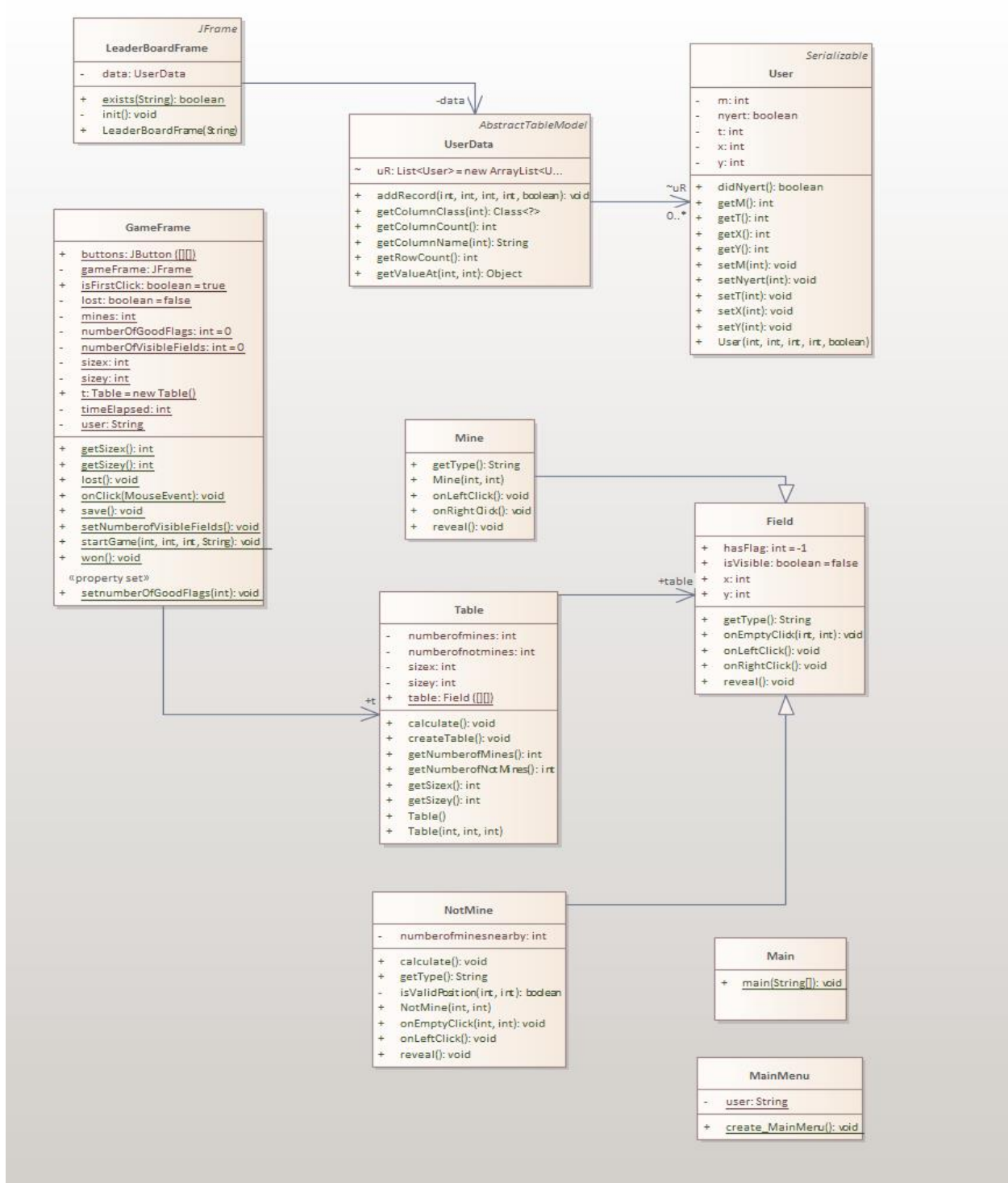


Aknakereső

1.Pont: Osztály diagram:



2.Pont: Osztályok leírása:

Field osztály

Az Field osztály egyetlen mezőt reprezentál az aknakereső játékban. Ez a mező lehet látható vagy rejtett, és rendelkezhet egy zászlóval. Az osztály felelős a mező tulajdonságainak kezeléséért, mint például a láthatóság, zászló jelenléte, és a mező típusának lekérdezése.

Adattagok

int x: A mező X koordinátája a játéktáblán.

int y: A mező Y koordinátája a játéktáblán.

int hasFlag: Egy zászló jelenlétét jelző érték. -1, ha nincs zászló, 1, ha van.

boolean isVisible: Jelzi, hogy a mező látható-e.

Metódusok

onRightClick()

A jobb kattintás esemény kezelése. Ha a mező látható, nem történik semmi. Ha nincs zászló a mezőn, hozzáad egy zászlót a képhez, egyébként eltávolítja azt. Az hasFlag értékét megváltoztatja.

onLeftClick()

A bal kattintás esemény kezelése. Ha a mezőn van zászló és látható, nem történik semmi. Ellenkező esetben a mező láthatóvá válik.

reveal()

A mező felfedését végző metódus. A leszármazott osztályok felüldefiniálják ezt a metódust a saját viselkedésüknek megfelelően.

onEmptyClick(int x, int y)

Az üres mezőre való kattintás eseményét kezelő metódus. A leszármazott osztályok felüldefiniálják ezt a metódust a saját viselkedésüknek megfelelően.

getType()

Visszaadja a mező típusát. Jelen esetben mindig "F", ami egy üres mezőt jelöl.

GameFrame osztály

Az GameFrame osztály felelős a játék ablakának kezeléséért, az események kezeléséért, és a játékállapot tárolásáért. Ennek az osztálynak az instanciái reprezentálják magát a játékot.

Adattagok

int sizeX: A játéktábla sorainak száma.

int sizeY: A játéktábla oszlopainak száma.

int mines: A játékban található aknák száma.

int timeElapsed: Az eltelt idő másodpercben.

int numberOfGoodFlags: A helyesen elhelyezett zászlók száma.

int numberOfVisibleFields: A látható mezők száma.

boolean lost: Jelzi, hogy a játékos veszített-e.

boolean isFirstClick: Jelzi, hogy az első kattintás történt-e.

JFrame gameFrame: A játék ablaka.

Table t: A játéktáblát reprezentáló objektum.

JBUTTON[][] buttons: A játéktáblát alkotó gombokat tároló mátrix.

String user: A játékos neve.

Metódusok

startGame(int sx, int sy, int ms, String u)

Inicializálja a játékot a megadott paraméterekkel, és megjeleníti a játék ablakát. Az ablakban szerepel egy időmérő a játék időtartamának követésére.

onClick(MouseEvent e)

A gombokra való kattintás eseményét kezeli. A bal egérgombbal felfedi a mezőt, a jobb egérgombbal helyezi vagy távolítja el a zászlót. A játék logikája ebben a metódusban valósul meg.

lost()

A játékos vesztesét kezeli. Felfedi az összes mezőt, majd megjelenít egy vesztes ablakot a visszatérési lehetőségekkel.

won()

A játékos győzelmét kezeli. Felfedi az összes mezőt, majd megjelenít egy győztes ablakot a visszatérési lehetőségekkel.

setNumberOfGoodFlags(int x)

Beállítja a helyesen elhelyezett zászlók számát.

setNumberOfVisibleFields()

Növeli a látható mezők számát.

getSizex()

Visszaadja a játéktábla sorainak számát.

getSizey()

Visszaadja a játéktábla oszlopainak számát.

save()

Ment egy felhasználót a ranglistára.

LeaderBoardFrame osztály

A LeaderBoardFrame osztály felelős a felhasználó rekordjait megjelenítő ablak kezeléséért.

Adattagok

UserData data: Az osztály tartalmaz egy UserData objektumot, amely a felhasználók rekordjait tárolja.

Metódusok

exists(String user): boolean

Egy statikus metódus, ami ellenőrzi, hogy egy adott felhasználó rekordjai már létezik-e.

init()

Inicializálja az ablakot, beállítva a layoutot és hozzáadva egy JTable-t a felhasználói adatok megjelenítéséhez. Ezenkívül beállít egy WindowAdapter-t, hogy az ablak bezárásakor mentse az adatokat.

LeaderBoardFrame(String userName)

A konstruktor inicializálja az ablakot a megadott felhasználó nevével. Betölti a felhasználó rekordjait és felkészíti az ablakot a megjelenítésre.

Main osztály

A Main osztály felelős a program futtatásáért és a kezdőmenü inicializálásáért.

Metódusok

main(String[] args)

A program belépési pontja. Ebben a metódusban hozzuk létre az alkalmazás első ablakát a kezdőmenüt, és ezt hozzuk létre egy eseményvezérlő szálon keresztül a SwingUtilities.invokeLater segítségével.

MainMenu osztály

A MainMenu osztály felelős a játék kezdőmenüjének létrehozásáért és kezeléséért.

Metódusok:

create_MainMenu()

Ez a metódus hozza létre a kezdőmenü ablakot és annak összetevőit.

startGameButton: A játék indításáért felelős gomb.

exitButton: Az alkalmazás kilépését lehetővé tevő gomb.

leaderBoardButton: A ranglista megtekintését lehetővé tevő gomb.

sizeXLabel, sizeYLabel, minesLabel: Címkék a pálya méreteinek beállításához.

sizeXField, sizeYField, minesField: Szövegmezők a pálya méreteinek beállításához.

userLabel, userField: Címkék és szövegmezők a felhasználónév beállításához.

A GridBagLayout segítségével rendezzük el ezeket az összetevőket a kezdőmenü ablakában.

A startGameButton gomb eseménykezelője ellenőrzi a bevitt adatokat, majd meghívja a GameFrame.startGame() metódust a megadott paraméterekkel, ha az adatok helyesek.

A leaderboardButton gomb eseménykezelője ellenőrzi, hogy a felhasználó neve már szerepel-e a ranglistán. Ha igen, megjeleníti a LeaderBoardFrame ablakot ezen névhez tartozó rekordokkal.

Adattagok:

user: Felhasználó nevét tároló adattag.

Mine osztály

A Mine osztály a leszármazottja a Field osztálynak, és reprezentálja az aknát a játékban.

Konstruktor:

Mine(int x, int y)

x: Az akna vízszintes pozíciója a táblán.

y: Az akna függőleges pozíciója a táblán.

Ez a konstruktor inicializálja az x és y adattagokat a megadott értékekkel.

Metódusok:

onLeftClick()

Ez az örökölt metódus felülírása. Akkor hívódik meg, amikor az aknára bal egérgombbal kattintanak. A GameFrame.lost() metódust hívja meg, ami kezeli a veszteséget.

onRightClick()

Ez az örökölt metódus felülírása. Akkor hívódik meg, amikor az aknára jobb egérgombbal kattintanak. Beállítja vagy eltávolítja a zászlót az akna mezőről, és frissíti a jó zászlók számát a GameFrame osztályban.

reveal()

Ez az örökölt metódus felülírása. Megjeleníti az akna ikonját a megfelelő gombon.

getType()

Ez az örökölt metódus felülírása. Visszaadja az akna típusát, ami "X".

Adattagok:

x: Az akna vízszintes pozíciója a táblán.

y: Az akna függőleges pozíciója a táblán.

NotMine osztály

A NotMine osztály a leszármazottja a Field osztálynak, és olyan mezőket reprezentál a játékban, amelyek nem tartalmaznak aknát.

Konstruktor:

NotMine(int x, int y)

x: A mező vízszintes pozíciója a táblán.

y: A mező függőleges pozíciója a táblán.

Ez a konstruktor inicializálja az x és y adattagokat a megadott értékekkel, valamint beállítja a numberOfminesnearby értékét nullára.

Metódusok:

onLeftClick()

Ez az örökölt metódus felülírása. Akkor hívódik meg, amikor a mezőre bal egérgombbal kattintanak. Meghívja a reveal metódust.

reveal()

Ez az örökölt metódus felülírása. Megjeleníti a mező értékét a megfelelő gombon, valamint beállítja a szöveg és háttérszínt az érték alapján. Ha a mező értéke 0, akkor meghívja az onEmptyClick metódust, ami rekurzívan felfedi a szomszédos üres mezőket.

calculate()

Ez a metódus meghatározza a mező szomszédos aknák számát.

```
onEmptyClick(int x, int y)
```

Ez a metódus rekurzívan felfedi az üres mezőket, és meghívja az onEmptyClick függvényt az üres szomszédos mezőkre.

```
isValidPosition(int x, int y)
```

Ez a segédfüggvény ellenőrzi, hogy a megadott pozíció a tábla határain belül van-e.

```
getType()
```

Ez az örökölt metódus felülírása. Visszaadja a mező szomszédos aknák számát szöveges formában.

Adattagok:

x: A mező vízszintes pozíciója a táblán.

y: A mező függőleges pozíciója a táblán.

numberOfminesnearby: A mező szomszédos aknák számát tárolja.

Table osztály

A Table osztály az aknakereső játék tábláját reprezentálja, ahol a mezők tartalmazhatnak aknákat vagy más értékeket.

Konstruktorok:

```
public Table()
```

Az alapértelmezett konstruktor, amely 10x10 méretű táblát hoz létre, ahol 10 akna található.

```
public Table(int sx, int sy, int nm)
```

Paraméterezhető konstruktor, amely egy adott méretű és aknaszámmal rendelkező táblát hoz létre.

Metódusok:

```
public void createTable()
```


Létrehozza a táblát, elhelyezi rajta az aknákat és a nem aknákat.

```
private void calculate()
```

Végigmegy a tábla mezőin, és ha a mező nem akna (NotMine), akkor kiszámolja, hány akna található a szomszédos mezőkön, és beállítja a megfelelő értéket a NotMine objektumon.

```
public int getSizex()
```

Visszaadja a tábla vízszintes méretét.

```
public int getSizey()
```

Visszaadja a tábla függőleges méretét.

```
public int getNumberofMines()
```

Visszaadja az aknák számát a táblán.

```
public int getNumberofNotMines()
```

Visszaadja a nem aknák számát a táblán.

Adattagok:

sizeX: A tábla vízszintes mérete.

sizeY: A tábla függőleges mérete.

numberofmines: Az aknák száma a táblán.

numberofnotmines: A nem aknák száma a táblán.

table: Egy kétdimenziós tömb, amely a tábla mezőit tárolja.

User osztály

A User osztály egy felhasználót reprezentál, aki játszott az aknakereső játékban. Ezen felhasználó rekordjait tárolja.

Metódusok:

```
public int getX()
```

Visszaadja az X méretét a játéktáblának.

```
public void setX(int a)
```

Beállítja az X méretét a játéktáblának.

```
public int getY()
```

Visszaadja az Y méretét a játéktáblának.

```
public void setY(int a)
```

Beállítja az Y méretét a játéktáblának.

```
public int getM()
```

Visszaadja az aknák számát.

```
public void setM(int a)
```

Beállítja az aknák számát.

```
public int getT()
```

Visszaadja az eltelt időt.

```
public void setT(int a)
```

Beállítja az eltelt időt.

```
public boolean didNyert()
```

Ellenőrzi, hogy a játékos nyert-e.

```
public void setNyert(int a)
```

Beállítja a nyert/vesztett állapotot a játékosnak.

```
public User(int x, int y, int m, int t, boolean n)
```

A User osztály konstruktora, ami inicializálja a felhasználó adatait.

Adattagok:

x: Az X méret a játéktáblának.

y: Az Y méret a játéktáblának.

m: Az aknák száma a játéktáblán.

t: Az eltelt idő.

nyert: Nyert/vesztett állapot.

UserData osztály

A UserData osztály egy táblázatot reprezentál, amely a játékosok rekordjait tárolja.

Metódusok:

```
public int getRowCount()
```

Visszaadja a táblázat sorainak számát, ami megegyezik a tárolt játékosok számával.

```
public int getColumnCount()
```

Visszaadja a táblázat oszlopainak számát. Itt mindig 5 oszlop található: X méret, Y méret, Aknák száma, Idő, Nyert/Vesztett.

```
public Object getValueAt(int rowIndex, int columnIndex)
```

Visszaadja a táblázat adott sorában és oszlopában található értéket. Az értékek a tárolt User objektumokból származnak.

```
public String getColumnName(int column)
```

Visszaadja a táblázat adott oszlopának nevét. Az oszlopok nevei: X, Y, Aknák, Idő, Nyert?

```
public Class<?> getColumnClass(int columnIndex)
```

Visszaadja az oszlopban található adatok típusát. Az első négy oszlopban integer típusú adatok találhatóak, az utolsó oszlopban pedig boolean típusú adatok (Nyert/Vesztett).

```
public void addRecord(int x, int y, int m, int t, boolean n)
```

Hozzáad egy új rekordot a táblázathoz a megadott értékekkel.

Adattagok:

uR: Egy lista, amely User objektumokat tárol, azaz a játékosok rekordjait.

3.Pont: Felhasználói dokumentáció:

A felhasználót egy főmenü fogja köszönteni a program futtatása után melyben található egy Felhasználói, X,Y és aknák száma mezőt, valamint három gombot.

Az egyik gomb a kilépésért felel.

Egy másik gomb meg megjeleníti az adott felhasználó rekordjait melyben rendezhetünk tetszőlegesen.

A harmadik gomb pedig a játék indítása melynek pályaméretét mi szabhatjuk meg.

A játék elkezdésekor elének tárul egy új ablak melyben megjelenik az általunk választott pálya valamint egy számláló ami az időt nézi.

A játékban a bal egérgombbal felfedhetünk mezőket, a jobbal pedig zászlót rakhatunk rá.

A játékot akkor nyerjük meg hogyha minden aknán mező van, és akkor bukjuk el ,hogya felfedtünk egy aknát. Amennyiben a játék véget ért. Egy kis ablak tárul elének melyben visszamehetünk a főmenübe vagy kiléphetünk a játékból.