

Chess

Wygenerowano przez Doxygen 1.9.1

1 Indeks hierarchiczny	1
1.1 Hierarchia klas	1
2 Indeks klas	3
2.1 Lista klas	3
3 Dokumentacja klas	5
3.1 Dokumentacja klasy Bishop	5
3.2 Dokumentacja klasy Chessboard	5
3.2.1 Opis szczegółowy	6
3.3 Dokumentacja klasy ChessboardPrinter	6
3.3.1 Opis szczegółowy	7
3.4 Dokumentacja klasy FileMoveWriter	7
3.5 Dokumentacja klasy Game	7
3.6 Dokumentacja klasy HumanPlayer	8
3.7 Dokumentacja klasy King	8
3.8 Dokumentacja klasy Knight	9
3.9 Dokumentacja struktury Move	9
3.9.1 Opis szczegółowy	10
3.10 Dokumentacja klasy Pawn	10
3.11 Dokumentacja klasy Piece	10
3.11.1 Opis szczegółowy	11
3.12 Dokumentacja klasy Player	12
3.13 Dokumentacja klasy Queen	12
3.14 Dokumentacja klasy RandomPlayer	13
3.15 Dokumentacja klasy Rook	13
3.16 Dokumentacja struktury ChessboardPrinter::TileInfo	14
Indeks	15

Rozdział 1

Indeks hierarchiczny

1.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

Chessboard	5
ChessboardPrinter	6
FileMoveWriter	7
Game	7
Move	9
Piece	10
Bishop	5
King	8
Knight	9
Pawn	10
Queen	12
Rook	13
Player	12
HumanPlayer	8
RandomPlayer	13
ChessboardPrinter::TileInfo	14

Rozdział 2

Indeks klas

2.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

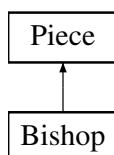
Bishop	5
Chessboard	5
ChessboardPrinter	6
FileMoveWriter	7
Game	7
HumanPlayer	8
King	8
Knight	9
Move	9
Pawn	10
Piece	10
Player	12
Queen	12
RandomPlayer	13
Rook	13
ChessboardPrinter::TileInfo	14

Rozdział 3

Dokumentacja klas

3.1 Dokumentacja klasy Bishop

Diagram dziedziczenia dla Bishop



Metody publiczne

- **Bishop** ([Chessboard](#) &chessboard, Team t, char pos)
- virtual Type **getType** () const override
- void **generateMoves** (std::vector< [Move](#) > &moves) const override

Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej klasy została wygenerowana z plików:

- ChessLib/ChessPieces/Bishop.h
- ChessLib/ChessPieces/Bishop.cpp

3.2 Dokumentacja klasy Chessboard

```
#include <Chessboard.h>
```

Metody publiczne

- **Chessboard** (const char *fen="rnbqkbnr/pppppppp/8/8/8/PPPPPPPP/RNBQKBNR")
- void **makeMove** ([Move](#) move)
- void **undoMove** ([Move](#) move)
- void **searchDirection** (char pos, char direction, std::vector< std::pair< char, [Piece](#) * >> &buffer) const
- void **searchDirection** (char pos, char direction, char max, std::vector< std::pair< char, [Piece](#) * >> &buffer) const
- bool **canMoveStep** (char current, char dest) const
- std::string **getFenString** () const
- void **getMoves** (char pos, std::vector< [Move](#) > &moves) const
- void **getTeamOffsets** (Team team, std::vector< char > &offsets) const
- void **switchPromotion** (char position, [Piece::Type](#) type)
- [King](#) * **getKing** (Team team) const
- [Piece](#) * **getPiece** (char pos) const

Przyjaciele

- class **Piece**

3.2.1 Opis szczegółowy

Klasa przechowująca pionki na szachownicy. Głównym zadaniem klasy jest zmiana stanów pionków podczas gry oraz odnajdywanie ich pozycji na planszy.

Dokumentacja dla tej klasy została wygenerowana z plików:

- ChessLib/Chessboard.h
- ChessLib/Chessboard.cpp

3.3 Dokumentacja klasy ChessboardPrinter

```
#include <ChessboardPrinter.h>
```

Komponenty

- struct [TileInfo](#)

Typy publiczne

- enum class **Color** : char {
Black , **Red** , **Green** , **Yellow** ,
Blue , **Magenta** , **Cyan** , **White** }

Metody publiczne

- **ChessboardPrinter** (Color player1=COLOR_BLUE, Color player2=COLOR_RED, Color background1=COLOR_←_WHITE, Color background2=COLOR_BLACK)
- void **print** (std::string fen) const
- void **clear** () const
- void **printMoves** (char pos, const [Chessboard](#) &board) const
- void **printTeam** (Team team, const [Chessboard](#) &board) const

3.3.1 Opis szczegółowy

Klasa wypisująca aktualny stan planszy do konsoli. Klasa posiada metody 'podświetlające' możliwe do wyboru pionki oraz wizualizuje możliwe do wykonania ruchy przez wybrany pionek.

Dokumentacja dla tej klasy została wygenerowana z plików:

- ChessLib/ChessboardPrinter.h
- ChessLib/ChessboardPrinter.cpp

3.4 Dokumentacja klasy FileMoveWriter

Metody publiczne

- **FileMoveWriter** (std::string filename="Moves.txt")
- std::string **getFilename** () const
- void **write** (const [Move](#) &move, const [Chessboard](#) &board, GameStatus status)
- void **write** (const [Move](#) &move, const [Chessboard](#) &board, GameStatus status, int moveNumber)

Dokumentacja dla tej klasy została wygenerowana z pliku:

- ChessLib/FileMoveWriter.h

3.5 Dokumentacja klasy Game

Metody publiczne

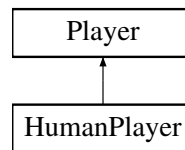
- **Game** (Player::Type first, Player::Type second)
- void **play** (unsigned delay=0)

Dokumentacja dla tej klasy została wygenerowana z plików:

- ChessLib/Game.h
- ChessLib/Game.cpp

3.6 Dokumentacja klasy HumanPlayer

Diagram dziedziczenia dla HumanPlayer



Metody publiczne

- **HumanPlayer** (`Team team`, `std::istream &stream=std::cin`)
- virtual `Move` **getMove** (`const Chessboard &board`, `const ChessboardPrinter &printer`) const
- virtual `Piece::Type` **getPromotion** () const

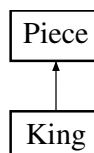
Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej klasy została wygenerowana z plików:

- `ChessLib/Players/HumanPlayer.h`
- `ChessLib/Players/HumanPlayer.cpp`

3.7 Dokumentacja klasy King

Diagram dziedziczenia dla King



Metody publiczne

- **King** (`Chessboard &chessboard`, `Team t`, `char pos`)
- virtual `Type` **getType** () const override
- void **generateMoves** (`std::vector< Move > &moves`) const override
- bool **willIndangereKing** (`const Move &move`) const
- char **inCheck** () const
- void **getCastling** (`std::vector< Move > &moves`) const

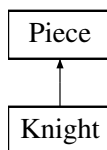
Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej klasy została wygenerowana z plików:

- `ChessLib/ChessPieces/King.h`
- `ChessLib/ChessPieces/King.cpp`

3.8 Dokumentacja klasy Knight

Diagram dziedziczenia dla Knight



Metody publiczne

- **Knight** ([Chessboard](#) &chessboard, Team t, char pos)
- virtual Type **getType** () const override
- void **generateMoves** (std::vector< [Move](#) > &moves) const override

Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej klasy została wygenerowana z plików:

- ChessLib/ChessPieces/Knight.h
- ChessLib/ChessPieces/Knight.cpp

3.9 Dokumentacja struktury Move

```
#include <Piece.h>
```

Typy publiczne

- enum class **Type** : char { }

Metody publiczne

- bool **operator**< (const [Move](#) &other) const

Atrybuty publiczne

- char **cStart**
- char **cDest**
- char **oStart**
- char **oDest**
- enum Move::Type **type**
- Piece::Type **promoteFigure** = Piece::Type::None

3.9.1 Opis szczegółowy

Klasa na podstawie, której pionki wykonują ruch. `cStart` i `cDest` oznaczają pozycję startową i końcową pionka wykonującego ruch. `oStart` i `oDest` oznaczają pozycję startową i końcową pionka na którego dany ruch wywrze wpływ np. pozycja zbijanego pionka `type` określa rodzaj ruchu. Rozróżniamy 4 rodzaje ruchu: zwykły ruch, zabicie, roszadę i bicie w przelocie `promoteFigure` określa na jaką figurę zmieni się pionek po wykonaniu ruchu

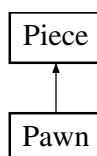
W przypadku roszady `cStart` i `cDest` zawsze odnosi się do króla, a `oStart`, `oDest` do wieży

Dokumentacja dla tej struktury została wygenerowana z plików:

- ChessLib/Piece.h
- ChessLib/Piece.cpp

3.10 Dokumentacja klasy Pawn

Diagram dziedziczenia dla Pawn



Metody publiczne

- **Pawn** ([Chessboard](#) &chessboard, Team team, char pos)
- virtual Type **getType** () const override
- void **generateMoves** (std::vector< [Move](#) > &moves) const override

Dodatkowe Dziedziczone Składowe

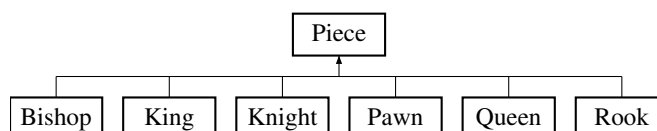
Dokumentacja dla tej klasy została wygenerowana z plików:

- ChessLib/ChessPieces/Pawn.h
- ChessLib/ChessPieces/Pawn.cpp

3.11 Dokumentacja klasy Piece

```
#include <Piece.h>
```

Diagram dziedziczenia dla Piece



Typy publiczne

- enum class **Type** : char {
 None = ' ', **King** = 'k', **Queen** = 'q', **Bishop** = 'b',
 Knight = 'n', **Rook** = 'r', **Pawn** = 'p' }
- enum class **State** : char { **NotMoved** , **FirstMove** , **Moved** }

Metody publiczne

- **Piece** ([Chessboard](#) &chessboard, Team t, char pos)
- Team **getTeam** () const
- State **getState** () const
- void **setState** (State s)
- void **setPos** (char pos)
- void **saveState** ()
- void **restoreState** ()
- virtual void **onMove** (char pos)
- virtual void **onNextTurn** ()
- **operator char** () const
- bool **canMove** () const
- bool **canPromote** () const
- bool **canPromote** (char pos) const
- virtual Type **getType** () const =0
- virtual char **getPos** () const
- virtual void **getMoves** (std::vector< [Move](#) > &moves) const
- virtual void **generateMoves** (std::vector< [Move](#) > &moves) const =0

Metody chronione

- int **getMaxDistance** () const
- void **getMoves** (const std::vector< char > &directions, std::vector< [Move](#) > &moves) const
- void **removeIllegalMoves** (std::vector< [Move](#) > &moves) const

Atrybuty chronione

- [Chessboard](#) & chessboard
- const Team team
- char pos
- State cState
- State lState

3.11.1 Opis szczegółowy

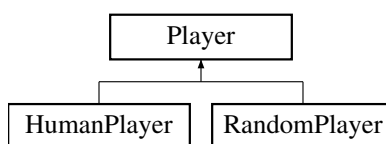
Klasa bazowa dla wszystkich pionków. Głównym zadaniem klasy jest generowanie możliwych do wykonania ruchów na podstawie pozycji własnej i innych pionków na szachownicy, które nie są sprzeczne z regułami gry.

Dokumentacja dla tej klasy została wygenerowana z plików:

- ChessLib/Piece.h
- ChessLib/Piece.cpp

3.12 Dokumentacja klasy Player

Diagram dziedziczenia dla Player



Typy publiczne

- enum class **Type** { **HumanPlayer** , **RandomPlayer** }

Metody publiczne

- **Player** (Team team)
- Team **getTeam** () const
- virtual **Move** **getMove** (const **Chessboard** &board, const **ChessboardPrinter** &printer) const =0
- virtual Piece::Type **getPromotion** () const =0

Atrybuty chronione

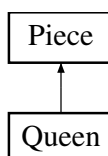
- Team **team**

Dokumentacja dla tej klasy została wygenerowana z pliku:

- ChessLib/Players/Player.h

3.13 Dokumentacja klasy Queen

Diagram dziedziczenia dla Queen



Metody publiczne

- **Queen** (**Chessboard** &chessboard, Team team, char pos)
- virtual Type **getType** () const override
- void **generateMoves** (std::vector< **Move** > &moves) const override

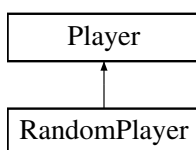
Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej klasy została wygenerowana z plików:

- ChessLib/ChessPieces/Queen.h
- ChessLib/ChessPieces/Queen.cpp

3.14 Dokumentacja klasy RandomPlayer

Diagram dziedziczenia dla RandomPlayer



Metody publiczne

- **RandomPlayer** (Team team)
- virtual `Move` **getMove** (const `Chessboard` &board, const `ChessboardPrinter` &printer) const
- virtual `Piece::Type` **getPromotion** () const

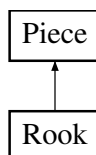
Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej klasy została wygenerowana z plików:

- ChessLib/Players/RandomPlayer.h
- ChessLib/Players/RandomPlayer.cpp

3.15 Dokumentacja klasy Rook

Diagram dziedziczenia dla Rook



Metody publiczne

- **Rook** (`Chessboard` &chessboard, Team team, char pos)
- virtual `Type` **getType** () const override
- void **generateMoves** (std::vector< `Move` > &moves) const override
- void **getCastling** (std::vector< `Move` > &moves) const

Dodatkowe Dziedziczone Składowe

Dokumentacja dla tej klasy została wygenerowana z plików:

- ChessLib/ChessPieces/Rook.h
- ChessLib/ChessPieces/Rook.cpp

3.16 Dokumentacja struktury ChessboardPrinter::TileInfo

Atrybuty publiczne

- char **character**
- Color **charColor**
- Color **bgColor**

Dokumentacja dla tej struktury została wygenerowana z pliku:

- ChessLib/ChessboardPrinter.h

Indeks

Bishop, [5](#)

Chessboard, [5](#)

ChessboardPrinter, [6](#)

ChessboardPrinter::TileInfo, [14](#)

FileMoveWriter, [7](#)

Game, [7](#)

HumanPlayer, [8](#)

King, [8](#)

Knight, [9](#)

Move, [9](#)

Pawn, [10](#)

Piece, [10](#)

Player, [12](#)

Queen, [12](#)

RandomPlayer, [13](#)

Rook, [13](#)