Minesweeper

Generated by Doxygen 1.15.0

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

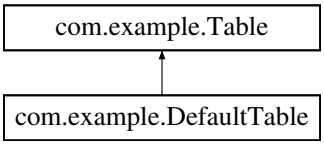
Chapter 3

Class Documentation

3.1 com.example.DefaultTable Class Reference

A klasszikus aknakereső játékmód tábláját megvalósító osztály.

Inheritance diagram for com.example.DefaultTable:



Public Member Functions

- **DefaultTable** (int rows, int columns, int allMines)
- void **checkNeighbors** ()
- List< **Position** > **getNeighborPositions** (**Position** p)
- void **selectingMines** (ArrayList< **Field** > availableFields)
- int **getOneMineFields** ()

Public Member Functions inherited from com.example.Table

- **Table** (int rowNumber, int columnNumber, int mineNumber)
- **Field** **getFieldByPosition** (**Position** p)
- **Position** **getPositionByField** (**Field** f)
- List< **Position** > **getNeighbors** (**Position** currentPosition)
- ArrayList< **Field** > **getavailableFields** (**Position** selectedPosition)
- int **getAllMines** ()
- int **getRows** ()
- int **getColumns** ()

Additional Inherited Members

Protected Attributes inherited from com.example.Table

- int **rows**
- int **columns**
- List< **Field** > **fields** = new ArrayList< **Field**>()
- int **allMines**

3.1.1 Detailed Description

A klasszikus aknakereső játékmód tábláját megvalósító osztály.

A **DefaultTable** (p. 5) a **Table** (p. 53) absztrakt osztály konkrét implementációja, amely a hagyományos aknakereső szabályai szerint kezeli a mezőket. Feladata a bombák véletlenszerű elhelyezése, a szomszédos bombák számának kiszámítása, valamint a szomszédos pozíciók meghatározása.

Ez az osztály felel a "normál" játékmód működéséért, ahol minden bomba egyetlen egységnyi értékkel szerepel a mezőkön.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 DefaultTable()

```
com.example.DefaultTable.DefaultTable (
            int rows,
            int columns,
            int allMines)
```

A **DefaultTable** (p. 5) meghívja az ősosztálya, a **Table** (p. 53) konstruktorát.

Parameters

rows	sorok száma
columns	oszlopok száma
allMines	bombák száma

3.1.3 Member Function Documentation

3.1.3.1 checkNeighbors()

```
void com.example.DefaultTable.checkNeighbors ()
```

Ellenőrzi, hogy egy mező szomszédjaiban összesen hány bomba található. Azon mezők vannak ellenőrizve, amelyek nem tartalmaznak bombák. Ha egy mező szomszédja tartalmaz bombát növeli a mezővel szomszédos bombák számlálóját eggyel. A mező szomszédos bombáinak száma a számláló értéke lesz

Reimplemented from **com.example.Table** (p. 55).

3.1.3.2 getNeighborPositions()

```
List< Position > com.example.DefaultTable.getNeighborPositions (
            Position p)
```

Ez a metódus visszatér egy mező szomszédjainak listájával. A getNeighbors metódust hívja meg a szomszédok listájának eléréséhez.

Parameters

p	jelenlegi pozíció, melynek szeretnénk megtudni a szomszédjait

Returns

szomszédpozíciók listája

Reimplemented from **com.example.Table** (p. 56).

3.1.3.3 getOneMineFields()

```
int com.example.DefaultTable.getOneMineFields ()
```

A metódus visszatér az egyetlen bombát tartalmazó mezők számával

Returns

az egyetlen bombát tartalmazó mezők száma

Reimplemented from **com.example.Table** (p. 58).

3.1.3.4 selectingMines()

```
void com.example.DefaultTable.selectingMines (
            ArrayList< Field > availableFields)
```

Kiválasztja azokat a mezőket, amelyek bombát tartalmaznak. A bombát tartalmazó mezők között nem lehet olyan, ami az elsőként kiválasztott pozíció szomszédja. Az elérhető mezők össze vannak keverve annak érdekében , hogy a táblán véletlenszerűen szerepljenek a bombák. A táblához tartozó bombák számával egyenlő mezőnek beállítja a hozzá tartozó bombák számát 1-re.

Parameters

availableFields	az elérhető pozíciók

Reimplemented from **com.example.Table** (p. 58).

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

- demo/src/main/java/com/example/DefaultTable.java

3.2 com.example.DefaultTableTest Class Reference

A **DefaultTable** (p. 5) osztály működését ellenőrző JUnit tesztosztály.

Public Member Functions

- void **testDefaultTableConstructor** ()
- void **testGetAvailableFields** ()
- void **testSelectingMines** ()
- void **testCheckNeighbors** ()
- void **testGetNeighborPositions** ()
- void **testGetFieldByPosition** ()
- void **testGetPositionByField** ()

3.2.1 Detailed Description

A **DefaultTable** (p. 5) osztály működését ellenőrző JUnit tesztosztály.

Teszteli többek között a konstruktor működését, a bombák helyes kiválasztását, a szomszédszámolást és a pozíció-mező kapcsolat helyes működését.

3.2.2 Member Function Documentation

3.2.2.1 testCheckNeighbors()

```
void com.example.DefaultTableTest.testCheckNeighbors ()
```

Teszteli, hogy a selectingMines a megadott elérhető mezők közül pontosan a kívánt számú mezőt jelöl ki bányaként.

3.2.2.2 testDefaultTableConstructor()

```
void com.example.DefaultTableTest.testDefaultTableConstructor ()
```

Teszteli, hogy a **DefaultTable** (p. 5) konstruktora helyesen állítja be a sorok, oszlopok, összes bánya és az egyaknás mezők számát.

3.2.2.3 testGetAvailableFields()

```
void com.example.DefaultTableTest.testGetAvailableFields ()
```

Teszteli, hogy a getAvailableFields nem ad vissza olyan mezőt, amely a kiválasztott pozícióban van, vagy annak szomszédja.

3.2.2.4 testGetFieldByPosition()

```
void com.example.DefaultTableTest.testGetFieldByPosition ()
```

Teszteli, hogy különböző pozíciókhoz külön **Field** (p. 11) objektum tartozik, vagyis nem ugyanazt az példányt adja vissza.

3.2.2.5 testGetNeighborPositions()

```
void com.example.DefaultTableTest.testGetNeighborPositions ()
```

Teszteli, hogy a getNeighborPositions megfelelő számú szomszédos pozíciót ad vissza sarok-, szél- és belső mezőre.

3.2.2.6 testGetPositionByField()

```
void com.example.DefaultTableTest.testGetPositionByField ()
```

Teszteli, hogy különböző pozíciókhoz külön **Field** (p. 11) objektum tartozik, vagyis nem ugyanazt az instance-t adja vissza.

3.2.2.7 testSelectingMines()

```
void com.example.DefaultTableTest.testSelectingMines ()
```

Teszteli, hogy a selectingMines a megadott elérhető mezők közül pontosan a kívánt számú mezőt jelöl ki bombaként.

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

- demo/src/test/java/com/example/DefaultTableTest.java

3.3 com.example.Difficulties Enum Reference

A játék nehézségi szintjeit reprezentáló enumeráció.

Public Attributes

- **EASY**
- **MEDIUM**
- **HARD**

3.3.1 Detailed Description

A játék nehézségi szintjeit reprezentáló enumeráció.

Három lehetséges értéke van:

- EASY: könnyű nehézség
- MEDIUM: közepes nehézség
- HARD: nehéz nehézség

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

- demo/src/main/java/com/example/Difficulties.java

3.4 com.example.Field Class Reference

A **Field** (p. 11) osztály a játék egy mezőjét reprezentálja.

Public Member Functions

- **Field** (boolean isMine)
- void **incrementFlags** (**GameModes** mode)
- void **resetFlags** ()
- int **maxFlag** (**GameModes** mode)
- void **setMineNumber** (int number)
- int **getMineNumber** ()
- void **setNumberOfNeighbors** (int number)
- int **getNumberOfNeighbors** ()
- boolean **getIsMine** ()
- int **getFlags** ()
- boolean **getFlagged** ()
- boolean **getRevealed** ()
- void **setRevealed** (boolean revealed)
- void **setFlagged** (boolean flagged)

3.4.1 Detailed Description

A **Field** (p. 11) osztály a játék egy mezőjét reprezentálja.

Egy mező tárolja, hogy tartalmaz-e bombát, hány szomszédja tartalmaz bombát, hány zászlóval van megjelölve, illetve hogy fel van-e fedve. A mező kezeli a zászlók számának növelését, visszaállítását, valamint a bomba- és szomszédinformációkat.

A **Field** (p. 11) objektumok a játéktábla elemi egységei, amelyeken keresztül a játékos jelölhet, felfedhet és információt kaphat a környező mezőkről.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 Field()

```
com.example.Field.Field (
            boolean isMine)
```

A **Field** (p. 11) osztály konstruktora.

Parameters

isMine	megadja, hogy a mező bomba-e, vagy sem.

3.4.3 Member Function Documentation

3.4.3.1 getFlagged()

```
boolean com.example.Field.getFlagged ()
```

Visszatér azzal a logikai értékkel, hogy a mező be van e jelölve zászlóként.

Returns

bejelölt-e

3.4.3.2 getFlags()

```
int com.example.Field.getFlags ()
```

Visszatér a mezőhöz tartozó zászlók számával.

Returns

zászlók száma.

3.4.3.3 getIsMine()

```
boolean com.example.Field.getIsMine ()
```

Visszatér azzal a logikai értékkel, hogy a mező tartalmaz-e bombát vagy se.

Returns

bomba-e

3.4.3.4 getMineNumber()

```
int com.example.Field.getMineNumber ()
```

A függvény a bombák számát téríti vissza.

Returns

bombák száma

3.4.3.5 getNumberOfNeighbors()

```
int com.example.Field.getNumberOfNeighbors ()
```

Visszatér a mezővel szomszédos, bombát tartalmazó mezők számával.

Returns

bombát tartlmazó szomszédok száma

3.4.3.6 getRevealed()

```
boolean com.example.Field.getRevealed ()
```

Visszatér azzal a logikai értékkel, hogy a mező fel van e fedve már.

Returns

felfedve-e

3.4.3.7 incrementFlags()

```
void com.example.Field.incrementFlags (
            GameModes mode)
```

Ez a metódus növeli a mezőhöz tartozó zászlók számát. Abban az esetben ha eléri a zászlók maximum lehetséges számát, visszatéríti a mezőt az eredeti állapotba.

Parameters

mode	A játékmód amelyben történik a zászló növelés, ez azért szükséges mert függ a játékmódtól a zászlók maximum száma.

3.4.3.8 maxFlag()

```
int com.example.Field.maxFlag (
            GameModes mode)
```

A függvény a játékmódhoz tartozó maximum zászló számmal tér vissza.

Parameters

mode	A játékmód

Returns

1, ha a mode default; 3, ha a mód rat.

3.4.3.9 resetFlags()

```
void com.example.Field.resetFlags ()
```

A függvény a mezőket visszatéríti eredeti állapotába, a zászlók száma 0 lesz, és a flagged változó hamis lesz.

3.4.3.10 setFlagged()

```
void com.example.Field.setFlagged (
            boolean flagged)
```

A függvény a mező zászlóként beállítottságát változatja meg.

Parameters

flagged	A mező zászlóként van-e bejelölve.

3.4.3.11 setMineNumber()

```
void com.example.Field.setMineNumber (
            int number)
```

A függvény beállítja a mezőhöz tartozó bombák számát Az isMineField változó értéke igaz lesz.

Parameters

number-	Bombák száma.

3.4.3.12 setNumberOfNeighbors()

```
void com.example.Field.setNumberOfNeighbors (
            int number)
```

Beállítja a mező numberofNeighbors változóját.

Parameters

number	A mezővel szomszédos, bombát tartalmazó mezők száma.

3.4.3.13 setRevealed()

```
void com.example.Field.setRevealed (
            boolean revealed)
```

A függvény a mező felfedettségét állítja be.

Parameters

revealed-	A mező fel van-e fedve.

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

- demo/src/main/java/com/example/Field.java

3.5 com.example.FieldTest Class Reference

A **Field** (p. 11) osztály metódusainak tesztelésért felelős osztály.

Public Member Functions

- void **testFieldConstructor** ()
- void **testSetMineNumber** ()
- void **testSetNumberOfNeighbors** ()
- void **testSetRevealed** ()
- void **testSetFlagged** ()
- void **testIncrementFlagsDefaultMode** ()

3.5.1 Detailed Description

A **Field** (p. 11) osztály metódusainak tesztelésért felelős osztály.

Ez az osztály a **Field** (p. 11) osztály metódusainak helyes működését vizsgálja, beleértve a zászlókezelést, a szomszédos bombák számát és a mezők állapotát meghatározó logikát.

3.5.2 Member Function Documentation

3.5.2.1 testFieldConstructor()

```
void com.example.FieldTest.testFieldConstructor ()
```

Teszteli a **Field** (p. 11) konstruktorát alapértelmezett értékekkel. Ellenőrzi, hogy az aknamentes mező megfelelő kezdőállapotot kap.

3.5.2.2 testIncrementFlagsDefaultMode()

```
void com.example.FieldTest.testIncrementFlagsDefaultMode ()
```

Teszteli a zászlózás logikáját alapértelmezett játékmódban: 1 -> 0 ciklust

3.5.2.3 testSetFlagged()

```
void com.example.FieldTest.testSetFlagged ()
```

Teszteli, hogy a zászlóként kijelölés állapota helyesen frissül.

3.5.2.4 testSetMineNumber()

```
void com.example.FieldTest.testSetMineNumber ()
```

Teszteli a **Field** (p. 11) konstruktorát alapértelmezett értékekkel. Ellenőrzi, hogy az aknamentes mező megfelelő kezdőállapotot kap.

3.5.2.5 testSetNumberOfNeighbors()

```
void com.example.FieldTest.testSetNumberOfNeighbors ()
```

Teszteli, hogy a setNumberOfNeighbors helyesen állítja be a szomszédos mezők számát.

3.5.2.6 testSetRevealed()

```
void com.example.FieldTest.testSetRevealed ()
```

Teszteli a mező felfedettségének beállítását.

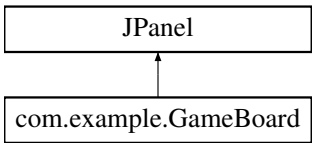
The documentation for this class was generated from the following file:

- demo/src/test/java/com/example/FieldTest.java

3.6 com.example.GameBoard Class Reference

A **GameBoard** (p. 16) osztály felelős a játéktábla vizuális megjelenítéséért.

Inheritance diagram for com.example.GameBoard:



Public Member Functions

- **GameBoard** ()
- void **setGameController** (**GameController** controller)
- **Position** **getCellFromCoordinates** (int x, int y)
- void **rePaintCell** (**Position** pos)
- void **drawBorders** (Graphics g)
- void **initializeBoard** (Graphics g)
- void **paintComponent** (Graphics g)
- void **drawNumber** (Graphics g, int number, int boardPositionX, int boardPositionY)
- void **drawNumberCell** (Graphics g, int neighborNumber, int boardPositionX, int boardPositionY)
- void **drawFlags** (Graphics g, int flagNumber, int boardPositionX, int boardPositionY, Color background)
- void **drawBomb** (Graphics g, int boardPositionX, int boardPositionY, Color background, int mineNumber)
- void **drawCheese** (Graphics g, int x, int y)
- void **drawMouse** (Graphics g, int x, int y)
- void **drawCell** (int row, int column, Graphics g, **Field** field)
- **GameController** **getController** ()

3.6.1 Detailed Description

A **GameBoard** (p. 16) osztály felelős a játéktábla vizuális megjelenítéséért.

Feladata a cellák, szomszédos bombák száma, zászlók, bombák, egér és sajt kirajzolása a panelen. Kezeli a repaint-et a változások szerint, és a felhasználó kattintásaira reagál, a **GameController** (p. 22) logikáját használva.

Kapcsolódó osztályok:

- **GameMouseAdapter** (p. 31): Az egérkattintásra való reagálás
- **GameController** (p. 22): a játék logikájának kezelése
- **Field** (p. 11): a játéktábla mezői
- **Position** (p. 40): a mezők pozícióinak kezelése
- **RatTable** (p. 45), **Rat** (p. 42): RAT játékmódhoz tartozó egér és célpozíció
- **GameModes** (p. 31), **Difficulties** (p. 10): játékmód és nehézségi szint információk

3.6.2 Constructor & Destructor Documentation

3.6.2.1 GameBoard()

```
com.example.GameBoard.GameBoard ()
```

Gameboard osztály konstruktora Hozzáadja az egérkattintások kezelését.

3.6.3 Member Function Documentation

3.6.3.1 drawBomb()

```
void com.example.GameBoard.drawBomb (
            Graphics g,
            int boardPositionX,
            int boardPositionY,
            Color background,
            int mineNumber)
```

Ez a metódus kirajzol egy bomba mezőt. Erre akkor van szükség, ha egy bomba helytelenül biztonságos mezőnek lett bejelölve. Minden esetben egy bomba van a cellába berajzolva. A bombák belsejében szerepel, hogy a cella hány bombát tartalmaz. A cella háttérszíne a paraméterként megkapott szín lesz.

Parameters

g	a grafikus objektum, amire rajzolunk
boardPositionX	a cella X-koordinátája a panelen
boardPositionY	a cella Y-koordinátája a panelen
background	a cella hátterének színe

mineNumber	a cellában szereplő bombák száma

3.6.3.2 drawBorders()

```
void com.example.GameBoard.drawBorders (
            Graphics g)
```

Ez a metódus berajzolja a panelen a cellák széleit fekete színúre.

Parameters

g	a grafikus objektum, amire rajzolunk

3.6.3.3 drawCell()

```
void com.example.GameBoard.drawCell (
            int row,
            int column,
            Graphics g,
             Field field)
```

Ez a metódus választja ki, hogy mi igaz a megadott mezőre és ez alapján, mit kell a panelre felrajzolni. Ha a játékos vesztett, a biztonságos mezőnek nézett bomba és a helytelen számú zászlóval bejelölt mező piros színű hátteret kapnak.

Parameters

row	a kijelölt mező sora
column	a kijelölt mező oszlopa
g	a grafikus objeltum amire rajzolunk
field	a kijelölt mező

3.6.3.4 drawCheese()

```
void com.example.GameBoard.drawCheese (
            Graphics g,
            int x,
            int y)
```

Ez a metódus sajtot rajzol arra a pozícióra, ami a RAT objektumnak a célpozíciója. A sajt kirajzolásához 2D-s grafika van alkalmazva.

Parameters

g	a grafikus objektum, amire rajzolunk

x	a cella X-koordinátája a panelen
y	a cella Y-koordinátája a panelen

3.6.3.5 drawFlags()

```
void com.example.GameBoard.drawFlags (
            Graphics g,
            int flagNumber,
            int boardPositionX,
            int boardPositionY,
            Color background)
```

Ez a metódus kirajzolja mezőkhöz tartozó zászlókat. Ha több zászló is van egy mezőben, a zászlók színe különbözik. A zászlóhoz tartozik egy fekete tartórúd is. A cella háttérszíne a paraméterként megkapott szín lesz.

Parameters

g	a grafikus objektum, amire rajzolunk
flagNumber	a zászlók száma
boardPositionX	a cella X-koordinátája a panelen
boardPositionY	a cella Y-koordinátája a panelen
background	a cella hátterének színe

3.6.3.6 drawMouse()

```
void com.example.GameBoard.drawMouse (
            Graphics g,
            int x,
            int y)
```

Ez a metódus az egeret rajzolja ki, a RAT objektum jelenlegi pozíciójára Az egér feje, fülei, szemei és orra van a panelre felrajzolva.

Parameters

g	a grafikus objektum, amire rajzolunk
x	a cella X-koordinátája a panelen
y	a cella Y-koordinátája a panelen

3.6.3.7 drawNumber()

```
void com.example.GameBoard.drawNumber (
            Graphics g,
            int number,
            int boardPositionX,
            int boardPositionY)
```

Ez a metódus felrajzolja a mező szomszédos bombáinak számát a panelre.

Parameters

g	a grafikus objektum, amire rajzolunk
number	a szomszédos bombák száma
boardPositionX	a cella X-koordinátája a panelen
boardPositionY	a cella Y-koordinátája a panelen

3.6.3.8 drawNumberCell()

```
void com.example.GameBoard.drawNumberCell (
            Graphics g,
            int neighborNumber,
            int boardPositionX,
            int boardPositionY)
```

Ez a metódus abban az esetben, ha a szomszédos bombák száma legalább 1, kirajzolja a panelre a számot. A szám színe a getNumberColor metódus által visszaadott lesz.

Parameters

g	a grafikus objektum, amire rajzolunk
neighborNumber	a szomszédos bombák száma
boardPositionX	a cella X-koordinátája a panelen
boardPositionY	a cella Y-koordinátája a panelen

3.6.3.9 getCellFromCoordinates()

```
Position com.example.GameBoard.getCellFromCoordinates (
            int x,
            int y)
```

Ez a metódus visszatér azzal a pozícióval, ami a panelen belüli koordinátákhoz tartozik.

Parameters

x	az X-koordináta panelen belül
y	az Y-koordináta panelen belül

Returns

a koordinátákhoz tartozó pozíció

3.6.3.10 getController()

```
GameController com.example.GameBoard.getController ()
```

Visszatéríti a játékpanelhez tartozó játékvezérlőt.

Returns

játékpanelhez tartozó játékvezérlő

3.6.3.11 initializeBoard()

```
void com.example.GameBoard.initializeBoard (
            Graphics g)
```

Ez a metódus berajzol minden mezőt a panelre a kezdeti állapot alapján.

Parameters

g	a grafikus objektum, amire rajzolunk

3.6.3.12 paintComponent()

```
void com.example.GameBoard.paintComponent (
            Graphics g)
```

Ez a metódus felel a komponensek kirajzolásáért. Elsőnek meghívja a JPanel paintComponent metódusát. Ezután ha panel nem volt még inicializálva minden mezőt berajzol. Ha a panel már inicializálva van csak azokat cellákat rajzolja meg. amelyek a repaint által érintett területbe beleesnek.

Parameters

g	a grafikus objektum, amire rajzolunk

3.6.3.13 rePaintCell()

```
void com.example.GameBoard.rePaintCell (
              Position pos)
```

Egy pozícióhoz tartozó cellát újrafest.

Parameters

pos	mező pozíciója a táblán

3.6.3.14 setGameController()

```
void com.example.GameBoard.setGameController (
              GameController controller)
```

Beállítja a játéktábla panelhez tartozó játékvezérlőt.

Parameters

controller	játékvezérlő

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

- demo/src/main/java/com/example/GameBoard.java

3.7 com.example.GameController Class Reference

A **GameController** (p. 22) osztály kezeli a játék logikáját és a játéktábla állapotát.

Public Member Functions

- **GameController** (**GameModes** mode, **Difficulties** dif, **GameBoard** board, **GameWindow** window)
- void **setGameTable** ()
- void **revealNeighborsOfEmptyFields** (**Field** f, Set< **Field** > visited)
- void **moveByOne** (List< **Position** > path)
- void **numberChoser** (**Field** selectedField)
- void **defaultMineChoser** (**Field** selectedField)
- void **ratMineChoser** (**Field** selectedField)
- void **mineChoser** (**Field** selectedField)
- void **generateTable** (**Position** selectedPosition)
- void **repaintMissingBombsAndWrongFlags** ()
- boolean **isEveryFieldSelected** ()
- void **setWinner** ()
- void **mouseChoser** (**MouseClicks** click, **Field** slectedField)
- boolean **getWinner** ()
- boolean **getTableGenerated** ()
- boolean **getBombActivated** ()
- **Table** **getTable** ()
- int **getUnflaggedOneMines** ()
- int **getUnflaggedTwoMines** ()
- int **getUnflaggedThreeMines** ()
- **GameModes** **getGameMode** ()

3.7.1 Detailed Description

A **GameController** (p. 22) osztály kezeli a játék logikáját és a játéktábla állapotát.

Az osztály felelős a játék indításáért, a tábla létrehozásáért, az egér mozgásáért (RAT mód), a mezők felfedéséért és a bombák kezeléséért. Kezeli a győzelem és veszteség logikáját, a bejelölt és be nem jelölt bombákat, valamint frissíti a játékablak felületét a változásoknak megfelelően.

Fő funkciók:

- Játéktábla létrehozása a nehézségi szintnek megfelelő paraméterekkel
- Üres mezők automatikus felfedése
- Egér lépése RAT módban
- Bejelölt bombák és helyes/hibás zászlók kezelése
- Győzelem és veszteség logika, ranglista frissítés
- Interakciók a felhasználóval (bal/jobb egérgomb)

Kapcsolódó osztályok és típusok:

- **GameBoard** (p. 16), **GameWindow** (p. 32): a felület és a vizuális megjelenítés kezelése
- **Table** (p. 53), **DefaultTable** (p. 5), **RatTable** (p. 45): a játék logikai táblája
- **Field** (p. 11): egy mező a táblán
- **Rat** (p. 42), **Position** (p. 40): egér pozíciójának kezelése RAT módban
- **GameModes** (p. 31), **Difficulties** (p. 10): a játékmód és nehézségi szint beállítása

3.7.2 Constructor & Destructor Documentation

3.7.2.1 GameController()

```
com.example.GameController.GameController (
              GameModes mode,
              Difficulties dif,
              GameBoard board,
              GameWindow window)
```

A **GameController** (p. 22) osztály konstruktora, beállítja a megfelelő tagváltozókat a paraméterként kapott értékekre.

Parameters

mode	a játékmód
dif	a játék nehézségi szintje
board	a játékhoz tartozó játéktábla panel
window	az ablak amelyen megjelenik a játék

3.7.3 Member Function Documentation

3.7.3.1 defaultMineChoser()

```
void com.example.GameController.defaultMineChoser (
              Field selectedField)
```

Ez a metódus csökkenti a bejelöletlen mezők számát, ha a mező bejelölt. Ha ez a mező bomba is, akkor a megmaradt bombák száma is csökken. Ha a mező nem bejelölt, a bejelöletlen mezők száma nő, mivel előtte bejelölt volt. Ha ez a mező bomba is, akkor a megmaradt bombák száma is nő. A bejelöletlen bombák száma a játék ablakában is megjelenik.

Parameters

selectedField	a kiválasztott mező

3.7.3.2 generateTable()

```
void com.example.GameController.generateTable (
              Position selectedPosition)
```

A metódus kiválasztja a táblában a bombák helyét. Ezután a megszámolja, hogy hány bomba szomszédos a többi mezővel. Ha a játékmód RAT, az egér kezdő és célpozícióját a kiválasztott pozícióra állítja.

Parameters

selectedPosition	a kiválasztott pozíció

3.7.3.3 getBombActivated()

```
boolean com.example.GameController.getBombActivated ()
```

Ez a metódus visszatér az aktivált bomba logikai értékkel.

Returns

aktivált bomba

3.7.3.4 getGameMode()

```
GameModes com.example.GameController.getGameMode ()
```

Ez a metódus visszatér a játékhoz tartozó játékmóddal.

Returns

játékmód

3.7.3.5 getTable()

```
Table com.example.GameController.getTable ()
```

Ez a metódus visszatér a játékhoz tartozó táblával.

Returns

játékhoz tartozó tábla

3.7.3.6 getTableGenerated()

```
boolean com.example.GameController.getTableGenerated ()
```

Ez a metódus visszatér a létrehozott tábla logikai értékkel.

Returns

létrehozott tábla logikai értéke.(true, ha létre van hozva)

3.7.3.7 getUnflaggedOneMines()

```
int com.example.GameController.getUnflaggedOneMines ()
```

Ez a metódus visszatér a be nem jelölt egyetlen bombát tartalmazó mezők számával

Returns

jelöletlen egyetlen bombát tartalmazó mezők száma

3.7.3.8 getUnflaggedThreeMines()

```
int com.example.GameController.getUnflaggedThreeMines ()
```

Ez a metódus visszatér a be nem jelölt három bombát tartalmazó mezők számával

Returns

három egyetlen bombát tartalmazó mezők száma

3.7.3.9 getUnflaggedTwoMines()

```
int com.example.GameController.getUnflaggedTwoMines ()
```

Ez a metódus visszatér a be nem jelölt két bombát tartalmazó mezők számával

Returns

jelöletlen két bombát tartalmazó mezők száma

3.7.3.10 getWinner()

```
boolean com.example.GameController.getWinner ()
```

Ez a metódus visszatér a győzelem logikai értékkel.

Returns

győzelem logikai értéke

3.7.3.11 isEveryFieldSelected()

```
boolean com.example.GameController.isEveryFieldSelected ()
```

Ez a metódus ellenőrzi, hogy nem maradt olyan mező, amely nincs se felfedve, se bejelölve.

Returns

true, ha nincs ki nem választott mező false, ha találtunk olyan mezőt, ami még nincs kiválasztva

3.7.3.12 mineChoser()

```
void com.example.GameController.mineChoser (
            Field selectedField)
```

Ez a metódus a mouse jobb gombnyomása utáni műveleteket végzi el. Ha a kiválasztott mező nem volt felfedve, növeli a mezőhöz tartozó zászlók számát. Az incrementflags metódus értelmében ha zászlók száma nagyobb már mint a maximum, a zászlók száma nulla lesz . Ezután ha a játékmód DEFAULT a defaultMineChoser, ha RAT, akkor a ratMineChoser metódust hívja meg a kiválasztott mezőre. Ha a kiválasztott mező már fel van fedve, és a játékmód **Rat** (p. 42), a kiválasztott mező lesz az egér célpozíciója. A legrövidebb útnak az egér jelenlegi és célpozíció közötti legrövidebb út lesz beállítva. A mezőn történt változások a játéktábla paneljára is felkerülnek.

Parameters

selectedField	a kiválasztott mező

3.7.3.13 mouseChoser()

```
void com.example.GameController.mouseChoser (
              MouseClicks click,
              Field slectedField)
```

Ez a metódus elvégzi a mouse által kapott gomb műveletét, ha a játék még nem ért véget. Bal gomb esetén, ha a tábla még nincs létrehozva előbb a táblát létrehozza. Minden gombnyomásra ellenőrzi, hogy vége-e a játéknak.

Parameters

click	gombnyomás típusa(jobb vagy bal)
slectedField	kibálasztott mező

3.7.3.14 moveByOne()

```
void com.example.GameController.moveByOne (
             List< Position > path)
```

A metódus által a táblán az egér lép egyet a paraméterként megadott úton. Ha olyan mezőre lép, amely nem tartalmaz bombát és nem volt még felfedve, se bejelölve azt felfedi. Ha üres mezőre lép az üres mezőnek szomszédait is felfedi Ha olyan mezőre lép amely tartalmaz bombát, de nem volt még bejelölve, a mezőn található bombák számával bejelöli. A bejelöletlen mezők száma is csökken ha ilyen mezőre lép az egér. A pozíció, amelyre lépett az egér ki lesz törölve az út listából.

Parameters

path	az út pozícióinak listája, amelyen kell lépkedjen az egér

3.7.3.15 numberChoser()

```
void com.example.GameController.numberChoser (
              Field selectedField)
```

Ez a metódus a mouse bal gombnyomása utáni műveleteket végzi el. Ha a mező nem volt még felfedve, se bejelölve, felfedi a mező tartalmát. Ha a mező nem bomba, felfedi a mezőhöz tartozó számot, ha ez nulla a szomszédjait is. RAT játékmód esetén az egér előre megy egyet a célhoz vezető úton. Ha a kiválasztott mező bomba, feljegyzi, hogy a bomba már aktiválva volt. A mezőn történt változások a játéktábla paneljára is felkerülnek.

Parameters

selectedField	a kiválasztott mező

3.7.3.16 ratMineChoser()

```
void com.example.GameController.ratMineChoser (
            Field selectedField)
```

Ez a metódus csökkenti a mező bejelölt zászlóinak megfelelő számlálót. Az eggyel kisebb számú bombát tartalmazó mezők számát növeli. Ha a mezőhöz egy bomba van bejelölve nem növel csak csökkent. Ha nincs bejelölve, csak növel, nem csökkent. Ha a helyes számú bomba van jelölve, a megmaradt bombamezők száma csökken. Ha a helyes számú bomba túl van lépve, a megmaradt bombamezők száma növekszik. A bejelöletlen bombák száma a játék ablakában is megjelenik.

Parameters

selectedField	a kiválasztott mező

3.7.3.17 repaintMissingBombsAndWrongFlags()

```
void com.example.GameController.repaintMissingBombsAndWrongFlags ()
```

Ez a metódus berajzolja azokat a bombákat amelyek nem voltak kijelölve. Emellett azon mezőknek kinézetét is megváltoztatja amelyek helytelenül voltak bejelölve.

3.7.3.18 revealNeighborsOfEmptyFields()

```
void com.example.GameController.revealNeighborsOfEmptyFields (
            Field f,
           Set< Field > visited)
```

Ez a metódus felfedi az üres mezővel szomszédos mezőket. Abban az esetben, ha a szomszédos mező már be van jelölve zászlóként, a mező nem lesz felfedve. Ha a szomszédok közül is van egy mező amely üres, annak szomszédait is felfedi. A visited set-ben tárolja a megtalált üres mezőket. Ez azért szükséges, hogy ugyanarra a mezőre ne legyen többször elvégezve a metódus, végtelen ciklust generálva.

Parameters

f	kiválasztott üres mező
visited	azon üres mezők, amelyekre a függvény már meg volt hívva.

3.7.3.19 setGameTable()

```
void com.example.GameController.setGameTable ()
```

Ez a metódus elindit egy új játékot . A játéktábla megkapja a játék nehézségéhez tartozó sor, oszlop és bombaszámot. Minden játék elején az aktivált bomba, nyertes és legenerált tábla logikai változók hamis értékűek lesznek. Default játékmód esetén a játék kezdetén a bejelöletlen bomba mezők száma egyenlő lesz az összes bomba számával. **Rat** (p. 42) játékmód esetén külön jelenik meg a bejelöletlen mezők amelyek egy, kettő és három bombát tartalmaznak. Mindegyik azt kapja meg értékként, hogy eredetileg külön-külön mennyi van belőlük a táblában.

3.7.3.20 setWinner()

```
void com.example.GameController.setWinner ()
```

A metódus leállítja az idő számlálását a játék ablakában. Abban az esetben ha a bomba már aktiválva van, a metódus a nyerést hamisra állítja. A ki nem rajzolt bombákat és rosszul bejelölt mezőket újrarajzolja. Emellett kiír egy üzenetet az ablakra. Ha a játékos nyert, a nyerés igaz lesz. A játékos idejét feljegyzi és bekerül a játékos neve a ranglistába, ha benne van az idő a top 10 leggyorsabban.

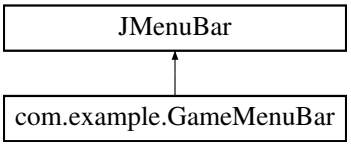
The documentation for this class was generated from the following file:

- demo/src/main/java/com/example/GameController.java

3.8 com.example.GameMenuBar Class Reference

A **GameMenuBar** (p. 29) kezeli a játék ablak menüsávját.

Inheritance diagram for com.example.GameMenuBar:



Public Member Functions

- JMenu **getDifficultyMenu** ()
- JMenu **getGameModeMenu** ()
- boolean **modeOrDifficultyIsNotSelected** ()
- JMenuItem **getNewGameMenuItem** ()
- **GameMenuBar** (**GameWindow** window)

3.8.1 Detailed Description

A **GameMenuBar** (p. 29) kezeli a játék ablak menüsávját.

Feladata, hogy a felhasználó kiválaszthassa a nehézségi szintet, játékmódot, valamint új játékot indíthasson.

Kapcsolódó osztályok:

- **GameWindow** (p. 32): az ablak, amelyhez a menüsáv tartozik
- **Difficulties** (p. 10): az elérhető nehézségi szinteket tartalmazó enum
- **GameModes** (p. 31): az elérhető játékmódokat tartalmazó enum
- JMenuBar, JMenu, JMenuItem: a Swing komponensek, amiket a menüsáv használ

3.8.2 Constructor & Destructor Documentation

3.8.2.1 GameMenuBar()

```
com.example.GameMenuBar.GameMenuBar (
            GameWindow window)
```

Létrehozza a **GameMenuBar** (p. 29) objektumot és hozzáadja a menüpontokat.

Parameters

window	az ablak amelyen szerepel a menü

3.8.3 Member Function Documentation

3.8.3.1 getDifficultyMenu()

```
JMenu com.example.GameMenuBar.getDifficultyMenu ()
```

Ez a metódus egy új JMenu-t generál, amely neve "Difficulties" Ehhez a könnyű, közepes és nehéz nehézségi szintek vannak hozzáadva. Minden menüponthoz hozzárendeli a megfelelő egérkattintás kezelést, amely beállítja a kiválasztott nehézséget.

Returns

a létrehozott JMenu objektum

3.8.3.2 getGameModeMenu()

```
JMenu com.example.GameMenuBar.getGameModeMenu ()
```

Ez a metódus egy új JMenu-t generál, amely neve "Gamemode" Ehhez a default és rat játékmód van hozzáadva. Mindkettő menüponthoz hozzárendeli a megfelelő egérkattintás kezelést, amely beállítja a kiválasztott játékmódot.

Returns

a létrehozott JMenu objektum

3.8.3.3 getNewGameMenuItem()

```
JMenuItem com.example.GameMenuBar.getNewGameMenuItem ()
```

Ez a metódus egy új JMenuItem-et generál, amely neve "Gamemode". Ha a nehézségi szint és játékmód is ki van választva, a menüpontra kattintva elindul egy új játék.

Returns

a létrehozott JMenuItem objektum

3.8.3.4 modeOrDifficultyIsNotSelected()

```
boolean com.example.GameMenuBar.modeOrDifficultyIsNotSelected ()
```

Ha a nehézségi szint, vagy a játékmód nincs kiválasztva megjelenik egy hibaüzenet. A metódus ebben az esetben true értékkel tér vissza. Ha mindkettő ki van választva false értékkel tér vissza.

Returns

true, ha valamelyik nincs kiválasztva false, ha mindkettő ki van választva

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

- demo/src/main/java/com/example/GameMenuBar.java

3.9 com.example.GameModes Enum Reference

Az elérhető játékmódok felsorolása.

Public Attributes

- **DEFAULT**
- **RAT**

3.9.1 Detailed Description

Az elérhető játékmódok felsorolása.

- DEFAULT: normál játék.
- RAT: egér-sajtos játékmód, speciális szabályokkal.

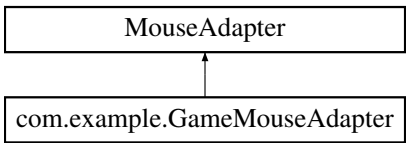
The documentation for this enum was generated from the following file:

- demo/src/main/java/com/example/GameModes.java

3.10 com.example.GameMouseAdapter Class Reference

A **GameMouseAdapter** (p. 31) kezeli az egér kattintásokat a játéktáblán.

Inheritance diagram for com.example.GameMouseAdapter:



Public Member Functions

- **GameMouseAdapter** (**GameBoard** board)
- void **mousePressed** (MouseEvent e)

3.10.1 Detailed Description

A **GameMouseAdapter** (p. 31) kezeli az egér kattintásokat a játéktáblán.

Feladata, hogy a **GameBoard** (p. 16) panelen történt egér eseményeket a **GameController** (p. 22) megfelelő metódusaihoz továbbítsa.

Kapcsolódó osztályok:

- **GameBoard** (p. 16): a panel, amin az adapter figyeli az egér eseményeket
- **MouseClicks** (p. 40): az egérgombok típusát reprezentáló enum
- **Position** (p. 40): a kattintás helyének táblabeli koordinátái

3.10.2 Constructor & Destructor Documentation

3.10.2.1 GameMouseAdapter()

```
com.example.GameMouseAdapter.GameMouseAdapter (
            GameBoard board)
```

A **GameMouseAdapter** (p. 31) osztály konstruktora, megkapja a hozzá tartozó játékpanelt.

Parameters

board	a játékpanel

3.10.3 Member Function Documentation

3.10.3.1 mousePressed()

```
void com.example.GameMouseAdapter.mousePressed (
            MouseEvent e)
```

Felülírja a gombnyomás műveletét, úgy, hogy a játékvezérlő mouseChoser metódusát hívja meg, azzal a gombbal ami meg lett nyomva

Parameters

e	az egérrel történt esemény

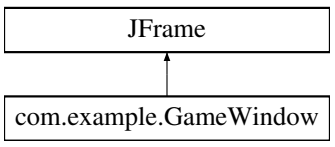
The documentation for this class was generated from the following file:

- demo/src/main/java/com/example/GameMouseAdapter.java

3.11 com.example.GameWindow Class Reference

A játék főablaka, amely tartalmazza a menüt, a játékpanelt, az információs panelt és a ranglistát.

Inheritance diagram for com.example.GameWindow:



Public Member Functions

- void **setInfoPanel** (**GameModes** mode)
- void **refreshHighscoreArea** ()
- void **saveHighscoreAfterWin** (int timePassed, File outputFile)
- void **writeMessageAfterLosing** ()
- **GameWindow** ()
- void **stopTimer** ()
- void **updateMines** (int remainingMines, int flagnumber)
- void **startNewGame** (**GameModes** mode, **Difficulties** dif)
- **HighscoreManager** **getHighscoreManager** ()
- int **getTimePassed** ()

3.11.1 Detailed Description

A játék főablaka, amely tartalmazza a menüt, a játékpanelt, az információs panelt és a ranglistát.

Feladata:

- A **GameBoard** (p. 16) és **GameController** (p. 22) összekapcsolása.
- Az eltelt idő és a be nem jelölt bombák számának megjelenítése.
- A **HighscoreManager** (p. 36) kezelése, játékos eredmények mentése és megjelenítése.
- Új játék indítása a kiválasztott nehézségi szint és játékmód alapján.

Kapcsolódó osztályok:

- **GameBoard** (p. 16): a játékmezők kirajzolása.
- **GameController** (p. 22): a játék logikájának vezérlése.
- **GameMenuBar** (p. 29): a játék menüje, innen lehet indítani új játékot.
- **HighscoreManager** (p. 36): a legjobb eredmények kezelése.

3.11.2 Constructor & Destructor Documentation

3.11.2.1 GameWindow()

```
com.example.GameWindow.GameWindow ()
```

A **GameWindow** (p. 32) osztály konstruktora. Hozzáadja az ablakhoz a gameMenuBar, infopanel, sidePanel és gameBoard objektumokat. Az ablak nagy részét a játékpanel teszi ki. Az infopanel a felső részre kerül, a sidePanel az ablak jobb oldalára. Kilépést az X-et megnyomva lehet végrehajtani.

3.11.3 Member Function Documentation

3.11.3.1 getHighscoreManager()

```
HighscoreManager com.example.GameWindow.getHighscoreManager ()
```

Ez a metódus visszatéríti az ablakhoz tartozó HighScoreManager objektumot.

Returns

jelenlegi játékhoz tartozó HighScoreManager objektum

3.11.3.2 getTimePassed()

```
int com.example.GameWindow.getTimePassed ()
```

Ez a metódus visszatéríti az eltelt időt másodpercben

Returns

az eltelt idő másodpercben

3.11.3.3 refreshHighscoreArea()

```
void com.example.GameWindow.refreshHighscoreArea ()
```

Ez a metódus frissíti a ranglista megjelenítését. Általa az új játékos eredménye is bekerül.

3.11.3.4 saveHighscoreAfterWin()

```
void com.example.GameWindow.saveHighscoreAfterWin (
            int timePassed,
            File outputFile)
```

Ez a metódus elmenti a nyertes eredmlnyt, ha az benne van a legjobb 10-ben. Ha nincs név beírva akkor a név helyére Anonymus kerül. Meghívja a legjobb eredmények hozzáadási metódusát. Ez akkor adja hozzá, ha a legjobb 10 idő között van. Ha hozzáadás után szerepel az új eredmény a legjobb eredmények listájában a legjobb eredmények frissülnek az ablakban és elmenti az eredményeket egy bináris fájlban. Más eredmény jelenik meg, ha bekerül a játékos a top 10-be, mint ha nem kerülne be.

Parameters

timePassed	az eltelt idő másodpercben
outputFile	a kimeneti bináris fájl

3.11.3.5 setInfoPanel()

```
void com.example.GameWindow.setInfoPanel (
            GameModes mode)
```

Ez a metódus beállítja az infoPanelt, ezen szerepel az eltelt idő és a bejelöletlen bombák száma. RAT játékmód esetén az is felkerül, hogy hány két és három bombát tartalmazó bejelöletlen mező van.

Parameters

mode	a kiválasztott játékmód

3.11.3.6 startNewGame()

```
void com.example.GameWindow.startNewGame (
            GameModes mode,
            Difficulties dif)
```

Ez a metódus elindít egy új játékot Felrajzolja a játékpanel kezdeti állapotát Elkezdi az idő számolását. Beolvassa az eddigi eredményeket ebben a játékmódban és nehézségben

Parameters

mode	a játékmód
dif	a játék nehézsége

3.11.3.7 stopTimer()

```
void com.example.GameWindow.stopTimer ()
```

Ez a metódus megállítja az eltelt idő számítását.

3.11.3.8 updateMines()

```
void com.example.GameWindow.updateMines (
            int remainingMines,
            int flagnumber)
```

Frissiti a be nem jelölt bombát tartalmazó mezők számát az ablakban.

Parameters

remainingMines	be nem jelölt bombák száma
flagnumber	bomba típusa(1-normál, 2-dupla, 3-tripla)

3.11.3.9 writeMessageAfterLosing()

```
void com.example.GameWindow.writeMessageAfterLosing ()
```

Ez a metódus vesztes játék után ír ki üzenetet.

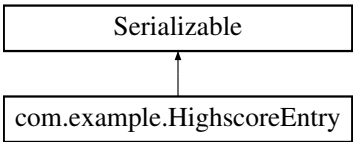
The documentation for this class was generated from the following file:

- demo/src/main/java/com/example/GameWindow.java

3.12 com.example.HighscoreEntry Class Reference

Egy játékos egyetlen eredményét tárolja a ranglistában.

Inheritance diagram for com.example.HighscoreEntry:



Public Member Functions

- **HighscoreEntry** (int time, String name)
- int **getTime** ()
- String **getName** ()

3.12.1 Detailed Description

Egy játékos egyetlen eredményét tárolja a ranglistában.

Tartalmazza:

- a játékos nevét,
- az elért időt másodpercben.

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

- demo/src/main/java/com/example/HighscoreEntry.java

3.13 com.example.HighscoreManager Class Reference

A játék ranglistáinak kezelése.

Public Member Functions

- **HighscoreManager** (int maxEntry)
- File **getTextFileByModeAndDifficulty** (**GameModes** mode, **Difficulties** difficulty)
- void **loadHighScores** (File readableHighscores)
- void **saveHighscore** (File writeableHighscores)
- void **addScore** (**HighscoreEntry** entry)
- ArrayList< **HighscoreEntry** > **getHighscoreEntries** ()

3.13.1 Detailed Description

A játék ranglistáinak kezelése.

Feladata:

- a játékosok eredményeinek tárolása,
- eredmények betöltése és mentése bináris fájlokba,
- legjobb eredmények listájának karbantartása.

Kapcsolódó osztályok:

- **HighscoreEntry** (p. 36): az egyes eredményeket tárolja.
- **GameWindow** (p. 32): az eredmények megjelenítése az ablakban.

3.13.2 Constructor & Destructor Documentation

3.13.2.1 HighscoreManager()

```
com.example.HighscoreManager.HighscoreManager (
            int maxEntry)
```

A **HighscoreManager** (p. 36) osztály konstruktora. Létrehoz egy listát és beállítja a lista maximális tárolható eredményszámot a paraméterként megkapottal

Parameters

maxEntry	maximum eredmény szám, amennyi benne lehet a listában.

3.13.3 Member Function Documentation

3.13.3.1 addScore()

```
void com.example.HighscoreManager.addScore (
             HighscoreEntry entry)
```

Ez a metódus hozzáadja a paraméterként megkapott eredményt, ha annak ideje a megengedett számú legjobb idő között szerepel. Ha már volt előtte a listában maximális számú eredmény és az új eredmény is bekerült az utolsó a listából törlődik. A listában az eredmények idő szerinti növekvő sorrendben szerepelnek.

Parameters

entry	az új eredmény

3.13.3.2 getHighscoreEntries()

```
ArrayList< HighscoreEntry > com.example.HighscoreManager.getHighscoreEntries ()
```

Ez a metódus visszatéríti az legjobb eredmények listáját

Returns

legjobb eredmények listája

3.13.3.3 getTextFileByModeAndDifficulty()

```
File com.example.HighscoreManager.getTextFileByModeAndDifficulty (
            GameModes mode,
            Difficulties difficulty)
```

Ez a metódus meghatározza a játékhoz tartozó bináris fájlt. Ezt a játékmód és a nehézség alapján teszi meg. Ezt a fájlt téríti vissza eredményként.

Parameters

mode	játékmód
difficulty	játék nehézsége

Returns

a metódus által meghatározott fájl

3.13.3.4 loadHighScores()

```
void com.example.HighscoreManager.loadHighScores (
            File readableHighscores)
```

Ez a metódus kiolvassa a bináris fájlban szereplő eredményeket. Ha a fájl nem létezik visszatér, ha pedig hiba történik az olvasás közben, a hibát elkapja.

Parameters

readableHighscores	az erdményeket tartalmazó bináris fájl

3.13.3.5 saveHighscore()

```
void com.example.HighscoreManager.saveHighscore (
            File writeableHighscores)
```

Ez a metódus kiíratja egy bináris fájlba a játék legjobb eredményeit. Ha hiba történik fájlba írás közben, a hibát elkapja.

Parameters

writeableHighscores	a fájl, amelybebe az eredményeket írjuk

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

- demo/src/main/java/com/example/HighscoreManager.java

3.14 com.example.Main Class Reference

A program belépési pontja.

Static Public Member Functions

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

3.14.1 Detailed Description

A program belépési pontja.

Feladata:

- elindítani a játékot egy új **GameWindow** (p. 32) ablak létrehozásával.

Kapcsolódó osztályok:

- **GameWindow** (p. 32): a játék grafikus felülete és fő logikája.

3.14.2 Member Function Documentation

3.14.2.1 main()

```
void com.example.Main.main (
            String[] args)  [static]
```

A **Main** (p. 39) osztály main metódusa egy új játékablakot indít el.

Parameters

args	

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

- demo/src/main/java/com/example/Main.java

3.15 com.example.MouseClicks Enum Reference

Az egérkattintások típusait reprezentáló felsorolás.

Public Attributes

- **LEFT**
- **RIGHT**

3.15.1 Detailed Description

Az egérkattintások típusait reprezentáló felsorolás.

- LEFT: bal egérgomb kattintás.
- RIGHT: jobb egérgomb kattintás.

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

- demo/src/main/java/com/example/MouseClicks.java

3.16 com.example.Position Class Reference

A **Position** (p. 40) osztály a játéktábla egy mezőjének koordinátáit reprezentálja.

Public Member Functions

- **Position** (int row, int column)
- int **getRow** ()
- int **getColumn** ()
- boolean **equals** (Object obj)
- int **hashCode** ()

3.16.1 Detailed Description

A **Position** (p. 40) osztály a játéktábla egy mezőjének koordinátáit reprezentálja.

Tárolja a mező sor- és oszlopszámát, és biztosítja ezek lekérését getter metódusokon keresztül. Felülírja az equals és hashCode metódusokat, hogy a pozíciók objektumként is összehasonlíthatók legyenek, illetve használhatók legyenek HashMap kulcsként.

3.16.2 Constructor & Destructor Documentation

3.16.2.1 Position()

```
com.example.Position.Position (
            int row,
            int column)
```

A pozíció osztály konstruktora. A pozíció sora és oszlopa kerül általa beállításra.

Parameters

row	a poz@ció sora
column	a pozíció oszlopa

3.16.3 Member Function Documentation

3.16.3.1 equals()

```
boolean com.example.Position.equals (
            Object obj)
```

Felülírja az equals metódust.

Parameters

obj	a hasonlított objektum

Returns

true, ha midnkét objektum a **Position** (p. 40) osztály egy példánya és a sorok és oszlopok megegyeznek;false más esetben

3.16.3.2 getColumn()

```
int com.example.Position.getColumn ()
```

Returns

a pozíció oszlopát téríti vissza

3.16.3.3 getRow()

```
int com.example.Position.getRow ()
```

Returns

a pozíció sorát téríti vissza

3.16.3.4 hashCode()

```
int com.example.Position.hashCode ()
```

Felülírja a hashCode metódust.

Returns

a pozícióhoz tartozü hash érték

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

- demo/src/main/java/com/example/Position.java

3.17 com.example.PositionTest Class Reference

A **Position** (p. 40) osztály működését ellenőrző JUnit tesztosztály.

Public Member Functions

- void **testPositionConstructorInitialValues** ()
- void **testEquals** ()
- void **testHashCode** ()

3.17.1 Detailed Description

A **Position** (p. 40) osztály működését ellenőrző JUnit tesztosztály.

Teszteli a **Position** (p. 40) konstruktorát, az equals metódust és a hashCode metódust. Ellenőrzi, hogy a sor- és oszlopszámok helyesen kerülnek beállításra, azonos koordinátájú pozíciók egyenlőek legyenek, és a hashCode értékek megfeleljenek az equals metódus logikájának.

3.17.2 Member Function Documentation

3.17.2.1 testEquals()

```
void com.example.PositionTest.testEquals ()
```

Teszteli a **Position** (p. 40) equals metódusát. Azonos koordinátákkal rendelkező pozíciók legyenek egyenlőek, eltérők pedig ne.

3.17.2.2 testHashCode()

```
void com.example.PositionTest.testHashCode ()
```

Teszteli, hogy azonos pozíciók azonos hashCode értéket kapnak, míg különböző pozíciók különbözőt.

3.17.2.3 testPositionConstructorInitialValues()

```
void com.example.PositionTest.testPositionConstructorInitialValues ()
```

Teszteli, hogy a **Position** (p. 40) konstruktor a megadott sor- és oszlopszámot helyesen állítja be.

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

- demo/src/test/java/com/example/PositionTest.java

3.18 com.example.Rat Class Reference

A **Rat** (p. 42) osztály az egér pozícióját kezeli a játéktáblán.

Public Member Functions

- **Rat** (**Position** currentPosition)
- void **setGoal** (**Position** goalPosition)
- void **setCurrentPosition** (**Position** currentPosition)
- **Position** **getCurrentPosition** ()
- **Position** **getGoalPosition** ()

3.18.1 Detailed Description

A **Rat** (p. 42) osztály az egér pozícióját kezeli a játéktáblán.

A **Rat** (p. 42) objetum a **Rat** (p. 42) játékmód esetén fogs szerepelni a játék tábláján. Az osztály tárolja az egér jelenlegi pozícióját és a célpozícióját. A konstruktorral beállított kezdőpozíció lesz a kezdeti célpont is. Tartalmaz getter és setter metódusokat mindkét pozíció kezelésére.

3.18.2 Constructor & Destructor Documentation

3.18.2.1 Rat()

```
com.example.Rat.Rat (
            Position currentPosition)
```

A **Rat** (p. 42) osztály konstruktora. Beállítja a jelenlegi és célpozíciót is a paraméterként megkapottra

Parameters

currentPosition	jelenlegi pozíció, a játék kezdetén a célpozíció is ezt az értéket kapja meg

3.18.3 Member Function Documentation

3.18.3.1 getCurrentPosition()

```
Position com.example.Rat.getCurrentPosition ()
```

isszatéríti a jelenlegi pozícióval.

Returns

jelenlegi pozíció

3.18.3.2 getGoalPosition()

```
Position com.example.Rat.getGoalPosition ()
```

Visszatéríti a célpozícióval.

Returns

célpozíció

3.18.3.3 setCurrentPosition()

```
void com.example.Rat.setCurrentPosition (
            Position currentPosition)
```

Beállítja a jelenlegi pozíció értékét.

Parameters

currentPosition-	jelenlegi pozíció értéke

3.18.3.4 setGoal()

```
void com.example.Rat.setGoal (
            Position goalPosition)
```

Beállítja a célpozíció értékét.

Parameters

goalPosition-	célpozíció értéke

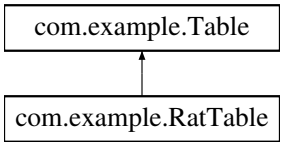
The documentation for this class was generated from the following file:

- demo/src/main/java/com/example/Rat.java

3.19 com.example.RatTable Class Reference

A wrap-around logikával működő aknakereső tábla, amelyet a **Rat** (p. 42) játékmód használ.

Inheritance diagram for com.example.RatTable:



Public Member Functions

- **RatTable** (int rownumber, int columnnumber, int mineNumber)
- void **setRat** (**Position** firstPosition)
- List< **Position** > **getNeighborPositions** (**Position** p)
- void **checkNeighbors** ()
- void **selectingMines** (ArrayList< **Field** > availableFields)
- List< **Position** > **reconstructPath** (Map< **Position**, **Position** > parent)
- List< **Position** > **getRightDirectionNeighbors** (**Position** current)
- List< **Position** > **findShortestPath** ()
- void **setShortestPath** ()
- List< **Position** > **getShortestPath** ()
- **Rat** **getRat** ()
- int **getOneMineFields** ()
- int **getTwoMineFields** ()
- int **getThreeMineFields** ()

Public Member Functions inherited from com.example.Table

- **Table** (int rowNumber, int columnNumber, int mineNumber)
- **Field getFieldByPosition** (**Position** p)
- **Position getPositionByField** (**Field** f)
- List< **Position** > **getNeighbors** (**Position** currentPosition)
- ArrayList< **Field** > **getavailableFields** (**Position** selectedPosition)
- int **getAllMines** ()
- int **getRows** ()
- int **getColumns** ()

Additional Inherited Members

Protected Attributes inherited from com.example.Table

- int **rows**
- int **columns**
- List< **Field** > **fields** = new ArrayList< **Field**>()
- int **allMines**

3.19.1 Detailed Description

A wrap-around logikával működő aknakereső tábla, amelyet a **Rat** (p. 42) játékmód használ.

A **RatTable** (p. 45) a **Table** (p. 53) absztrakt osztály speciális implementációja, ahol a tábla szélei összeérnek (wrap-around), így minden mezőnek nyolc szomszédja van akkor is, ha a tábla szélén található. A mezőkben lehet akár több darab bomba is(1, 2 vagy 3 lehet), és ezek arányai az összes bombamező számától függően vannak kiosztva.

Az osztály tartalmaz egy **Rat** (p. 42) objektumot, amelynek célpozíciójához a tábla képes meghatározni a legrövidebb utat BFS algoritmus segítségével.
A legrövidebb út meghatározása figyelembe veszi a cél felé "helyes irányban" található szomszédos pozíciókat is.

3.19.2 Constructor & Destructor Documentation

3.19.2.1 RatTable()

```
com.example.RatTable.RatTable (
            int rownumber,
            int columnumber,
            int mineNumber)
```

A **RatTable** (p. 45) meghívja az ősosztálya, a **Table** (p. 53) konstruktorát. Az egyetlen bombát tartalmazó mezők száma az összes bomba felével lesz egyenlő. A két bombát tartalmazó mezők száma az összes bomba/4-el egyenlő. Ugyanaz igaz a 3 bombát tartalmazó mezőkre, mint a 2 bombásokra.

Parameters

rows	sorok száma
columns	oszlopok száma

allMines	bombák száma

3.19.3 Member Function Documentation

3.19.3.1 checkNeighbors()

```
void com.example.RatTable.checkNeighbors ()
```

Ellenőrzi, hogy egy mező szomszédjaiban összesen hány bomba található. Azon mezők vannak ellenőrizve, amelyek nem tartalmaznak bombák. Ha egy mező szomszédja tartalmaz bombát növeli a mezővel szomszédos bombák számlálóját annyival amennyi bombát tartalmaz a szomszédos mező. A mező szomszédos bombáinak száma a számláló értéke lesz. A szomszédok között ebben a számításban is beletartoznak a szélső pozíciókhoz tartozó másik oldalon lévő pozíciók.

Reimplemented from **com.example.Table** (p. 55).

3.19.3.2 findShortestPath()

```
List< Position > com.example.RatTable.findShortestPath ()
```

A legrövidebb út megtalálása BFS algoritmust használva. Az egér jelenlegi pozíciójához képest keresi meg a célpozícióhoz a legrövidebb utat.

A metódus működése:

- A BFS egy queue-t használ a bejárási sorrend biztosításához.
- A visited halmaz megakadályozza, hogy ugyanazt a pozíciót többször vizsgálja (így elkerülhetők a ciklusok).
- A parent map minden csúcs esetén eltárolja, hogy melyik pozícióból értük el — ez teszi lehetővé az út rekonstrukcióját a cél elérésekor.

Returns

A célpozícióhoz vezető út legrövidebb út pozíciónak listája vagy üres lista, ha nem létezik elérhető út

3.19.3.3 getNeighborPositions()

```
List< Position > com.example.RatTable.getNeighborPositions (
            Position p)
```

Ez a metódus visszatér egy mező szomszédjainak listájával. A getNeighborsWrapAround metódust hívja meg a szomszédok listájának eléréséhez.

Parameters

p	jelenlegi pozíció, melynek szeretnénk megtudni a szomszédjait

Returns

szomszédpozíciók listája

Reimplemented from **com.example.Table** (p. 56).

3.19.3.4 getOneMineFields()

```
int com.example.RatTable.getOneMineFields ()
```

A metódus visszatér az egyetlen bombát tartalmazó mezők számával

Returns

az egyetlen bombát tartalmazó mezők száma

Reimplemented from **com.example.Table** (p. 58).

3.19.3.5 getRat()

```
Rat com.example.RatTable.getRat ()
```

Visszatéríti a táblához tartozó **Rat** (p. 42) objektumot.

Returns

táblához tartozó **Rat** (p. 42) objektum

3.19.3.6 getRightDirectionNeighbors()

```
List< Position > com.example.RatTable.getRightDirectionNeighbors (
            Position current)
```

Ez a függvény visszaadja egy pozíció olyan szomszédjait amelyek nincsenek egyik irányból távolabb az egér célpozíciójától. Ez azért fontos, hogy a legrövidebb útba ne kerüljenek olyan pozíciók, amelyek az azelőtti pozícióhoz képest távolabb vannak a céltól valamelyik irányban.

Parameters

current	a jelenlegi pozíció

Returns

szomszédos pozíciók listája, amelyek nincsenek távolabb a céltól egyik irányból se

3.19.3.7 getShortestPath()

```
List< Position > com.example.RatTable.getShortestPath ()
```

Visszatéríti a legrövidebb utat

Returns

legrövidebb út pozícióinak listája

3.19.3.8 getThreeMineFields()

```
int com.example.RatTable.getThreeMineFields ()
```

A metódus visszatér a három bombát tartalmazó mezők számával

Returns

a három bombát tartalmazó mezők száma

3.19.3.9 getTwoMineFields()

```
int com.example.RatTable.getTwoMineFields ()
```

A metódus visszatér a két bombát tartalmazó mezők számával

Returns

a két bombát tartalmazó mezők száma

3.19.3.10 reconstructPath()

```
List< Position > com.example.RatTable.reconstructPath (
            Map< Position, Position > parent)
```

Viszaépíti a findShortestPath által megkeresett legrövidebb utat. A parent HashMapban minden pozíció kulcsa az előtte lévőknek. így ezt végigjárva hozzáadhatjuk a pozíciókat az úthoz. A végén meg kell fordítani az utat, hogy a megfelelő sorrendben legyenek benne a pozíciók.

Parameters

parent	a legrövidebb út pozícióinak HashMapje

Returns

a legrövidebb út pozícióinak listája

3.19.3.11 selectingMines()

```
void com.example.RatTable.selectingMines (
            ArrayList< Field > availableFields)
```

Kiválasztja azokat a mezőket, amelyek bombát tartalmaznak. A bombát tartalmazó mezők között nem lehet olyan, ami az elsőként kiválasztott pozíció szomszédja. Az elérhető mezők össze vannak keverve annak érdekében , hogy a táblán véletlenszerűen szerepljenek a bombák. A táblához tartozó bombák számával egyenlő mezőnek felének beállítja a hozzá tartozó bombák számát 1-re, a negyedét 2-re és az utolsó negyedét 3-ra.

Parameters

availableFields	az elérhető pozíciók

Reimplemented from **com.example.Table** (p. 58).

3.19.3.12 setRat()

```
void com.example.RatTable.setRat (
            Position firstPosition)
```

A tábla **Rat** (p. 42) objektuma megkapja egy új, létrehozott **Rat** (p. 42) értékét. A **Rat** (p. 42) konstruktorában a kezdőpozíciót kapja meg.

Parameters

firstPosition	a **Rat** (p. 42) kezdőpozíciója

3.19.3.13 setShortestPath()

```
void com.example.RatTable.setShortestPath ()
```

A shortestPath tagváltozó megkapja a findShortestPath metódus által visszatérített listát.

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

- demo/src/main/java/com/example/RatTable.java

3.20 com.example.RatTableTest Class Reference

A **RatTable** (p. 45) osztály múködését ellenőrző JUnit tesztosztály.

Public Member Functions

- void **testRatTableConstructorInitialValues** ()
- void **testSetRat** ()
- void **testGetNeighborPositions** ()
- void **testCheckNeighbors** ()
- void **testSelectingMines** ()
- void **testGetRightDirectionNeighbors** ()
- void **testRatMovementSimple** ()

3.20.1 Detailed Description

A **RatTable** (p. 45) osztály múködését ellenőrző JUnit tesztosztály.

Teszteli többek között a konstruktor működését, a bombák helyes kiválasztását, a wrap-around működő szomszédszámlálást. Emellett teszteli a táblához tartozó egér helyes elhelyezését és a legegyszerűbb egérmozgás helyességét, beleértve a célhoz vezető útvonal helyes kiszámítását.

3.20.2 Member Function Documentation

3.20.2.1 testCheckNeighbors()

```
void com.example.RatTableTest.testCheckNeighbors ()
```

Teszteli, hogy a checkNeighbors helyesen számolja ki a szomszédos mezők aknaszámát a beállított bombák alapján.

3.20.2.2 testGetNeighborPositions()

```
void com.example.RatTableTest.testGetNeighborPositions ()
```

Teszteli, hogy a **RatTable** (p. 45) konstruktora helyesen állítja be a sorok, oszlopok, összes bánya és a különböző típusú aknamezők számát.

3.20.2.3 testGetRightDirectionNeighbors()

```
void com.example.RatTableTest.testGetRightDirectionNeighbors ()
```

Teszteli, hogy az egér cél irányába eső szomszédos pozíciók valóban közelebb kerülnek a célhoz sor- és oszlopirányban.

3.20.2.4 testRatMovementSimple()

```
void com.example.RatTableTest.testRatMovementSimple ()
```

Teszteli, hogy az egér cél irányába eső szomszédos pozíciók valóban közelebb kerülnek a célhoz sor- és oszlopirányban.

3.20.2.5 testRatTableConstructorInitialValues()

```
void com.example.RatTableTest.testRatTableConstructorInitialValues ()
```

Teszteli, hogy a **RatTable** (p. 45) konstruktora helyesen állítja be a sorok, oszlopok, összes bánya és a különböző típusú aknamezők számát.

3.20.2.6 testSelectingMines()

```
void com.example.RatTableTest.testSelectingMines ()
```

Teszteli, hogy a bombák kiválasztása megfelelően történik: a mezők aknaszámai és a mezők száma helyesen alakul.

3.20.2.7 testSetRat()

```
void com.example.RatTableTest.testSetRat ()
```

Teszteli, hogy a setRat metódus megfelelően beállítja az egér aktuális és célpozícióját.

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

- demo/src/test/java/com/example/RatTableTest.java

3.21 com.example.RatTest Class Reference

A **Rat** (p. 42) osztály működését ellenőrző JUnit tesztosztály.

Public Member Functions

- void **testRatConstructorInitialValues** ()
- void **testSetCurrentAndGoal** ()

3.21.1 Detailed Description

A **Rat** (p. 42) osztály működését ellenőrző JUnit tesztosztály.

Teszteli a **Rat** (p. 42) konstruktorát és a pozíciók kezelését.
Ellenőrzi, hogy a kezdőpozíció helyesen kerül beállításra, és hogy a currentPosition és goalPosition pozíciók helye megfelelően módosítható a setter metódusok segítségével.

3.21.2 Member Function Documentation

3.21.2.1 testRatConstructorInitialValues()

```
void com.example.RatTest.testRatConstructorInitialValues ()
```

Teszteli, hogy a **Rat** (p. 42) konstruktora helyesen állítja be a kezdő és a cél pozíciót azonos értékre.

3.21.2.2 testSetCurrentAndGoal()

```
void com.example.RatTest.testSetCurrentAndGoal ()
```

Teszteli, hogy a **Rat** (p. 42) objektum current és goal pozíciói helyesen módosíthatók a setter metódusokkal.

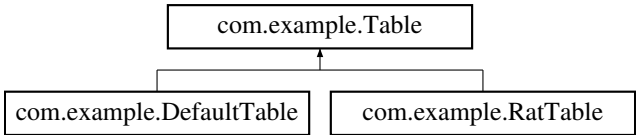
The documentation for this class was generated from the following file:

- demo/src/test/java/com/example/RatTest.java

3.22 com.example.Table Class Reference

Absztrakt táblaosztály az aknakereső játékhoz.

Inheritance diagram for com.example.Table:



Public Member Functions

- **Table** (int rowNumber, int columnNumber, int mineNumber)
- **Field getFieldByPosition** (**Position** p)
- **Position getPositionByField** (**Field** f)
- List< **Position** > **getNeighbors** (**Position** currentPosition)
- abstract List< **Position** > **getNeighborPositions** (**Position** p)
- abstract void **checkNeighbors** ()
- ArrayList< **Field** > **getavailableFields** (**Position** selectedPosition)
- abstract void **selectingMines** (ArrayList< **Field** > availableFields)
- int **getAllMines** ()
- abstract int **getOneMineFields** ()
- int **getRows** ()
- int **getColumns** ()

Protected Attributes

- int **rows**
- int **columns**
- List< **Field** > **fields** = new ArrayList< **Field**>()
- int **allMines**

3.22.1 Detailed Description

Absztrakt táblaosztály az aknakereső játékhoz.

A **Table** (p. 53) osztály reprezentál egy játékteret, amely sorokból és oszlopokból áll,
és mezőket (**Field** (p. 11)) tartalmaz. Emellett tartalmazza azt is, hogy hány bombamező szerepel a táblában. Az osztály felelős a mezők tárolásáért, szomszédsági viszonyok kezeléséért, valamint a bombák elhelyezésének logikájáért. A konkrét játékmódok (**DefaultTable** (p. 5), **RatTable** (p. 45)) ebből az osztályból származnak, és megvalósítják a játékmód-specifikus funkciókat, mint például a szomszédpozíciók meghatározása vagy a bombák kijelölése.

3.22.2 Constructor & Destructor Documentation

3.22.2.1 Table()

```
com.example.Table.Table (
            int rowNumber,
            int columnNumber,
            int mineNumber)
```

A tábla osztály konstruktora. A tagváltozók megkapják a paraméterben átadott érékeket. A tábla méretéhez igazodva minden pozícióhoz hozzáadódik egy új, bomba nélküli mező.

Parameters

rowNumber	sorok száma
columnNumber	oszlopok száma
mineNumber	bombák száma

3.22.3 Member Function Documentation

3.22.3.1 checkNeighbors()

```
abstract void com.example.Table.checkNeighbors ()  [abstract]
```

Ez a metódus a **DefaultTable** (p. 5) és **RatTable** (p. 45) osztályokban van megvalósítva. Ellenőrzi, hogy egy mező szomszédjaiban összsen hány bomba található.

Reimplemented in **com.example.DefaultTable** (p. 6), and **com.example.RatTable** (p. 47).

3.22.3.2 getAllMines()

```
int com.example.Table.getAllMines ()
```

Ez a metódus felfedi az üres mezővel szomszédos mezőket. Abban az esetben, ha a szomszédos mező már be van jelölve zászlóként, a mező nem lesz felfedve. Ha a szomszédok közül is van egy mező amely üres, annak szomszédait is felfedi A visited set-ben tárolja a megtalált üres mezőket. Ez azért szükséges, hogy ugyanarra a mezőre ne legyen többször elvégezve a metódus, végtelen ciklust generálva.

Parameters

f	kiválasztott üres mező
visited	azon üres mezők, amelyekre a függvény már meg volt hívva. Visszatér a bombát tartalmazó mezők számával

Returns

bombát tartalmazó mezők száma

3.22.3.3 getavailableFields()

```
ArrayList<  Field > com.example.Table.getavailableFields (
              Position selectedPosition)
```

Visszatér azokkal a mezőkkel amelyek nem szomszédosak a paraméterként kiválasztottal. Egy új availableFields lista megkapja a fields lista elemeit. Innen törlődnek a meagdott pozícióval szomszédos pozíciókhoz tartozó mezők. Valamint törlődik a megadott pozícióhoz tartozó mezők is

Parameters

selectedPosition	a kiválasztott mező

Returns

összes mező-(kiválasztott+szomszédok) listája

3.22.3.4 getColumns()

```
int com.example.Table.getColumns ()
```

A függvény visszatér a tábla oszlopainak számával

Returns

tábla oszlopainak száma

3.22.3.5 getFieldByPosition()

```
Field com.example.Table.getFieldByPosition (
            Position p)
```

Visszatér a mezővel a field listából a pozíció alapján

Parameters

p	a mező pozíciója

Returns

a fieldben található mező, amely megfelel a pozíciónak

3.22.3.6 getNeighborPositions()

```
abstract List< Position > com.example.Table.getNeighborPositions (
            Position p) [abstract]
```

Ez a metódus a **DefaultTable** (p. 5) és **RatTable** (p. 45) osztályokban van megvalósítva. Mindkét estetben a játékmód táblájának megfelelő pozíciókkal tér vissza.

Parameters

p	jelenlegi pozíció, melynek szeretnénk megtudni a szomszédjait

Returns

szomszédpozíciók listája

Reimplemented in **com.example.DefaultTable** (p. 7), and **com.example.RatTable** (p. 47).

3.22.3.7 getNeighbors()

```
List< Position > com.example.Table.getNeighbors (
            Position currentPosition)
```

Ez a metódus egy pozícióval szomszédos pozíciókkal tér vissza. Abban az esetben, ha a szomszédos pozíció szerepel a táblán (nem lépi túl a tábla méreteit) hozzáadódik a szomszéd pozíciók listához.

Parameters

currentPosition	jelenlegi pozició, melynek szeretnénk megtudni a szomszédjait

Returns

szomszédpozíciók listája

3.22.3.8 getOneMineFields()

```
abstract int com.example.Table.getOneMineFields ()  [abstract]
```

Ez a metódus a **DefaultTable** (p. 5) és **RatTable** (p. 45) osztályokban van megvalósítva. A függvény visszatér az egyetlen bombát tartalmazó mezők számával

Returns

az egyetlen bombát tartalmazó mezők száma

Reimplemented in **com.example.DefaultTable** (p. 8), and **com.example.RatTable** (p. 48).

3.22.3.9 getPositionByField()

```
Position com.example.Table.getPositionByField (
            Field f)
```

Visszatér a mező pozíciójával. A fields listában keresi meg a paraméterként megadott mezőt.

Parameters

f	a mező

Returns

a mezőhöz tartozó pozíció; amennyiben nem található meg a mező a fields listában, (-1,-1) pozícióval tér vissza.

3.22.3.10 getRows()

```
int com.example.Table.getRows ()
```

A függvény visszatér a tábla sorainak számával

Returns

tábla sorainak száma

3.22.3.11 selectingMines()

```
abstract void com.example.Table.selectingMines (
            ArrayList< Field > availableFields)  [abstract]
```

Ez a metódus a **DefaultTable** (p. 5) és **RatTable** (p. 45) osztályokban van megvalósítva. Kiválasztja azokat a mezőket, amelyek bombát tartalmaznak. A bombát tartalmazó mezők között nem lehet olyan, ami az elsőként kiválasztott pozíció szomszédja.

Parameters

availableFields	az elérhető pozíciók

Reimplemented in **com.example.DefaultTable** (p. 8), and **com.example.RatTable** (p. 49).

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

- demo/src/main/java/com/example/Table.java