::mytasks::Reports
  - getAdviceLine, [25](#)
  - showCompletionChart, [25](#)
  - showDifficultyChart, [25](#)
  - showOnTimeChart, [25](#)
  - timesTakenPerDifficulty, [25](#)
  - underOverOrNearPerDifficulty, [26](#)
- com::cpssc481::mytasks::SetTimeFragment
  - onCreateDialog, [27](#)
- com::cpssc481::mytasks::TabListener< T extends Frag-ment >
  - TabListener, [28](#)
- com::cpssc481::mytasks::chart::AbstractChart
  - buildBarDataset, [10](#)
  - buildBarRenderer, [10](#)
  - buildCategoryDataset, [10](#)
  - buildCategoryRenderer, [10](#)
  - buildDataset, [10](#)
  - buildDateDataset, [11](#)
  - buildMultipleCategoryDataset, [11](#)
  - buildRenderer, [11](#)
  - setChartSettings, [12](#)
- com::cpssc481::mytasks::chart::ChartInterface
  - DESC, [15](#)

- execute, 15
- getDesc, 15
- getName, 15
- NAME, 15
- com::cpssc481::mytasks::chart::PieChart
  - execute, 23
  - getDesc, 23
  - getName, 24
- com::cpssc481::mytasks::customclasses::Task
  - getDifficulty, 29
  - getDueDate, 29
  - getDueTime, 29
  - getName, 29
  - getStartDate, 29
  - getStartTime, 29
  - setDifficulty, 30
  - setDueDate, 30
  - setDueTime, 30
  - setName, 30
  - setStartDate, 30
  - setStartTime, 30
- com::cpssc481::mytasks::databasestorage::TaskDataStore
  - addNewTask, 35
  - completeTask, 35
  - findAllTasks, 35
  - findSelectForTask, 35
  - findTask, 35
  - findTasksForDifficulty, 36
  - getCompletedOnTimeVsNot, 36
  - getDifficultyID, 36
  - getDueDateID, 36
  - getDueTimeID, 36
  - getEndDateID, 37
  - getEndTimeID, 37
  - getEstimatedTimeForDifficulty, 37
  - getStartDateID, 37
  - getStartTimeID, 38
  - getTaskNameID, 38
  - getTotalCompletedVsNot, 38
  - hasSetTime, 38
  - isComplete, 38
  - isTaskStarted, 39
  - startTask, 39
  - updateTimeForDifficulty, 39
- completeTask
  - com::cpssc481::mytasks::databasestorage::TaskDataStore, 35
- createTask
  - com::cpssc481::mytasks::AddTask, 13
- DESC
  - com::cpssc481::mytasks::chart::ChartInterface, 15
- execute
  - com::cpssc481::mytasks::chart::ChartInterface, 15
  - com::cpssc481::mytasks::chart::PieChart, 23
- fillData
  - com::cpssc481::mytasks::CurrentTaskListFragment, 17
- findAllTasks
  - com::cpssc481::mytasks::databasestorage::TaskDataStore, 35
- findSelectForTask
  - com::cpssc481::mytasks::databasestorage::TaskDataStore, 35
- findTask
  - com::cpssc481::mytasks::databasestorage::TaskDataStore, 35
- findTasksForDifficulty
  - com::cpssc481::mytasks::databasestorage::TaskDataStore, 36
- getAdviceLine
  - com::cpssc481::mytasks::Reports, 25
- getCompletedOnTimeVsNot
  - com::cpssc481::mytasks::databasestorage::TaskDataStore, 36
- getDesc
  - com::cpssc481::mytasks::chart::ChartInterface, 15
  - com::cpssc481::mytasks::chart::PieChart, 23
- getDifficulty
  - com::cpssc481::mytasks::customclasses::Task, 29
- getDifficultyID
  - com::cpssc481::mytasks::databasestorage::TaskDataStore, 36
- getDueDate
  - com::cpssc481::mytasks::customclasses::Task, 29
- getDueDateID
  - com::cpssc481::mytasks::databasestorage::TaskDataStore, 36
- getDueTime
  - com::cpssc481::mytasks::customclasses::Task, 29
- getDueTimeID
  - com::cpssc481::mytasks::databasestorage::TaskDataStore, 36
- getEndDateID
  - com::cpssc481::mytasks::databasestorage::TaskDataStore, 37
- getEndTimeID
  - com::cpssc481::mytasks::databasestorage::TaskDataStore, 37
- getEstimatedTimeForDifficulty
  - com::cpssc481::mytasks::databasestorage::TaskDataStore, 37
- getInfo
  - com::cpssc481::mytasks::MainActivity, 22
- getName
  - com::cpssc481::mytasks::chart::ChartInterface, 15
  - com::cpssc481::mytasks::chart::PieChart, 24
  - com::cpssc481::mytasks::customclasses::Task, 29
- getStartDate
  - com::cpssc481::mytasks::customclasses::Task, 29
- getStartDateID
  - com::cpssc481::mytasks::databasestorage::TaskDataStore, 37
- getStartTime
  - com::cpssc481::mytasks::customclasses::Task, 29

- com::cpssc481::mytasks::customclasses::Task, 29
- getStartTimeID
  - com::cpssc481::mytasks::databasestorage::Task-DataStore, 38
- getTaskNameID
  - com::cpssc481::mytasks::databasestorage::Task-DataStore, 38
- getTotalCompletedVsNot
  - com::cpssc481::mytasks::databasestorage::Task-DataStore, 38
- hasSetTime
  - com::cpssc481::mytasks::databasestorage::Task-DataStore, 38
- isComplete
  - com::cpssc481::mytasks::databasestorage::Task-DataStore, 38
- isTaskStarted
  - com::cpssc481::mytasks::databasestorage::Task-DataStore, 39
- NAME
  - com::cpssc481::mytasks::chart::ChartInterface, 15
- onCreate
  - com::cpssc481::mytasks::AddTask, 13
  - com::cpssc481::mytasks::CurrentTaskListFragment, 18
- onCreateDialog
  - com::cpssc481::mytasks::SetTimeFragment, 27
- onCreateOptionsMenu
  - com::cpssc481::mytasks::AddTask, 13
- onDateSet
  - com::cpssc481::mytasks::DueDatePickerFragment, 19
- onListItemClick
  - com::cpssc481::mytasks::CurrentTaskListFragment, 18
- onTimeSet
  - com::cpssc481::mytasks::DueTimePickerFragment, 20
- removeTask
  - com::cpssc481::mytasks::CurrentTaskListFragment, 18
- setChartSettings
  - com::cpssc481::mytasks::chart::AbstractChart, 12
- setDifficulty
  - com::cpssc481::mytasks::customclasses::Task, 30
- setDueDate
  - com::cpssc481::mytasks::customclasses::Task, 30
- setDueTime
  - com::cpssc481::mytasks::customclasses::Task, 30
- setHandler
  - com::cpssc481::mytasks::DueDatePickerFragment, 20
  - com::cpssc481::mytasks::DueTimePickerFragment, 20
- setName
  - com::cpssc481::mytasks::customclasses::Task, 30
- setStartDate
  - com::cpssc481::mytasks::customclasses::Task, 30
- setStartTime
  - com::cpssc481::mytasks::customclasses::Task, 30
- showCompletionChart
  - com::cpssc481::mytasks::Reports, 25
- showDatePicker
  - com::cpssc481::mytasks::AddTask, 14
- showDiff
  - com::cpssc481::mytasks::MainActivity, 22
- showDiffBegin
  - com::cpssc481::mytasks::CurrentTaskListFragment, 18
- showDifficultyChart
  - com::cpssc481::mytasks::Reports, 25
- showOnTimeChart
  - com::cpssc481::mytasks::Reports, 25
- showTimePicker
  - com::cpssc481::mytasks::AddTask, 14
- startTask
  - com::cpssc481::mytasks::CurrentTaskListFragment, 18
  - com::cpssc481::mytasks::databasestorage::Task-DataStore, 39
- TabListener
  - com::cpssc481::mytasks::TabListener< T extends Fragment >, 28
- timesTakenPerDifficulty
  - com::cpssc481::mytasks::Reports, 25
- underOverOrNearPerDifficulty
  - com::cpssc481::mytasks::Reports, 26
- updateTimeForDifficulty
  - com::cpssc481::mytasks::databasestorage::Task-DataStore, 39