

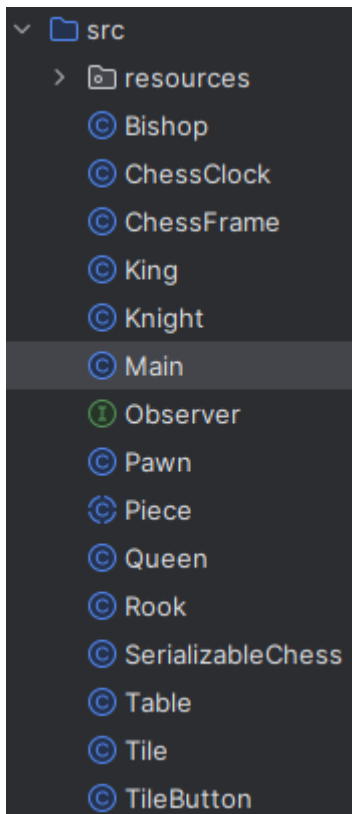
Házi feladat programozói dokumentáció – Sakkjáték

Móczár Dániel – CHYD8H

Fejlesztői környezet:

A programot a Java-ban a Swing library felhasználásával készítettem IntelliJ fejlesztői környezetben.

Projekt, forrásfájlok:



A projekt a következő forrásfájlokból épül fel:

- SerializableChess.java
- ChessFrame.java
- ChessClock.java
- Table.java
- Tile.java
- TileButton.java
- Piece.java
- Bishop.java
- King.java
- Knight.java
- Pawn.java
- Queen.java
- Rook.java
- Observer.java
- Main.java

A felsorolt forrásfájlok az alábbi interface és osztálydefiníciókat tartalmazzák:

Interface és osztálydefiníciók:

Az alább felsorolt osztályokat két külön kategóriába igyekeztem sorolni. Az egyik kategóriába tartozók a sakkjáték logikai felépítését és funkcióit tartalmazzák. Ezek az osztályok szerializálhatók. Az osztály tartalmának mentésekor a sakkjáték állása teljes egészében elmenthető, illetve betöltésekor visszaállítható.

Az osztályok másik csoportja a megjelenítésért felelős. Természetesen a megjelenítésért felelős osztályok és a logikáért felelős osztályok kapcsolódnak egymáshoz, de ezeket a kapcsolatokat minimalizálni igyekeztem.

A logikai felépítéshez tartozó osztályok:

- **SerializableChess:**

Tartalmazza a sakktáblát (Table), mely mezők (Tile) 8x8-as tömbjéből épül fel. Tartalmazza továbbá a sakkóra (ChessClock) objektumát is. A SerializableChess a sakkjáték működéséhez szükséges funkciókat is megvalósítja.
- **Table:**

A sakktáblát modellező osztály, mezők (Tile) 8x8-as tömbjéből épül fel. A mezők X és Y koordinátákkal címezhetők.
- **Tile:**

A sakkmezőt reprezentáló osztály. Ha sakkfigura helyezkedik el a mezőn, akkor tárolja annak objektumát.
- **Piece:**

A sakkfigurák absztrakt alaposztálya, melyből a sakkfigurák származnak. A sakkfigurák jellemzőin kívül tartalmazza a lépésekhez szükséges metódusokat.
- **King, Queen, Bishop, Knight, Rook, Pawn:**

A Piece alaposztályon alapuló, az adott sakkfigura típust megvalósító osztályok. Megvalósítja a Piece osztályban deklarált legalMoves() függvényt, mely visszaadja a sakkfigura típusának megfelelő lehetséges lépéseket.
- **ChessClock:**

A sakkórát megvalósító osztály időzítővel és a játékosok játékidejével. Az observer módszerrel tartja a kapcsolatot a grafikus megjelenítésért felelős objektummal.

A grafikus megjelenítéshez tartozó osztályok:

- ChessFrame:

A programablakot jeleníti meg, mely menüsávot, a sakktáblát, vezérlőgombokat, sakkórát és információs mezőket jelenít meg. A ChessFrame példánya a SerializableChess példányán keresztül végzi a sakkfunkciókat.

A ChessFrame tartalmazza a vezérlőgombokon, illetve a sakktábla mezejein végzett kattintások lekezelő osztályait.

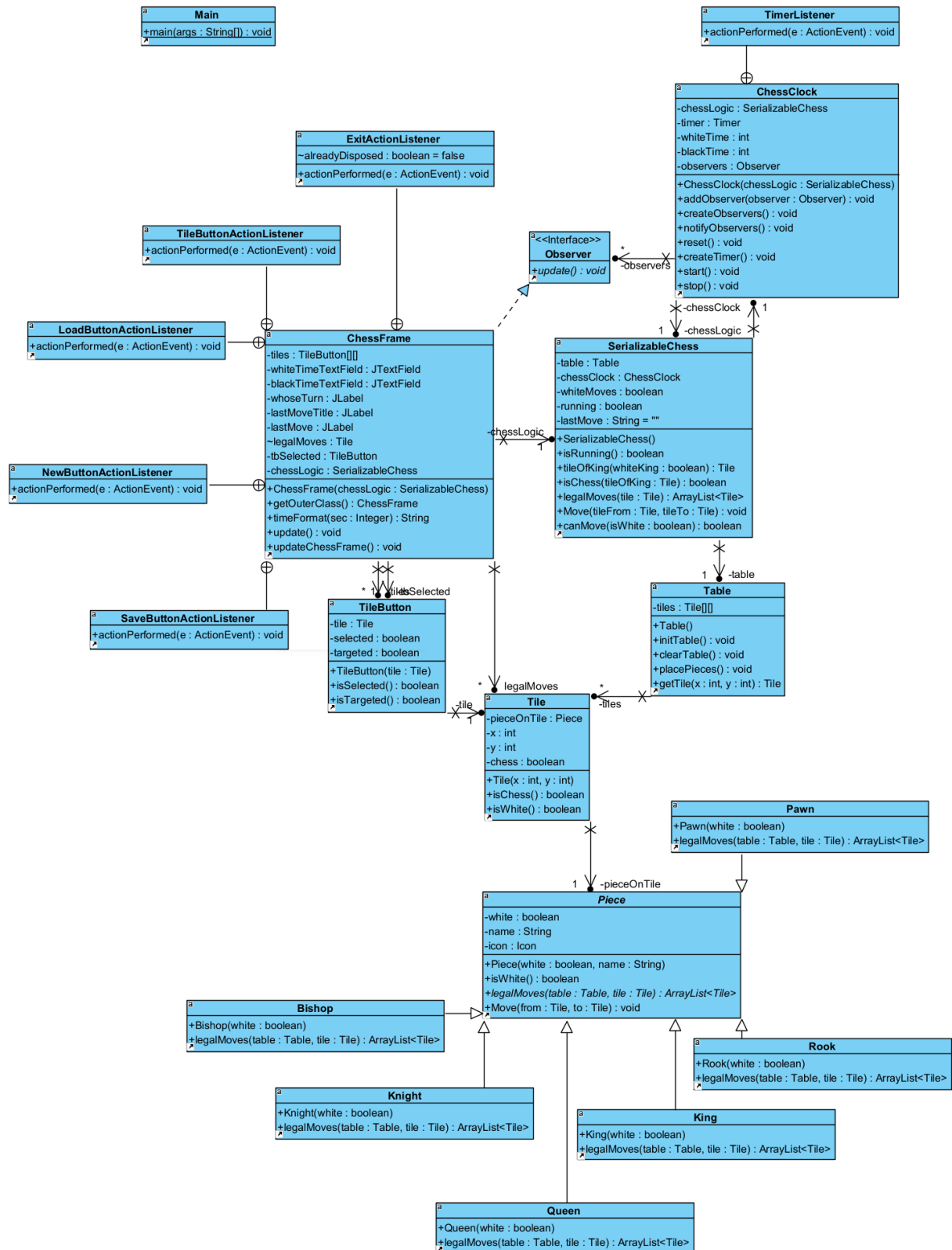
A ChessFrame az Observer interface-t implementálja, a ChessClock időzítője jelzésére hívódik az Observer update függvénye.

- TileButton:

A sakktábla mezejeit megjelenítő osztály. Kapcsolatot tartalmaz a sakkmező logikai funkcióját megvalósító Tile példánnyal.

A program belépési pontját a Main osztály tartalmazza, létrehozza a SerializableChess és ChessFrame példányait.

Osztálydiagram



Serializáció:

A játék mentése és visszatöltése serializációval történik. A SerializableChess és tagváltozói implementálják a Serializable interface-t.

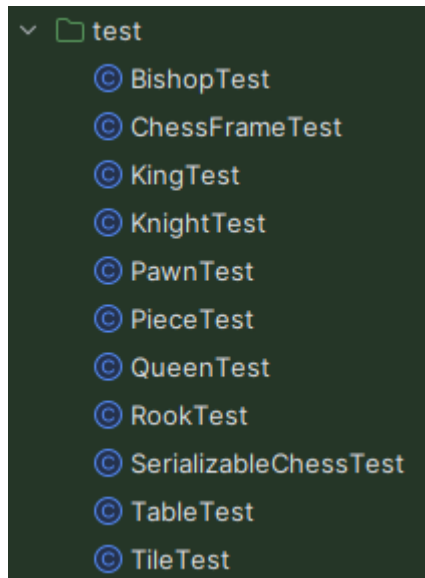
A játék visszatöltése funkcióban néhány tranziens tagváltozót újra létre kell hozni (ChessClock, timer és observers mezejei), valamint létre kell hozni a TileButton-Tile és ChessFrame-SerializableChess kapcsolatokat, ahogy az alábbi kódrészlet mutatja:

```
chessLogic = (SerializableChess) in.readObject();

//
for (int i = 0; i < 8; i++) {
    for (int j = 0; j < 8; j++) {
        //. . .
        tiles[i][j].setTile(chessLogic.getTable().getTile(i, j));
    }
}
//. . .
chessLogic.getChessClock().createObservers();
chessLogic.getChessClock().addObserver(getOuterClass());
chessLogic.getChessClock().createTimer();
if (chessLogic.isRunning())
    chessLogic.getChessClock().start();
```

Tesztelés:

A program tesztelését JUnit tesztekkel végeztem. A projekt külön könyvtárban tárolja a tesztesetekhez tartozó forrásfájlokat.



- BishopTest.java
A Bishop osztály legalMoves() függvényét teszteli.
- KingTest.java
A King osztály legalMoves() függvényét teszteli.
- KnightTest.java
A Knight osztály legalMoves() függvényét teszteli.
- PawnTest.java
A Pawn osztály legalMoves() függvényét teszteli.
- QueenTest.java
A Queen osztály legalMoves() függvényét teszteli.

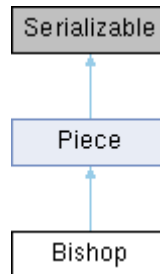
- RookTest.java
A Rook osztály legalMoves() függvényét teszteli.
- PieceTest.java
A Piece osztály Move() függvényét teszteli.
- ChessFrameTest.java
A ChessFrame osztály timeFormat() függvényét teszteli.
- TableTest.java
A Table osztály clearTable() függvényét teszteli.
- TileTest.java
A Tile osztály isWhite() függvényét teszteli.
- SerializableTest.java
A SerializableChess osztály tileOfKing(), isChess(), canMove() függvényeit teszteli.

Osztályok dokumentációja

Ezt a fejezetet a Doxygen generálta

Bishop osztályreferencia

A Bishop osztály származási diagramja:



Publikus tagfüggvények

- **Bishop** (boolean *white*)
- `ArrayList< Tile > legalMoves (Table table, Tile tile)`

Publikus tagfüggvények a(z) Piece osztályból származnak

- **Piece** (boolean *white*, String *name*)
- String **getName** ()
- Icon **getIcon** ()
- void **setIcon** (Icon icon)
- boolean **isWhite** ()
- abstract `ArrayList< Tile > legalMoves (Table table, Tile tile)`
- void **Move** (Tile from, Tile to)

Részletes leírás

A futóhoz tartozó osztály, a **Piece** osztályból származik.

Konstruktorok és destruktorok dokumentációja

Bishop.Bishop (boolean *white*)

Bishop konstruktor. Beállítja a bábu színét, nevét és ikonját.

Paraméterek

<i>white</i>	A bábu színe.
--------------	---------------

Tagfüggvények dokumentációja

`ArrayList< Tile > Bishop.legalMoves (Table table, Tile tile)`

Kiszámolja az adott mezőn elhelyezkedő bábu lehetséges lépéseit. Definiálja a **Piece** osztályban deklarált függvényt.

Paraméterek

<i>table</i>	Sakktábla.
--------------	------------

<i>tile</i>	Az adott mező ahol a bábu áll.
-------------	--------------------------------

Visszatérési érték

Lista, amely tartalmazza azokat a mezőket ahova a bábu léphet.

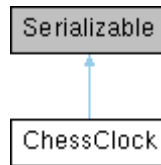
Újrimplementált ősök: **Piece** (o.22).

Ez a dokumentáció az osztályról a következő fájl alapján készült:

Bishop.java

ChessClock osztályreferencia

A ChessClock osztály származási diagramja:



Osztályok

- class **TimerListener**

Publikus tagfüggvények

- **ChessClock** (**SerializableChess** chessLogic)
- int **getWhiteTime** ()
- int **getBlackTime** ()
- void **addObserver** (**Observer** observer)
- void **createObservers** ()
- void **notifyObservers** ()
- void **reset** ()
- void **createTimer** ()
- void **start** ()
- void **stop** ()

Részletes leírás

A sakkórát leíró osztály. Implementálja a Szerializáláshoz szükséges Serializable interface-t

Konstruktorok és destruktorok dokumentációja

ChessClock.ChessClock (SerializableChess chessLogic)

ChessClock konstruktor.

- Létrehozza az observers listát.
- Létrehozza az időzítőt.
- Visszaállítja az időzítőt a 2 órás kezdeti értékre.
- Beállítja a sakklogikát.

Paraméterek

<i>chessLogic</i>	Az osztály metódusainak implementálásához szükséges sakklogika.
-------------------	---

Tagfüggvények dokumentációja

void ChessClock.addObserver (Observer observer)

Hozzáad egy figyelőt az observers listához.

Paraméterek

<i>observer</i>	A figyelő, amelyet a listához szeretnénk adni.
-----------------	--

void ChessClock.createObservers ()

Létrehozza az observers listát.

void ChessClock.createTimer ()

Létrehozza az időzítőt, mely másodpercenként jelez.

int ChessClock.getBlackTime ()

Getter, mely visszaadja a fekete játékos rendelkezésére álló játékidőt.

Visszatérési érték

A fekete játékos hátralevő ideje.

int ChessClock.getWhiteTime ()

Getter, mely visszaadja a fehér játékos rendelkezésére álló játékidőt.

Visszatérési érték

A fehér játékos hátralevő ideje.

void ChessClock.notifyObservers ()

Értesíti az observers listában lévő figyelőket. Meghívja azok update függvényeit.

void ChessClock.reset ()

Visszaállítja a játékosok rendelkezésére álló játékidőket.

void ChessClock.start ()

Elindítja az időzítőt.

void ChessClock.stop ()

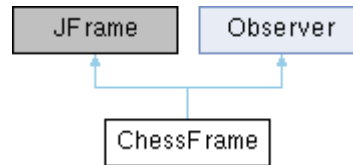
Megállítja az időzítőt.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

ChessClock.java

ChessFrame osztályreferencia

A ChessFrame osztály származási diagramja:



Osztályok

- class **ExitActionListener**
- class **LoadButtonActionListener**
- class **NewButtonActionListener**
- class **SaveButtonActionListener**
- class **TileButtonActionListener**

Publikus tagfüggvények

- **ChessFrame** (**SerializableChess** chessLogic)
- **ChessFrame** **getOuterClass** ()
- String **timeFormat** (Integer sec)
- void **update** ()
- void **updateChessFrame** ()
- void **update** ()

Részletes leírás

A sakkjáték grafikáját megvalósító osztály.

Konstruktorok és destruktorok dokumentációja

ChessFrame.ChessFrame (SerializableChess chessLogic)

ChessFrame konstruktor. Létrehozza a sakkjáték GUI elemeit, mint menüsáv, sakktábla, vezérlő gombok, sakkóra, stb.

Paraméterek

<i>chessLogic</i>	SerializableChess típusú objektum, mely a sakkjáték logikáját írja le.
-------------------	---

Tagfüggvények dokumentációja

ChessFrame ChessFrame.getOuterClass ()

Getter, a belső osztályok számára elérhetővé teszi a **ChessFrame** objektum referenciáját.

Visszatérési érték

A **ChessFrame** objektum.

String ChessFrame.timeFormat (Integer sec)

Visszaadja óó:pp:mm formátumban a rendelkezésre álló játékidőt.

Paraméterek

<i>sec</i>	Játékidő másodpercekben.
------------	--------------------------

Visszatérési érték

óó:pp:mm formátumú String.

void ChessFrame.update ()

Az **Observer** update függvényének implementációja.

A sakkóra jelzésekor hívódik, és frissíti a sakkóra GUI elemeit.

Megvalósítja a következőket: **Observer** (o.18).

void ChessFrame.updateChessFrame ()

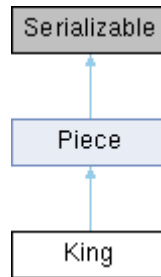
A sakkjáték GUI elemeinek frissítését végző függvény.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

ChessFrame.java

King osztályreferencia

A King osztály származási diagramja:



Publikus tagfüggvények

- **King** (boolean white)
- ArrayList< **Tile** > **legalMoves** (**Table** table, **Tile** tile)

Publikus tagfüggvények a(z) Piece osztályból származnak

- **Piece** (boolean white, String name)
- String **getName** ()
- Icon **getIcon** ()
- void **setIcon** (Icon icon)
- boolean **isWhite** ()
- abstract ArrayList< **Tile** > **legalMoves** (**Table** table, **Tile** tile)
- void **Move** (**Tile** from, **Tile** to)

Részletes leírás

A királyhoz tartozó osztály, a **Piece** osztályból származik.

Konstruktorok és destruktorok dokumentációja

King.King (boolean *white*)

King konstruktor. Beállítja a bábu színét, nevét és ikonját.

Paraméterek

<i>white</i>	A bábu színe.
--------------	---------------

Tagfüggvények dokumentációja

ArrayList< **Tile** > **King.legalMoves** (**Table** *table*, **Tile** *tile*)

Kiszámolja az adott mezőn elhelyezkedő bábu lehetséges lépéseit. Definiálja a **Piece** osztályban deklarált függvényt.

Paraméterek

<i>table</i>	Sakktábla.
<i>tile</i>	Az adott mező ahol a bábu áll.

Visszatérési érték

Lista, amely tartalmazza azokat a mezőket ahova a bábu léphet.

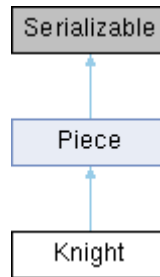
Újrimplementált ősök: **Piece** (o.22).

Ez a dokumentáció az osztályról a következő fájl alapján készült:

King.java

Knight osztályreferencia

A Knight osztály származási diagramja:



Publikus tagfüggvények

- **Knight** (boolean white)
- ArrayList< **Tile** > **legalMoves** (**Table** table, **Tile** tile)

Publikus tagfüggvények a(z) Piece osztályból származnak

- **Piece** (boolean white, String name)
- String **getName** ()
- Icon **getIcon** ()
- void **setIcon** (Icon icon)
- boolean **isWhite** ()
- abstract ArrayList< **Tile** > **legalMoves** (**Table** table, **Tile** tile)
- void **Move** (**Tile** from, **Tile** to)

Részletes leírás

A huszárhoz tartozó osztály, a **Piece** osztályból származik.

Konstruktorok és destruktorok dokumentációja

Knight.Knight (boolean *white*)

Knight konstruktor. Beállítja a bábu színét, nevét és ikonját.

Paraméterek

<i>white</i>	A bábu színe.
--------------	---------------

Tagfüggvények dokumentációja

ArrayList< **Tile** > **Knight.legalMoves** (**Table** *table*, **Tile** *tile*)

Kiszámolja az adott mezőn elhelyezkedő bábu lehetséges lépéseit. Definiálja a **Piece** osztályban deklarált függvényt.

Paraméterek

<i>table</i>	Sakktábla.
<i>tile</i>	Az adott mező ahol a bábu áll.

Visszatérési érték

Lista, amely tartalmazza azokat a mezőket ahova a bábu léphet.

Újraimplementált ősök: **Piece** (o.22).

Ez a dokumentáció az osztályról a következő fájl alapján készült:

Knight.java

Main osztályreferencia

Statikus publikus tagfüggvények

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

Tagfüggvények dokumentációja

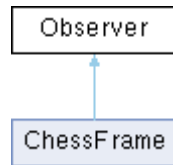
`static void Main.main (String[] args) [static]`

Ez a dokumentáció az osztályról a következő fájl alapján készült:

Main.java

Observer interfészreferencia

Az Observer osztály származási diagramja:



Publikus tagfüggvények

- void `update ()`

Részletes leírás

Interface a sakkóra és a GUI-t megvalósító objektum között.

Tagfüggvények dokumentációja

void Observer.update ()

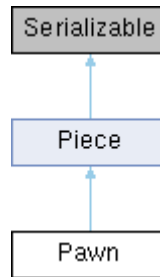
Megvalósítják a következők: **ChessFrame** (*o.12*).

Ez a dokumentáció az interfésről a következő fájl alapján készült:

Observer.java

Pawn osztályreferencia

A Pawn osztály származási diagramja:



Publikus tagfüggvények

- **Pawn** (boolean white)
- ArrayList< **Tile** > **legalMoves** (**Table** table, **Tile** tile)

Publikus tagfüggvények a(z) Piece osztályból származnak

- **Piece** (boolean white, String name)
- String **getName** ()
- Icon **getIcon** ()
- void **setIcon** (Icon icon)
- boolean **isWhite** ()
- abstract ArrayList< **Tile** > **legalMoves** (**Table** table, **Tile** tile)
- void **Move** (**Tile** from, **Tile** to)

Részletes leírás

A gyaloghoz tartozó osztály, a **Piece** osztályból származik.

Konstruktorok és destruktorok dokumentációja

Pawn.Pawn (boolean *white*)

Pawn konstruktor. Beállítja a bábu színét, nevét és ikonját.

Paraméterek

<i>white</i>	A bábu színe.
--------------	---------------

Tagfüggvények dokumentációja

ArrayList< **Tile** > **Pawn.legalMoves** (**Table** *table*, **Tile** *tile*)

Kiszámolja az adott mezőn elhelyezkedő bábu lehetséges lépéseit. Definiálja a **Piece** osztályban deklarált függvényt.

Paraméterek

<i>table</i>	Sakktábla.
<i>tile</i>	Az adott mező ahol a bábu áll.

Visszatérési érték

Lista, amely tartalmazza azokat a mezőket ahova a bábu léphet.

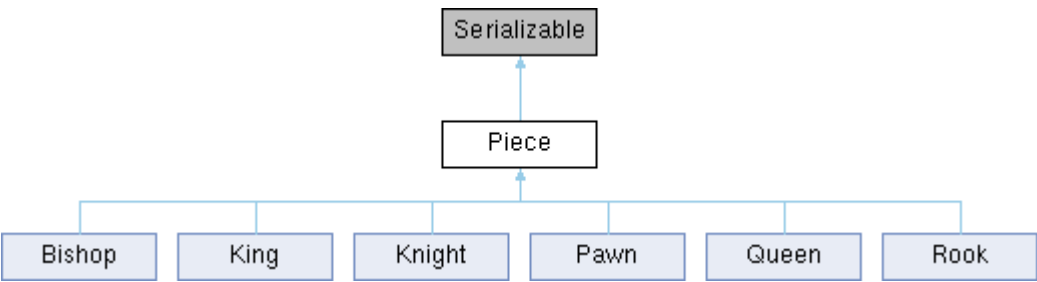
Újrimplementált ősök: **Piece** (o.22).

Ez a dokumentáció az osztályról a következő fájl alapján készült:

Pawn.java

Piece osztályreferencia

A Piece osztály származási diagramja:



Publikus tagfüggvények

- **Piece** (boolean white, String name)
- String **getName** ()
- Icon **getIcon** ()
- void **setIcon** (Icon icon)
- boolean **isWhite** ()
- abstract ArrayList< **Tile** > **legalMoves** (Table table, **Tile** tile)
- void **Move** (Tile from, **Tile** to)

Részletes leírás

Egy sakkfigurát megvalósító absztrakt osztály.

Konstruktorok és destruktorok dokumentációja

Piece.Piece (boolean *white*, String *name*)

Piece konstruktor

Paraméterek

<i>white</i>	Fehér vagy fekete.
<i>name</i>	A bábu neve.

Tagfüggvények dokumentációja

Icon **Piece.getIcon** ()

Getter a bábu ikonjához.

Visszatérési érték

A bábu ikonja.

String **Piece.getName** ()

Getter a bábu nevéhez.

Visszatérési érték

A bábu neve.

boolean **Piece.isWhite** ()

Getter a bábu színéhez.

Visszatérési érték

A bábu színe.

abstract ArrayList< Tile > Piece.legalMoves (Table *table*, Tile *tile*) [abstract]

Absztakt függvény, mely visszaadja a bábuk megengedett lépéseit.

Paraméterek

<i>table</i>	A tábla.
<i>tile</i>	A sakkmező, amelyen a bábu elhelyezkedik.

Visszatérési érték

A lehetséges lépések listája.

Újrimplementáló leszármazottak: **Bishop** (o.7), **King** (o.13), **Knight** (o.15), **Pawn** (o.19), **Queen** (o.23) és **Rook** (o.25).

void Piece.Move (Tile *from*, Tile *to*)

A bábu mozgását végzi mezőről mezőre.

Paraméterek

<i>from</i>	A sakkmező ahol a bábu tartózkodik.
<i>to</i>	A sakkmező ahova a bábút mozgatjuk.

void Piece.setIcon (Icon *icon*)

Setter a bábu ikonjához.

Paraméterek

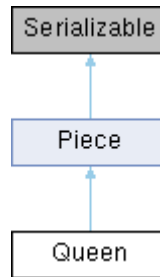
<i>icon</i>	A bábu ikonja.
-------------	----------------

Ez a dokumentáció az osztályról a következő fájl alapján készült:

Piece.java

Queen osztályreferencia

A Queen osztály származási diagramja:



Publikus tagfüggvények

- **Queen** (boolean white)
- ArrayList< **Tile** > **legalMoves** (**Table** table, **Tile** tile)

Publikus tagfüggvények a(z) Piece osztályból származnak

- **Piece** (boolean white, String name)
- String **getName** ()
- Icon **getIcon** ()
- void **setIcon** (Icon icon)
- boolean **isWhite** ()
- abstract ArrayList< **Tile** > **legalMoves** (**Table** table, **Tile** tile)
- void **Move** (**Tile** from, **Tile** to)

Részletes leírás

A vezérhez tartozó osztály, a **Piece** osztályból származik.

Konstruktorok és destruktorok dokumentációja

Queen.Queen (boolean *white*)

Queen konstruktor. Beállítja a bábu színét, nevét és ikonját.

Paraméterek

<i>white</i>	A bábu színe.
--------------	---------------

Tagfüggvények dokumentációja

ArrayList< **Tile** > **Queen.legalMoves** (**Table** *table*, **Tile** *tile*)

Kiszámolja az adott mezőn elhelyezkedő bábu lehetséges lépéseit. Definiálja a **Piece** osztályban deklarált függvényt.

Paraméterek

<i>table</i>	Sakktábla.
<i>tile</i>	Az adott mező ahol a bábu áll.

Visszatérési érték

Lista, amely tartalmazza azokat a mezőket ahova a bábu léphet.

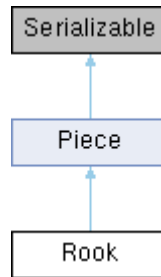
Újraimplementált ősök: **Piece** (o.22).

Ez a dokumentáció az osztályról a következő fájl alapján készült:

Queen.java

Rook osztályreferencia

A Rook osztály származási diagramja:



Publikus tagfüggvények

- **Rook** (boolean white)
- ArrayList< **Tile** > **legalMoves** (**Table** table, **Tile** tile)

Publikus tagfüggvények a(z) Piece osztályból származnak

- **Piece** (boolean white, String name)
- String **getName** ()
- Icon **getIcon** ()
- void **setIcon** (Icon icon)
- boolean **isWhite** ()
- abstract ArrayList< **Tile** > **legalMoves** (**Table** table, **Tile** tile)
- void **Move** (**Tile** from, **Tile** to)

Részletes leírás

A bástyához tartozó osztály, a **Piece** osztályból származik.

Konstruktorok és destruktorok dokumentációja

Rook.Rook (boolean white)

Rook konstruktor. Beállítja a bábu színét, nevét és ikonját.

Paraméterek

<i>white</i>	A bábu színe.
--------------	---------------

Tagfüggvények dokumentációja

ArrayList< Tile > Rook.legalMoves (Table table, Tile tile)

Kiszámolja az adott mezőn elhelyezkedő bábu lehetséges lépéseit. Definiálja a **Piece** osztályban deklarált függvényt.

Paraméterek

<i>table</i>	Sakktábla.
<i>tile</i>	Az adott mező ahol a bábu áll.

Visszatérési érték

Lista, amely tartalmazza azokat a mezőket ahova a bábu léphet.

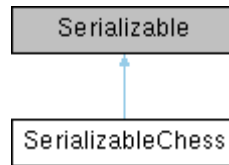
Újrimplementált ősök: **Piece** (o.22).

Ez a dokumentáció az osztályról a következő fájl alapján készült:

Rook.java

SerializableChess osztályreferencia

A SerializableChess osztály származási diagramja:



Publikus tagfüggvények

- **SerializableChess ()**
- **Table getTable ()**
- **ChessClock getChessClock ()**
- **boolean getWhiteMoves ()**
- **void setWhiteMoves (boolean v)**
- **boolean isRunning ()**
- **void setRunning (boolean running)**
- **String getLastMove ()**
- **void setLastMove (String lastMove)**
- **Tile tileOfKing (boolean whiteKing)**
- **boolean isChess (Tile tileOfKing)**
- **ArrayList< Tile > legalMoves (Tile tile)**
- **void Move (Tile tileFrom, Tile tileTo)**
- **boolean canMove (boolean isWhite)**

Részletes leírás

A sakkjáték logikáját megvalósító osztály. Implementálja a szerializáláshoz szükséges Serializable interface-t.

Konstruktorok és destruktorok dokumentációja

SerializableChess.SerializableChess ()

SerializableChess Konstruktor

- Létrehozza a sakktáblát azok mezejeivel.
- Létrehozza a sakkórát.*

Tagfüggvények dokumentációja

boolean SerializableChess.canMove (boolean *isWhite*)

A megadott színû bábuk esetén leellenõrzi, hogy van-e lehetőség lépésre. Amennyiben nincs, az vagy sakk-matt vagy patthelyzetet jelent és a játék végét jelenti.

Paraméterek

<i>isWhite</i>	Fehér vagy fekete.
----------------	--------------------

Visszatérési érték

Visszaadja, hogy van-e mozgatósi lehetőség.

ChessClock SerializableChess.getChessClock ()

Getter a sakkóra (chessClock) objektumhoz.

Visszatérési érték

A chessClock objektum.

String SerializableChess.getLastMove ()

Getter a lastMove tagváltozóhoz.

Visszatérési érték

String érték, mely az utolsó lépés információját tartalmazza.

Table SerializableChess.getTable ()

Getter a sakktábla (table) objektumhoz.

Visszatérési érték

A table objektum.

boolean SerializableChess.getWhiteMoves ()

Getter a whiteMoves tagváltozóhoz.

Visszatérési érték

Boolean érték, mely megadja hogy a fehér vagy a fekete játékos lépése következik.

boolean SerializableChess.isChess (Tile tileOfKing)

Megvizsgálja, hogy az adott mezőn tartózkodó király sakk állapotban van-e.

Paraméterek

<i>tileOfKing</i>	A király bábut tartalmazó mező.
-------------------	---------------------------------

Visszatérési érték

Megadja, hogy a király sakkbán van-e.

boolean SerializableChess.isRunning ()

Getter a running tagváltozóhoz.

Visszatérési érték

Boolean érték, mely megadja hogy a játék futó állapotban van-e.

ArrayList< Tile > SerializableChess.legalMoves (Tile tile)

Az adott mezőn álló bábu lehetséges lépéseit visszaadó függvény.

Összegyűjti az adott bábu lehetséges lépéseit. Az ellentétes színű király mezejét kiszűri. Kiszűri azokat a lépéseket, melyre lépve a saját király bábunk sakkbá kerülne.

Paraméterek

<i>tile</i>	Az adott bábut tartalmazó sakkmező.
-------------	-------------------------------------

Visszatérési érték

Azon sakkmezők listája ahova a bábu léphet.

void SerializableChess.Move (Tile tileFrom, Tile tileTo)

A sakkbábu mozgását végző függvény.

A bábu mozgatsán túl leellenőrzi és beállítja a király bábukat tartalmazó mezők sakk állapotát. Beállítja a lastMove String tartalmát.

Paraméterek

<i>tileFrom</i>	A bábut tartalmazó sakkmező.
<i>tileTo</i>	A kiválasztott célmező.

void SerializableChess.setLastMove (String *lastMove*)

Setter a lastMove tagváltozóhoz.

Paraméterek

<i>lastMove</i>	Az utolsó lépés információját tartalmazó String.
-----------------	--

void SerializableChess.setRunning (boolean *running*)

Setter a running tagváltozóhoz.

Paraméterek

<i>running</i>	Boolean érték, mely beállítja a játék futó állapotát.
----------------	---

void SerializableChess.setWhiteMoves (boolean *v*)

Setter a whiteMoves tagváltozóhoz.

Paraméterek

<i>v</i>	Boolean érték a játékos megadására.
----------	-------------------------------------

Tile SerializableChess.tileOfKing (boolean *whiteKing*)

Visszaadja a megadott színű király mezejét.

Paraméterek

<i>whiteKing</i>	Fehér vagy fekete
------------------	-------------------

Visszatérési érték

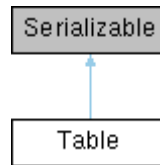
A király bábut tartalmazó mező.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

SerializableChess.java

Table osztályreferencia

A Table osztály származási diagramja:



Publikus tagfüggvények

- **Table ()**
- void **initTable ()**
- void **clearTable ()**
- void **placePieces ()**
- **Tile** **getTile** (int x, int y)

Részletes leírás

A sakktáblát leíró osztály

A sakkmezők kétdimenziós tömbjét tartalmazza.

Konstruktorok és destruktorok dokumentációja

Table.Table ()

Table konstruktor Létrehozza a sakkmezőkből álló 8x8-as táblát.

Tagfüggvények dokumentációja

void Table.clearTable ()

Üres tábla inicializálása.

Tile Table.getTile (int x, int y)

Getter a tábla egy x, y által meghatározott mezejéhez.

Paraméterek

x	A mező x koordinátája.
y	A mező y koordinátája.

Visszatérési érték

A meghatározott mező.

void Table.initTable ()

Inicializálja a tábla mezejeit.

void Table.placePieces ()

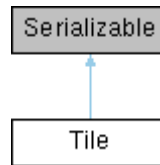
Elhelyezi a bábukat a játék kezdőállapotának megfelelően.

Ez a dokumentáció az osztályról a következő fájl alapján készült:

Table.java

Tile osztályreferencia

A Tile osztály származási diagramja:



Publikus tagfüggvények

- **Tile** (int x, int y)
- boolean **isChess** ()
- void **setChess** (boolean chess)
- **Piece** **getPieceOnTile** ()
- void **setPieceOnTile** (**Piece** pieceOnTile)
- int **getX** ()
- int **getY** ()
- boolean **isWhite** ()

Részletes leírás

Egy sakkmezőt leíró osztály.

Konstruktorok és destruktorok dokumentációja

Tile.Tile (int x, int y)

Tile konstruktor

Paraméterek

x	A mező x koordinátája.
y	A mező y koordinátája.

Tagfüggvények dokumentációja

Piece **Tile.getPieceOnTile** ()

Visszaadja a sakkmezőn található sakkfigurát.

Visszatérési érték

A sakkfigura, üres mező esetén null.

int **Tile.getX** ()

Getter a mező x koordinátájához.

Visszatérési érték

A mező x koordinátája.

int **Tile.getY** ()

Getter a mező y koordinátájához.

Visszatérési érték

A mező y koordinátája.

boolean Tile.isChess ()

A király bábut tartalmazó mező esetén megadja, hogy a király sakkban van-e.

Visszatérési érték

Sakk státusz.

boolean Tile.isWhite ()

A sakkmező színe.

Visszatérési érték

Fehér vagy fekete.

void Tile.setChess (boolean *chess*)

A királyt tartalmazó mezőre beállítja, hogy a király sakkban van-e.

Paraméterek

<i>chess</i>	Sakk státusz.
--------------	---------------

void Tile.setPieceOnTile (Piece *pieceOnTile*)

Elhelyez egy sakkfigurát a mezőn.

Paraméterek

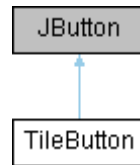
<i>pieceOnTile</i>	A sakkfigura.
--------------------	---------------

Ez a dokumentáció az osztályról a következő fájl alapján készült:

Tile.java

TileButton osztályreferencia

A TileButton osztály származási diagramja:



Publikus tagfüggvények

- **TileButton** (**Tile** *tile*)
- **Tile** **getTile** ()
- void **setTile** (**Tile** *tile*)
- boolean **isSelected** ()
- void **setSelected** (boolean *selected*)
- boolean **isTargeted** ()
- void **setTargeted** (boolean *targeted*)

Részletes leírás

A sakkmezőt megjelenítő GUI elem, a JButton származéka.

Konstruktorok és destruktorok dokumentációja

TileButton.TileButton (**Tile** *tile*)

TileButton konstruktor

Paraméterek

<i>tile</i>	A kapcsolt sakkmező (Tile) objektum referenciája.
-------------	--

Tagfüggvények dokumentációja

Tile **TileButton.getTile** ()

Getter a kapcsolódó sakkmező (**Tile**) objektumhoz.

Visszatérési érték

A sakkmező referenciája.

boolean **TileButton.isSelected** ()

Getter a selected tagváltozóhoz.

Visszatérési érték

A selected tagváltozó értéke.

boolean **TileButton.isTargeted** ()

Getter a targeted tagváltozóhoz.

Visszatérési érték

A targeted tagváltozó értéke.

void **TileButton.setSelected** (boolean *selected*)

Setter a selected tagváltozóhoz.

Paraméterek

<i>selected</i>	A selected tagváltozó új értéke.
-----------------	----------------------------------

void TileButton.setTargeted (boolean *targeted*)

Setter a targeted tagváltozóhoz.

Paraméterek

<i>targeted</i>	A targeted tagváltozó új értéke.
-----------------	----------------------------------

void TileButton.setTile (Tile *tile*)

Setter, mely beállítja a kapcsolódó sakkmezőt (**Tile**).

Paraméterek

<i>tile</i>	A sakkmező referenciája.
-------------	--------------------------

Ez a dokumentáció az osztályról a következő fájl alapján készült:

TileButton.java