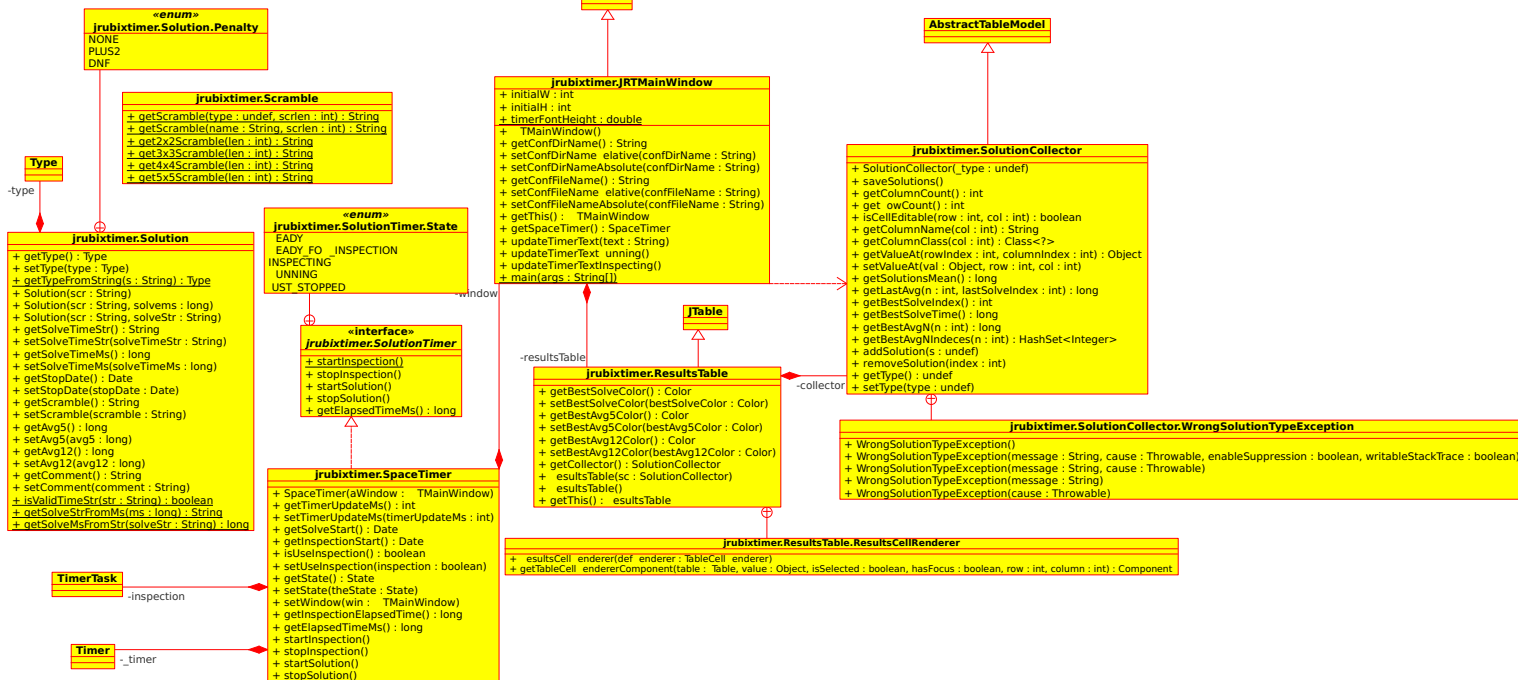
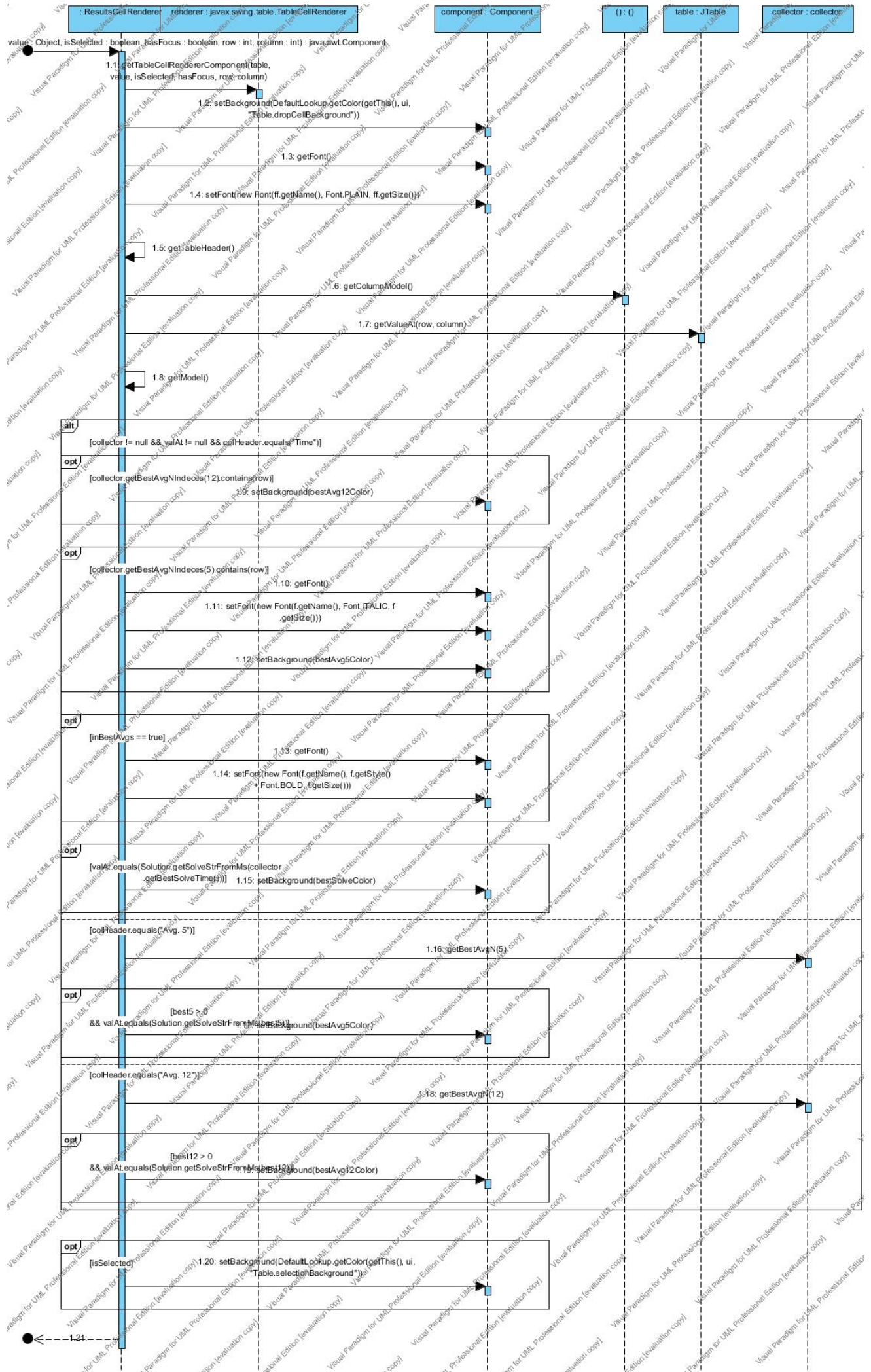


JRubixTimer

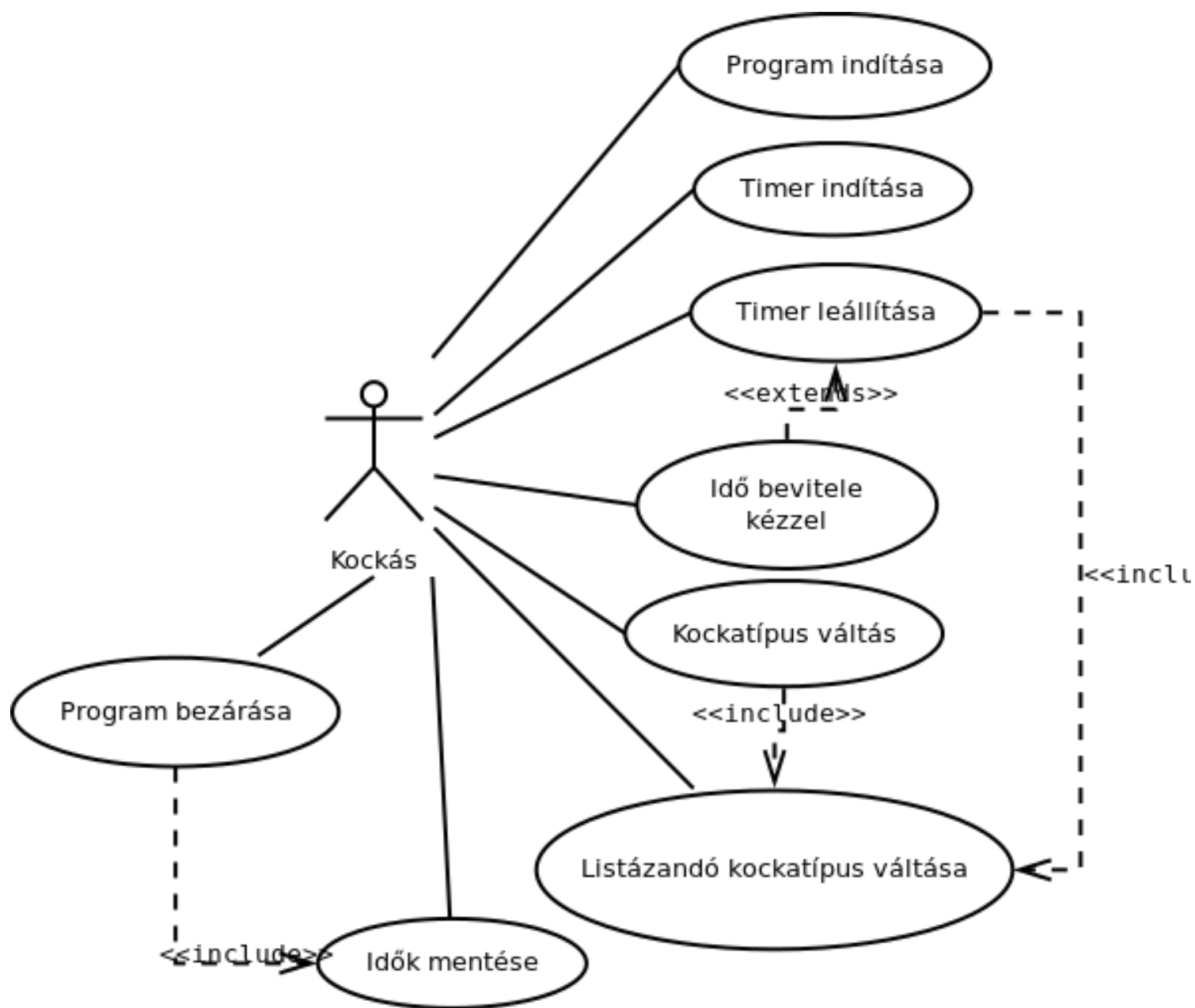
Készítette Doxygen 1.7.6.1

Tue Dec 3 2013 16:22:35





A program használatának use-case diagramja:



Tartalomjegyzék

1. Osztálymutató	1
1.1. Osztályhierarchia	1
2. Osztálymutató	3
2.1. Osztálylista	3
3. Osztályok dokumentációja	5
3.1. jrubixtimer.JRTMainWindow osztályreferencia	5
3.1.1. Konstruktorok és destruktork dokumentációja	6
3.1.1.1. JRTMainWindow	6
3.1.2. Tagfüggvények dokumentációja	6
3.1.2.1. getConfDirName	6
3.1.2.2. getConfFileName	6
3.1.2.3. getSpaceTimer	6
3.1.2.4. getThis	6
3.1.2.5. main	7
3.1.2.6. setConfDirNameAbsolute	7
3.1.2.7. setConfDirNameRelative	7
3.1.2.8. setConfFileNameAbsolute	7
3.1.2.9. setConfFileNameRelative	8
3.1.2.10. updateTimerText	8
3.1.2.11. updateTimerTextInspecting	8
3.1.2.12. updateTimerTextRunning	8
3.2. jrubixtimer.ResultsTable.ResultsCellRenderer osztályreferencia	8
3.2.1. Részletes leírás	9
3.2.2. Konstruktorok és destruktork dokumentációja	9

3.2.2.1.	ResultsCellRenderer	9
3.3.	jrubixtimer.ResultsTable osztályreferencia	9
3.3.1.	Részletes leírás	10
3.4.	jrubixtimer.Scramble osztályreferencia	10
3.4.1.	Részletes leírás	10
3.5.	jrubixtimer.Solution osztályreferencia	10
3.5.1.	Részletes leírás	11
3.5.2.	Konstruktorok és destruktorok dokumentációja	11
3.5.2.1.	Solution	11
3.5.2.2.	Solution	12
3.5.2.3.	Solution	12
3.5.3.	Tagfüggvények dokumentációja	12
3.5.3.1.	getSolveMsFromStr	12
3.5.3.2.	getSolveStrFromMs	12
3.5.3.3.	getType	13
3.5.3.4.	getTypeFromString	13
3.5.3.5.	isValidTimeStr	13
3.5.3.6.	setType	13
3.6.	jrubixtimer.SolutionCollector osztályreferencia	14
3.6.1.	Részletes leírás	15
3.6.2.	Konstruktorok és destruktorok dokumentációja	15
3.6.2.1.	SolutionCollector	15
3.6.3.	Tagfüggvények dokumentációja	15
3.6.3.1.	addSolution	15
3.6.3.2.	getBestAvgN	15
3.6.3.3.	getBestAvgNIndeces	16
3.6.3.4.	getBestSolveIndex	16
3.6.3.5.	getBestSolveTime	16
3.6.3.6.	getColumnClass	16
3.6.3.7.	getColumnCount	16
3.6.3.8.	getColumnName	17
3.6.3.9.	getLastAvg	17
3.6.3.10.	getRowCount	17
3.6.3.11.	getSolutionsMean	17

3.6.3.12.	getType	17
3.6.3.13.	getValueAt	18
3.6.3.14.	isCellEditable	18
3.6.3.15.	removeSolution	18
3.6.3.16.	saveSolutions	18
3.6.3.17.	setType	18
3.6.3.18.	setValueAt	19
3.6.4.	Adattagok dokumentációja	19
3.6.4.1.	solutions	19
3.7.	jrubixtimer.SolutionTimer interfészreferencia	19
3.7.1.	Részletes leírás	20
3.7.2.	Tagfüggvények dokumentációja	20
3.7.2.1.	getElapsedTimeMs	20
3.7.2.2.	startInspection	20
3.7.2.3.	startSolution	20
3.7.2.4.	stopInspection	20
3.7.2.5.	stopSolution	20
3.8.	jrubixtimer.SpaceTimer osztályreferencia	20
3.8.1.	Részletes leírás	21
3.8.2.	Konstruktorok és destruktorok dokumentációja	21
3.8.2.1.	SpaceTimer	21
3.8.3.	Tagfüggvények dokumentációja	22
3.8.3.1.	getElapsedTimeMs	22
3.8.3.2.	getInspectionElapsedTime	22
3.8.3.3.	getInspectionStart	22
3.8.3.4.	getSolveStart	22
3.8.3.5.	getState	22
3.8.3.6.	getTimerUpdateMs	23
3.8.3.7.	isUseInspection	23
3.8.3.8.	setState	23
3.8.3.9.	setTimerUpdateMs	23
3.8.3.10.	setUseInspection	23
3.8.3.11.	setWindow	24
3.8.3.12.	startInspection	24

3.8.3.13. startSolution	24
3.8.3.14. stopInspection	24
3.8.3.15. stopSolution	24
3.9. jrubixtimer.SolutionCollector.WrongSolutionTypeException osztályreferencia	25
3.9.1. Részletes leírás	25

1. fejezet

Osztálymutató

1.1. Osztályhierarchia

Majdnem (de nem teljesen) betűrendbe szedett leszármazási lista:

jrubixtimer.JRTMainWindow	5
jrubixtimer.ResultsTable.ResultsCellRenderer	8
jrubixtimer.ResultsTable	9
jrubixtimer.Scramble	10
jrubixtimer.Solution	10
jrubixtimer.SolutionCollector	14
jrubixtimer.SolutionTimer	19
jrubixtimer.SpaceTimer	20
jrubixtimer.SolutionCollector.WrongSolutionTypeException	25

2. fejezet

Osztálymutató

2.1. Osztálylista

Az összes osztály, struktúra, unió és interfész listája rövid leírásokkal:

jrubixtimer.JRTMainWindow	5
jrubixtimer.ResultsTable.ResultsCellRenderer	8
jrubixtimer.ResultsTable	9
jrubixtimer.Scramble	10
jrubixtimer.Solution	10
jrubixtimer.SolutionCollector	14
jrubixtimer.SolutionTimer	19
jrubixtimer.SpaceTimer	20
jrubixtimer.SolutionCollector.WrongSolutionTypeException	25

3. fejezet

Osztályok dokumentációja

3.1. jrubixtimer.JRTMainWindow osztályreferencia

Ősök: JFrame.

Publikus tagfüggvények

- [JRTMainWindow](#) ()
- String [getConfDirName](#) ()
- void [setConfDirNameRelative](#) (String confDirName)
- void [setConfDirNameAbsolute](#) (String confDirName)
- String [getConfFileName](#) ()
- void [setConfFileNameRelative](#) (String confFileName)
- void [setConfFileNameAbsolute](#) (String confFileName)
- [JRTMainWindow](#) [getThis](#) ()
- [SpaceTimer](#) [getSpaceTimer](#) ()
- void [updateTimerText](#) (String text)
- void [updateTimerTextRunning](#) ()
- void [updateTimerTextInspecting](#) ()

Statikus publikus tagfüggvények

- static void [main](#) (String[] args)

Publikus attribútumok

- int **initialW**
- int **initialH**

Statikus publikus attribútumok

- static final double **timerFontHeight** = 0.25

3.1.1. Konstruktorok és destruktorok dokumentációja

3.1.1.1. `jrubixtimer.JRTMainWindow.JRTMainWindow ()`

A fő ablak konstruktora. Beolvassa a beállításokat és az elmentett időket, valamint megjeleníti az ablakot.

3.1.2. Tagfüggvények dokumentációja

3.1.2.1. `String jrubixtimer.JRTMainWindow.getConfDirName ()`

A `confDirName` változó getter függvénye.

Visszatérési érték

A konfigurációs beállításokat tartalmazó könyvtár neve. Lehet abszolút vagy relatív is!!!

3.1.2.2. `String jrubixtimer.JRTMainWindow.getConfFileName ()`

A `confFileName` változó getter függvénye.

Visszatérési érték

A konfigurációs fájl neve. Lehet abszolút és relatív elérési út is!!

3.1.2.3. `SpaceTimer jrubixtimer.JRTMainWindow.getSpaceTimer ()`

Visszatérési érték

Az ablakhoz tartozó, space billentyűvel vezérelt timer.

3.1.2.4. `JRTMainWindow jrubixtimer.JRTMainWindow.getThis ()`

Kis hekkelés. A `this` "pointert" adja vissza. Erre akkor lehet szükség, ha az egyik komponens listenerének függvényeiben szeretnénk az osztályt elérni (mert ott ugye a `this` az magát a komponenst jelenti...)

Visszatérési érték

Az objektum referenciája.

3.1.2.5. static void jrubixtimer.JRTMainWindow.main (String[] args) [static]

A program belépési pontja. Megpróbálja beállítani az ablak megjelenését a rendszerére, majd megjeleníti a timer ablakát.

Paraméterek

<i>args</i>	Parancssori argumentumok, jelenleg nem kezelt.
-------------	--

3.1.2.6. void jrubixtimer.JRTMainWindow.setConfDirNameAbsolute (String confDirName)

A konfigurációs beállításokat tartalmazó mappa neve, abszolút elérési útvonallal.

Paraméterek

<i>confDir-Name</i>	A mappa neve, pl "/home/gipszjakab/.jrubixtimer", "C:\Users\Izidor\Settings"
---------------------	--

3.1.2.7. void jrubixtimer.JRTMainWindow.setConfDirNameRelative (String confDirName)

A konfigurációs beállításokat tartalmazó mappa neve, relatív elérési útvonallal a felhasználó home könyvtárán belül.

Paraméterek

<i>confDir-Name</i>	A mappa neve a home könyvtárban belül, pl ".jrubixtimer" vagy "-Beállítások"
---------------------	--

3.1.2.8. void jrubixtimer.JRTMainWindow.setConfFileNameAbsolute (String confFileName)

Beállítja a konfigurációs fájlt a megadott helyre, abszolút elérési útvonallal.

Paraméterek

<i>confFile-Name</i>	A konfigurációs fájl abszolút elérési útvonala, pl "/home/user/jrubixtimer.conf", vagy "C:\Users\Izidor\Asztal\jrubixtimer_beallitasok.txt"
----------------------	---

3.1.2.9. void jrubixtimer.JRTMainWindow.setConfFileNameRelative (String confFileName)

Beállítja a konfigurációs fájlt a megadott helyre, relatív elérési útvonallal a confDir mappán belül.

Paraméterek

<i>confFileName</i>	A konfigurációs fájl neve a confDir mappán belül, pl "jrt.rc", "sajat_jrubixtimer_beallitasok.conf" stb
---------------------	---

3.1.2.10. void jrubixtimer.JRTMainWindow.updateTimerText (String text)

Kényelmi funkció a timerText frissítésére.

Paraméterek

<i>text</i>	A kijelzett string
-------------	--------------------

3.1.2.11. void jrubixtimer.JRTMainWindow.updateTimerTextInspecting ()

Inspection alatt ezzel a függvénnyel frissíti a timer a kijelzett időt. 15-től számol vissza, 0 és -2 közt "+2"-t, utána DNF-et jelenít meg.

3.1.2.12. void jrubixtimer.JRTMainWindow.updateTimerTextRunning ()

Futás közben ezzel a függvénnyel frissíti a timer a kijelzett időt.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- JRTMainWindow.java

3.2. jrubixtimer.ResultsTable.ResultsCellRenderer osztályreferencia

Ősök: TableCellRenderer.

Publikus tagfüggvények

- [ResultsCellRenderer](#) (TableCellRenderer defRenderer)
- Component **getTableCellRendererComponent** (JTable table, Object value, boolean isSelected, boolean hasFocus, int row, int column)

3.2.1. Részletes leírás

Ez az osztály végzi a cellák kirajzolását. Hogy nekünk ne kelljen foglalkozni a low-level dolgokkal, eltárolunk egy TableCellRenderert, és az általa kirajzolt komponensnek állítjuk át a paramétereit.

3.2.2. Konstruktorok és destruktorok dokumentációja

3.2.2.1. jrubixtimer.ResultsTable.ResultsCellRenderer.ResultsCellRenderer (TableCellRenderer defRenderer)

Paraméterek

<i>defRenderer</i>	Egy TableCellRenderer, amely alapból kirajzolná a cellákat. - Jellemzően: aTable.setDefaultRenderer(valami.class, new ResultsCellRenderer(getDefaultRenderer(valami.class)));
--------------------	---

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- ResultsTable.java

3.3. jrubixtimer.ResultsTable osztályreferencia

Ősök: JTable.

Osztályok

- class ResultsCellRenderer

Publikus tagfüggvények

- Color **getBestSolveColor** ()
- void **setBestSolveColor** (Color bestSolveColor)
- Color **getBestAvg5Color** ()
- void **setBestAvg5Color** (Color bestAvg5Color)
- Color **getBestAvg12Color** ()
- void **setBestAvg12Color** (Color bestAvg12Color)
- SolutionCollector **getCollector** ()
- ResultsTable (SolutionCollector sc)
- ResultsTable **getThis** ()

3.3.1. Részletes leírás

Egy idők megjelenítésére alkalmas JTable osztály. TableModel-ként kötelezően egy - SolutionCollectort használ, és ennek a függvényeit hívogatva jeleníti meg az időket. A színeket/megjelenítési módokat a jövőben szeretném konfigurálhatóvá tenni. Addig is az alábbiak érvényesek: A legjobb idő háttérszíne zöld. A legjobb 5 átlag idejének háttérszíne világoskék. Ha több is van, az összes színes. A legutóbbi legjobb 5 átlagban lévő idők háttérszíne is világoskék, betűtípusuk dőlt és vastag. 12 átlagnál világoskék helyett világoslilát használok, és a betűtípus csak vastag, nem dőlt.

Csak az idők módosítására van lehetőség utólag, és módosításkor az átlagok újraszámolódnak. Új idő hozzáadáskor a lista aljára görgetünk.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- ResultsTable.java

3.4. jrubixtimer.Scramble osztályreferencia

Statikus publikus tagfüggvények

- static String **getScramble** (Solution.Type type, int scrLen)
- static String **getScramble** (String name, int scrLen)
- static String **get2x2Scramble** (int len)
- static String **get3x3Scramble** (int len)
- static String **get4x4Scramble** (int len)
- static String **get5x5Scramble** (int len)

3.4.1. Részletes leírás

A keveréseket generáló osztály, csak statikus függvényei vannak, hogy ne kelljen példányt létrehozni belőle. Jelenleg 2x2-5x5 kockákhoz tud keverést adni. Ezek pseudorandom generált keverések, tehát 2x2 és 3x3 esetén nem hivatalosak (ahhoz random state keverést kéne generálni, ami sokkal bonyolultabb folyamat.). A jövőben ezt az osztály brutális módon fel lehet okosítani és bővíteni is lehet újabb típusú keverésekkel (persze csak olyanokkal, amelyek benne vannak a Solution.Type enumban).

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- Scramble.java

3.5. jrubixtimer.Solution osztályreferencia

Osztályok

- enum **Penalty**
- enum **Type**

Publikus tagfüggvények

- Type `getType` ()
- void `setType` (Type type)
- `Solution` (String scr)
- `Solution` (String scr, long solvems)
- `Solution` (String scr, String solveStr)
- String `getSolveTimeStr` ()
- void `setSolveTimeStr` (String solveTimeStr)
- long `getSolveTimeMs` ()
- void `setSolveTimeMs` (long solveTimeMs)
- Date `getStopDate` ()
- void `setStopDate` (Date stopDate)
- String `getScramble` ()
- void `setScramble` (String scramble)
- long `getAvg5` ()
- void `setAvg5` (long avg5)
- long `getAvg12` ()
- void `setAvg12` (long avg12)
- String `getComment` ()
- void `setComment` (String comment)

Statikus publikus tagfüggvények

- static Solution.Type `getTypeFromString` (String s)
- static boolean `isValidTimeStr` (String str)
- static String `getSolveStrFromMs` (long ms)
- static long `getSolveMsFromStr` (String solveStr)

3.5.1. Részletes leírás

A kirakást tároló osztály. Jelenleg a típust, kirakás idejét, a keverést, a hozzá tartozó kommentet, valamint az ezzel az idővel befejeződő 5 és 12 rakás átlagát tárolja.

Az osztálynak vannak statikus függvényei is, ezek segítik az idő konvertálását szöveggé és vissza, stb.

3.5.2. Konstruktorok és destruktorok dokumentációja

3.5.2.1. jrubixtimer.Solution.Solution (String scr)

Konstruktor, amely csak beállítja a keverést a kirakáshoz. A többi tulajdonságot a setter függvényekkel kell beállítani.

Paraméterek

<code>scr</code>	A keverés.
------------------	------------

3.5.2.2. `jrubixtimer.Solution.Solution (String scr, long solvems)`

Konstruktor, amely beállítja a keverést és a kirakási időt.

Paraméterek

<code>scr</code>	A keverés
<code>solvems</code>	A kirakás ideje, ezredmásodpercben.

3.5.2.3. `jrubixtimer.Solution.Solution (String scr, String solveStr)`

Konstruktor, amely beállítja a keverést és a kirakási időt egy string alapján.

Paraméterek

<code>scr</code>	A keverés
<code>solveStr</code>	A kirakás ideje, szöveges formában. A formátumot nem ellenőrzi a függvény!

3.5.3. Tagfüggvények dokumentációja

3.5.3.1. `static long jrubixtimer.Solution.getSolveMsFromStr (String solveStr)` [static]

Statikus segédfüggvény. A stringként reprezentált időt átkonvertálja milliszekundumokká.

Paraméterek

<code>solveStr</code>	Az idő reprezentációja szöveggént. Nincs formátumellenőrzés!
-----------------------	--

Visszatérési érték

Az idő ezredmásodpercekben.

3.5.3.2. `static String jrubixtimer.Solution.getSolveStrFromMs (long ms)` [static]

Statikus segédfüggvény. Átkonvertálja a milliszekundumban lévő időt stringgé. Ha `ms = -1`, akkor "N/A"-t ad vissza, ha `-2`, akkor az DNF-et jelent. Nem szálbiztos!!! (A SimpleDateFormat használata miatt)

Paraméterek

<code>ms</code>	A kirakás ideje ezredmásodpercben.
-----------------	------------------------------------

Visszatérési érték

Az idő reprezentációja stringben.

3.5.3.3. Type jrubixtimer.Solution.getType ()

Visszatérési érték

A kirakáshoz tartozó kocka típusa.

3.5.3.4. static Solution.Type jrubixtimer.Solution.getTypeFromString (String s)
[static]

Egy inverz toString(), megadja a paraméterként kapott stringről, hogy milyen típus tartozik hozzá. Ha nem talál, null-t ad vissza.

Paraméterek

s	A típus neve
---	--------------

Visszatérési érték

A paraméterhez tartozó típus, vagy null.

3.5.3.5. static boolean jrubixtimer.Solution.isValidTimeStr (String str) [static]

Statikus segédfüggvény. Megmondja, hogy a paraméterként kapott string teljesíti-e az alábbi feltételeket: 1) Az óra, perc megadása opcionális, de ha mindekető meg van adva, kettőspont van köztük. 2) Ha csak a perc van megadva pluszban, akkor közte és a másodperc közt kettőspont van. 3) A másodpercet ha megadjuk, akkor ponttal elválasztva tőle legalább egy tizedesjegyig (tizedmp) meg kell adni a törtmásodpercet is. 4) Ha simán csak egy számot adunk meg, akkor a program azt ezredmásodpercként értelmezi.

Paraméterek

str	Az időt tartalmazó string
-----	---------------------------

Visszatérési érték

A megadott string formátuma érvényes-e a fentebb említett szempontok alapján.

3.5.3.6. void jrubixtimer.Solution.setType (Type type)

Beállítja a kirakáshoz tartozó játék típusát.

Paraméterek

<i>type</i>	A típus.
-------------	----------

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- `Solution.java`

3.6. `jrubixtimer.SolutionCollector` osztályreferencia

Ősök: `AbstractTableModel`.

Osztályok

- class [WrongSolutionTypeException](#)

Publikus tagfüggvények

- [SolutionCollector](#) (`Solution.Type _type`)
- void [saveSolutions](#) ()
- int [getColumnCount](#) ()
- int [getRowCount](#) ()
- boolean [isCellEditable](#) (int row, int col)
- String [getColumnName](#) (int col)
- Class<?> [getColumnClass](#) (int col)
- Object [getValueAt](#) (int rowIndex, int columnIndex)
- void [setValueAt](#) (Object val, int row, int col)
- long [getSolutionsMean](#) ()
- long [getLastAvg](#) (int n, int lastSolveIndex)
- int [getBestSolveIndex](#) ()
- long [getBestSolveTime](#) ()
- long [getBestAvgN](#) (int n)
- HashSet< Integer > [getBestAvgNIndeces](#) (int n)
- void [addSolution](#) ([Solution](#) s) throws `WrongSolutionTypeException`
- void [removeSolution](#) (int index)
- `Solution.Type` [getType](#) ()
- void [setType](#) (`Solution.Type` type)

Csomag attribútumok

- List< [Solution](#) > [solutions](#) = new ArrayList<[Solution](#)>()

3.6.1. Részletes leírás

A kirakásokat gyűjtő, valamint azokat modellbe rendező osztály. A [ResultsTable](#) osztályt használjuk a megjelenítésére.

3.6.2. Konstruktorok és destruktorok dokumentációja

3.6.2.1. jrubixtimer.SolutionCollector.SolutionCollector (Solution.Type _type)

Konstruktor, amely beolvassa a futási könyvtárban lévő jrt_stats.csv fájlból az esetleges előzőleg kimentett időket, és el is tárolj azokat. Magyarul ugyan onnan folytathatjuk a kockázást, ahol abbahagytuk.

Paraméterek

<code>_type</code>	A kirakások típusa.
--------------------	---------------------

3.6.3. Tagfüggvények dokumentációja

3.6.3.1. void jrubixtimer.SolutionCollector.addSolution (Solution s) throws WrongSolutionTypeException

A lista végére fűz egy kirakást, újraszámolja az átlagokat és frissíti a megjelenített táblázatot.

Paraméterek

<code>s</code>	Egy inicializált Solution , jellemzően a legutóbbi rakás adataival.
----------------	---

Kivételek

WrongSolutionTypeException	
--	--

3.6.3.2. long jrubixtimer.SolutionCollector.getBestAvgN (int n)

Megadja a legjobb n-átlagot a rakásokból.

Paraméterek

<code>n</code>	Hány rakás átlagaiból adja meg a legjobbat. Jelenleg 5 vagy 12 lehet csak!
----------------	--

Visszatérési érték

A legjobb n-átlag, ezredmásodpercben.

3.6.3.3. `HashSet<Integer> jrubixtimer.SolutionCollector.getBestAvgNIndices (int n)`

Ha több helyen is ugyanannyi az *n*-átlag, mint a legjobbnak, akkor ez a függvény megadja az összes ilyen hely indexét.

Paraméterek

<i>n</i>	Hány rakás átlagaiból adja meg a legjobbakat. Jelenleg 5 vagy 12 lehet csak!
----------	--

Visszatérési érték

Az indexek egy HashSet-ben.

3.6.3.4. `int jrubixtimer.SolutionCollector.getBestSolveIndex ()`

Visszatérési érték

A legjobb idő indexe.

3.6.3.5. `long jrubixtimer.SolutionCollector.getBestSolveTime ()`

Visszatérési érték

A legjobb idő, ezredmásodpercben.

3.6.3.6. `Class<?> jrubixtimer.SolutionCollector.getColumnClass (int col)`

Az oszlopok osztálya, jelenleg mindenhol String.

Lásd még

`javax.swing.table.AbstractTableModel::getColumnClass(int)`

3.6.3.7. `int jrubixtimer.SolutionCollector.getColumnCount ()`

Az oszlopok száma jelenleg 4: Idő, 5 átlag, 12 átlag, komment. Ha ez a jövőben változik, akkor elég sok mindent meg kell még változtatni ezen a függvényen kívül is!

Lásd még

`javax.swing.table.TableModel::getColumnCount()`

3.6.3.8. String jrubixtimer.SolutionCollector.getColumnName (int col)

Az oszlopok headerjei adja vissza. Ez jelenleg hard-code-olva van, a jövőben szeretném konfigurálhatóvá tenni.

Lásd még

```
javax.swing.table.AbstractTableModel::getColumnName(int)
```

3.6.3.9. long jrubixtimer.SolutionCollector.getLastAvg (int n, int lastSolveIndex)

Megadja az utolsó n idő átlagát (legjobb és legrosszabb idő nélkül számolva) a lastSolveIndex-edik időig bezárólag. Értelmszerű okokból $n \geq 3$. Jelenleg jellemzően $n = 5$ vagy $n = 12$.

Paraméterek

n	Hány idő átlagát szeretnénk tudni.
<i>lastSolveIndex</i>	Az átlag utolsó idejének indexe

Visszatérési érték

Az átlag, ezredmásodpercben.

3.6.3.10. int jrubixtimer.SolutionCollector.getRowCount ()

A táblázatban lévő sorok számát adja vissza (ez a kirakások száma).

Lásd még

```
javax.swing.table.TableModel::getRowCount()
```

3.6.3.11. long jrubixtimer.SolutionCollector.getSolutionsMean ()

Megadja az összes rakás átlagát, a legjobb és legrosszabb idővel együtt számolva. Csak akkor lesz ez DNF, ha _csak_ DNF idők vannak.

Visszatérési érték

Az összes rakás átlaga.

3.6.3.12. Solution.Type jrubixtimer.SolutionCollector.getType ()

Visszatérési érték

A tárolt idők típusa.

3.6.3.13. Object `jrubixtimer.SolutionCollector.getValueAt (int rowIndex, int columnIndex)`

A cellák értékét adja meg. Mindig a *rowIndex*-edik időnek adja meg az idejét, 5 és 12 átlagát, valamint a hozzá tartozó kommentet, szöveges formában.

Lásd még

`javax.swing.table.TableModel::getValueAt(int, int)`

3.6.3.14. boolean `jrubixtimer.SolutionCollector.isCellEditable (int row, int col)`

Az első és a negyedik oszlop szerkeszthetőek csak. (Idő és komment)

Lásd még

`javax.swing.table.AbstractTableModel::isCellEditable(int, int)`

3.6.3.15. void `jrubixtimer.SolutionCollector.removeSolution (int index)`

Kitörli a megadott indexű időt. Újrászámolja az átlagokat, frissíti a megjelenített táblázatot.

Paraméterek

<i>index</i>	A törölni kívánt kirakás indexe.
--------------	----------------------------------

3.6.3.16. void `jrubixtimer.SolutionCollector.saveSolutions ()`

Kimentti az időket a `jrt_stats.csv` fájlba. Azért nem szerializál, hogy utólag bármikor bemásolhassuk az időket keveréssel, mindenestül bármilyen fórumra, facebookra stb.

3.6.3.17. void `jrubixtimer.SolutionCollector.setType (Solution.Type type)`

Beállítja a tárolt idők típusát. Jelenleg ezt nem nagyon kéne használni, ugyanis az az eset nincs lekezelve, hogy már tárol néhány A típust, és átállítjuk a típust B-re.

Paraméterek

<i>type</i>	Az új típus.
-------------	--------------

3.6.3.18. void jrubixtimer.SolutionCollector.setValueAt (Object *val*, int *row*, int *col*)

Az adott cella értékét állítja be. Ha az időt szeretnénk átírni kézzel utólag, ez a függvény ellenőrzi a formátumát.

Lásd még

```
javax.swing.table.AbstractTableModel::setValueAt(java.lang.Object, int, int)
```

3.6.4. Adattagok dokumentációja

3.6.4.1. List<Solution> jrubixtimer.SolutionCollector.solutions = new ArrayList<Solution>() [package]

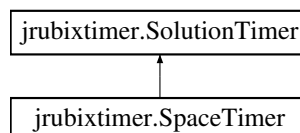
A kirakásokat tároló lista. Csak egyféle típusú játékhoz tartozó kirakásokat tartalmazhat.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- SolutionCollector.java

3.7. jrubixtimer.SolutionTimer interfészreferencia

A jrubixtimer.SolutionTimer osztály származási diagramja:



Osztályok

- enum **State**

Publikus tagfüggvények

- void [startInspection](#) ()
- void [stopInspection](#) ()
- void [startSolution](#) ()
- void [stopSolution](#) ()
- long [getElapsedTimeMs](#) ()

3.7.1. Részletes leírás

Az idő/inspection mérésére alkalmas interfész.

3.7.2. Tagfüggvények dokumentációja

3.7.2.1. `long jrubixtimer.SolutionTimer.getElapsedTimeMs ()`

Az eltelt idő lekérdezésére szolgáló függvény.

Visszatérési érték

Az indítás óta eltelt idő ezredmásodpercekben.

Megvalósítják a következők: [jrubixtimer.SpaceTimer](#).

3.7.2.2. `void jrubixtimer.SolutionTimer.startInspection ()`

Az inspection indítására szolgáló függvény.

Megvalósítják a következők: [jrubixtimer.SpaceTimer](#).

3.7.2.3. `void jrubixtimer.SolutionTimer.startSolution ()`

A tényleges időmérés indítására szolgáló függvény.

Megvalósítják a következők: [jrubixtimer.SpaceTimer](#).

3.7.2.4. `void jrubixtimer.SolutionTimer.stopInspection ()`

Az inspection leállítására szolgáló függvény.

Megvalósítják a következők: [jrubixtimer.SpaceTimer](#).

3.7.2.5. `void jrubixtimer.SolutionTimer.stopSolution ()`

Az időmérés leállítására szolgáló függvény.

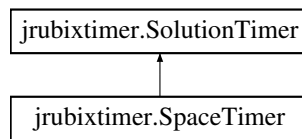
Megvalósítják a következők: [jrubixtimer.SpaceTimer](#).

Ez a dokumentáció az interfészeiről a következő fájl alapján készült:

- `SolutionTimer.java`

3.8. `jrubixtimer.SpaceTimer` osztályreferencia

A `jrubixtimer.SpaceTimer` osztály származási diagramja:



Publikus tagfüggvények

- [SpaceTimer](#) ([JRTMainWindow](#) [aWindow](#))
- int [getTimerUpdateMs](#) ()
- void [setTimerUpdateMs](#) (int timerUpdateMs)
- Date [getSolveStart](#) ()
- Date [getInspectionStart](#) ()
- boolean [isUseInspection](#) ()
- void [setUseInspection](#) (boolean inspection)
- State [getState](#) ()
- void [setState](#) (State theState)
- void [setWindow](#) ([JRTMainWindow](#) win)
- long [getInspectionElapsedTime](#) ()
- long [getElapsedTimeMs](#) ()
- void [startInspection](#) ()
- void [stopInspection](#) ()
- void [startSolution](#) ()
- void [stopSolution](#) ()

3.8.1. Részletes leírás

Az kirakás idejét/inspectiont mérő timer osztály. A név picit csalóka, ugyanis az osztály nem tud róla, hogy őt mi vezérli, space gombnyomások, egérkattintások vagy cuki kismacskák pislogásai.

Hogy tudja frissíteni az ablakon kijelzett időt, meg kell kapnia egy [JRTMainWindow](#) objektumot.

3.8.2. Konstruktorok és destruktorok dokumentációja

3.8.2.1. jrubixtimer.SpaceTimer.SpaceTimer ([JRTMainWindow](#) [aWindow](#))

Konstruktor, alapértelmezetten 1 ms frissítési időközt állít be, és használ inspectiont.

Paraméterek

aWindow	Az időt megjelenítő ablak.
-------------------------	----------------------------

3.8.3. Tagfüggvények dokumentációja

3.8.3.1. long jrubixtimer.SpaceTimer.getElapsedTimeMs ()

Megadja a timer indítása óta eltelt időt. Ha a timert már leállították, akkor a futási idejét adja meg.

Lásd még

[jrubixtimer.SolutionTimer::getElapsedTimeMs\(\)](#)

Megvalósítja a következőket: [jrubixtimer.SolutionTimer](#).

3.8.3.2. long jrubixtimer.SpaceTimer.getInspectionElapsedTime ()

Megadja, hogy mennyi idő telt el az inspection kezdete óta.

Visszatérési érték

Az eltelt idő ezredmásodpercekben.

3.8.3.3. Date jrubixtimer.SpaceTimer.getInspectionStart ()

Megadja, hogy mikor kezdődött az inspection.

Visszatérési érték

Az inspection kezdetének ideje.

3.8.3.4. Date jrubixtimer.SpaceTimer.getSolveStart ()

Megadja, hogy mikor kezdődött a kirakás.

Visszatérési érték

A kirakás kezdetének ideje.

3.8.3.5. State jrubixtimer.SpaceTimer.getState ()

Visszatérési érték

A timer állapota.

3.8.3.6. int jrubixtimer.SpaceTimer.getTimerUpdateMs ()

Ezzel a függvénnyel lehet lekérdezni a timer frissítési időközét.

Visszatérési érték

A timer frissítési időköze, ezredmásodpercben.

3.8.3.7. boolean jrubixtimer.SpaceTimer.isUseInspection ()

Megadja, hogy a timer használ-e inspectiont.

Visszatérési érték

A useInspection értéke.

3.8.3.8. void jrubixtimer.SpaceTimer.setState (State theState)

A timer állapotát állítja be a kívánt értékre.

Paraméterek

<i>theState</i>	A beállítandó állapot.
-----------------	------------------------

3.8.3.9. void jrubixtimer.SpaceTimer.setTimerUpdateMs (int timerUpdateMs)

Beállítja a timer frissítési időközét.

Paraméterek

<i>timer-UpdateMs</i>	A kívánt idő, ezredmásodpercben.
-----------------------	----------------------------------

3.8.3.10. void jrubixtimer.SpaceTimer.setUseInspection (boolean inspection)

Beállítja, hogy a timer használjon-e inspectiont. A timer állapotát is átállítja, ha kell (-READY és READY_FOR_INSPECTION közt). Tipp: A timer alapállapotba állítása a következőképpen is elérhető: aTimer.setUseInspection(aTimer.isUseInspection());

Paraméterek

<i>inspection</i>	Használjon-e inspectiont a timer.
-------------------	-----------------------------------

3.8.3.11. void jrubixtimer.SpaceTimer.setWindow (JRTMainWindow win)

Ha nem a konstruktorban adtuk meg az ablakot, akkor ezzel a függvénnyel megtehetjük.

Paraméterek

<i>win</i>	Az ablak.
------------	-----------

3.8.3.12. void jrubixtimer.SpaceTimer.startInspection ()

Lásd még

[jrubixtimer.SolutionTimer::startInspection\(\)](#)

Megvalósítja a következőket: [jrubixtimer.SolutionTimer](#).

3.8.3.13. void jrubixtimer.SpaceTimer.startSolution ()

Elindítja az időmérést. Ha a frissítési időköz 0, akkor a window timerText-jén nem frissíti az időt, csak beállítja a szövegét "Running"-ra.

Lásd még

[jrubixtimer.SolutionTimer::startSolution\(\)](#)

Megvalósítja a következőket: [jrubixtimer.SolutionTimer](#).

3.8.3.14. void jrubixtimer.SpaceTimer.stopInspection ()

Az inspection leállítására szolgáló függvény.

Megvalósítja a következőket: [jrubixtimer.SolutionTimer](#).

3.8.3.15. void jrubixtimer.SpaceTimer.stopSolution ()

Leállítja a kirakást. Még egyszer lefrissíti az eltelt időt is.

Lásd még

[jrubixtimer.SolutionTimer::stopSolution\(\)](#)

Megvalósítja a következőket: [jrubixtimer.SolutionTimer](#).

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- SpaceTimer.java

3.9. jrubixtimer.SolutionCollector.WrongSolutionTypeException osztályreferencia

Publikus tagfüggvények

- **WrongSolutionTypeException** (String message, Throwable cause, boolean enableSuppression, boolean writableStackTrace)
- **WrongSolutionTypeException** (String message, Throwable cause)
- **WrongSolutionTypeException** (String message)
- **WrongSolutionTypeException** (Throwable cause)

3.9.1. Részletes leírás

Külön exception arra az esetre, ha rossz típusú rakást próbálnánk hozzáadni.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

- SolutionCollector.java