Package connect4

package connect4

All Classes and Interfaces | Interfaces | Classes | Enum Classes | Record Classes

Class	Description
Game	A game of Connect 4.
Game.Status	Represents the status of a Game.
Game.Status.Ended	Possible status of a Game; represents that the Game has ended with a result.
Game.Status.Turn	Possible status of a Game; represents that it's some Player's turn.
GameResult	The result of a Game: a RED win, YELLOW win, or draw.
Player	One of the two players in Connect 4.

Package connect4

Class Game

java.lang.Object
    connect4.Game

```
public class Game
extends Object
```

A game of Connect 4. Consists of a board and whose turn it is (or the winner, if any).

Methods on `Game` can change the board and the `Game`'s `Status`.

Nested Class Summary

Nested Classes

Modifier and Type	Class	Description
static interface	`Game.Status`	Represents the status of a `Game`.

Field Summary

Fields

Modifier and Type	Field	Description
static final int	`HEIGHT`	Height of a Connect 4 board.
static final int	`WIDTH`	Width of a Connect 4 board.

Constructor Summary

Constructors

Constructor	Description
`Game(Game game)`	Create a deep copy of `game`.
`Game(Player startingPlayer)`	Construct a new `Game` with an empty board, with `startingPlayer` moving first.

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method	Description
`Optional<Player>[][]`	`getBoard()`	Return an array representing the Connect 4 board for this `Game`.
`Optional<GameResult>`	`getGameResult()`	Returns the result of the `Game`, if it has ended.
`Set<Integer>`	`getPossibleMoves()`	Returns the set of all columns (0-indexed) that are valid moves for either player.
`Optional<Player>`	`getSquare(int row, int col)`	Get which player's chip, if any, occupies the square located at the given row and column.
`Game.Status`	`getStatus()`	Return the current `Status` of the `Game`.
`Optional<Player>`	`getTurn()`	Returns the `Player` whose turn it is to move, if the game has not ended.
void	`move(Player player, int col)`	Update the board in response to a player placing a chip into `col`, 0-indexed.

Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Field Details

HEIGHT

```
public static final int HEIGHT
```

Height of a Connect 4 board.

See Also:

Constant Field Values

WIDTH

`public static final int WIDTH`

Width of a Connect 4 board.

See Also:
Constant Field Values

Constructor Details

Game

`public Game(Player startingPlayer)`

Construct a new `Game` with an empty board, with `startingPlayer` moving first.

Parameters:
`startingPlayer` - `Player` who moves first in the returned `Game`

Game

`public Game(Game game)`

Create a deep copy of `game`.

Parameters:
`game` - `Game` to copy

Method Details

getStatus

`public Game.Status getStatus()`

Return the current `Status` of the `Game`.

The `Status` of a `Game` consists of whose turn it is (if the game hasn't ended); or, if the game has ended, the winner (if any).

Returns:
the `Status` of this `Game`

See Also:
`Game.Status`

getTurn

`public Optional<Player> getTurn()`

Returns the `Player` whose turn it is to move, if the game has not ended.

API Note:
This is a convenience function. You can also pattern match on `getStatus()`'s return value to ask whose turn it is; the benefit is that this handles the case where the game has already ended.

Returns:
`Player` whose turn it is, if the game has not ended

getGameResult

`public Optional<GameResult> getGameResult()`

Returns the result of the `Game`, if it has ended.

API Note:
This is a convenience function. You can also use `getStatus()` to check the result of the `Game` while handling all cases.

Returns:
result of the `Game`, if it has ended

getSquare

```
public Optional<Player> getSquare(int row,
                                  int col)
```

Get which player's chip, if any, occupies the square located at the given row and column.

If the square is empty, this method returns `Optional.empty()`.

`row` and `col` must be within the bounds of the Connect 4 board: `0 <= row < HEIGHT` and `0 <= col < WIDTH`.

Parameters:

`row` - Row of board. The top row is index 0.

`col` - Column of board. The leftmost column is index 0.

Returns:

`Player` who occupies the given square, if any.

getBoard

```
public Optional<Player>[][] getBoard()
```

Return an array representing the Connect 4 board for this `Game`.

If `board` is the return value of this function, the element at `board[row][col]` is equal to `getSquare(row, col)`.

The returned array is a new array. Writing to it does not affect this `Game`.

Returns:

array representing the Connect 4 board

move

```
public void move(Player player,
                 int col)
```

Update the board in response to a player placing a chip into `col`, 0-indexed.

If the move cannot be made, throws an exception and does not mutate the board.

Preconditions for a successful move:

- `status.equals(new Status.Turn(player))`
- `getPossibleMoves().contains(col)`

Parameters:

`player` - `Player` making this move

`col` - 0-indexed column

Throws:

`IllegalArgumentException` - if `col` is out of bounds

`IllegalStateException` - if the player cannot legally move in `col`, due to game over or trying to play out of turn, or because that column is full

getPossibleMoves

```
public Set<Integer> getPossibleMoves()
```

Returns the set of all columns (0-indexed) that are valid moves for either player.

Returns:

`Set` of all valid moves

Package connect4

Interface Game.Status

All Known Implementing Classes:

`Game.Status.Ended`, `Game.Status.Turn`

Enclosing class:

`Game`

```
public static sealed interface Game.Status
permits Game.Status.Turn, Game.Status.Ended
```

Represents the status of a `Game`. It can be some `Player`'s turn, or a `GameResult` if the game has ended.

Nested Class Summary

Nested Classes

Modifier and Type	Interface	Description
`static final record`	**`Game.Status.Ended`**	Possible status of a `Game`; represents that the `Game` has ended with a result.
`static final record`	**`Game.Status.Turn`**	Possible status of a `Game`; represents that it's some `Player`'s turn.

Package connect4

Record Class Game.Status.Ended

java.lang.Object
  java.lang.Record
    connect4.Game.Status.Ended

Record Components:

`result` - result of the `Game`

All Implemented Interfaces:

`Game.Status`

Enclosing interface:

`Game.Status`

```
public static record Game.Status.Ended(GameResult result)
extends Record
implements Game.Status
```

Possible status of a `Game`; represents that the `Game` has ended with a result.

Nested Class Summary

Nested classes/interfaces inherited from interface connect4.Game.Status

`Game.Status.Ended`, `Game.Status.Turn`

Constructor Summary

Constructors

Constructor	Description
`Ended(GameResult result)`	Creates an instance of a Ended record class.

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method	Description
`final boolean`	`equals(Object o)`	Indicates whether some other object is "equal to" this one.
`final int`	`hashCode()`	Returns a hash code value for this object.
`GameResult`	`result()`	Returns the value of the `result` record component.
`final String`	`toString()`	Returns a string representation of this record class.

Methods inherited from class java.lang.Object

`clone`, `finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

Constructor Details

Ended

```
public Ended(GameResult result)
```

Creates an instance of a Ended record class.

Parameters:

`result` - the value for the `result` record component

Method Details

toString

```
public final String toString()
```

Returns a string representation of this record class. The representation contains the name of the class, followed by the name and value of each of the record components.

Specified by:

`toString` in class `Record`

Returns:

a string representation of this object

hashCode

```
public final int hashCode()
```

Returns a hash code value for this object. The value is derived from the hash code of each of the record components.

Specified by:

`hashCode` in class `Record`

Returns:

a hash code value for this object

equals

```
public final boolean equals(Object o)
```

Indicates whether some other object is "equal to" this one. The objects are equal if the other object is of the same class and if all the record components are equal. All components in this record class are compared with `Objects::equals(Object,Object)`.

Specified by:

`equals` in class `Record`

Parameters:

`o` - the object with which to compare

Returns:

`true` if this object is the same as the `o` argument; `false` otherwise.

result

```
public GameResult result()
```

Returns the value of the `result` record component.

Returns:

the value of the `result` record component

Package connect4

Record Class Game.Status.Turn

java.lang.Object
  java.lang.Record
    connect4.Game.Status.Turn

Record Components:

`player` - `Player` whose turn it is

All Implemented Interfaces:

`Game.Status`

Enclosing interface:

`Game.Status`

```
public static record Game.Status.Turn(Player player)
extends Record
implements Game.Status
```

Possible status of a `Game`; represents that it's some `Player`'s turn.

Nested Class Summary

Nested classes/interfaces inherited from interface connect4.Game.Status

`Game.Status.Ended`, `Game.Status.Turn`

Constructor Summary

Constructors

Constructor	Description
`Turn(Player player)`	Creates an instance of a `Turn` record class.

Method Summary

All Methods | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method	Description
`final boolean`	`equals(Object o)`	Indicates whether some other object is "equal to" this one.
`final int`	`hashCode()`	Returns a hash code value for this object.
`Player`	`player()`	Returns the value of the `player` record component.
`final String`	`toString()`	Returns a string representation of this record class.

Methods inherited from class java.lang.Object

`clone`, `finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

Constructor Details

Turn

```
public Turn(Player player)
```

Creates an instance of a `Turn` record class.

Parameters:

`player` - the value for the `player` record component

Method Details

toString

```
public final String toString()
```

Returns a string representation of this record class. The representation contains the name of the class, followed by the name and value of each of the record components.

Specified by:

`toString` in class `Record`

Returns:

a string representation of this object

hashCode

```
public final int hashCode()
```

Returns a hash code value for this object. The value is derived from the hash code of each of the record components.

Specified by:

`hashCode` in class `Record`

Returns:

a hash code value for this object

equals

```
public final boolean equals(Object o)
```

Indicates whether some other object is "equal to" this one. The objects are equal if the other object is of the same class and if all the record components are equal. All components in this record class are compared with `Objects::equals(Object,Object)`.

Specified by:

`equals` in class `Record`

Parameters:

`o` - the object with which to compare

Returns:

`true` if this object is the same as the `o` argument; `false` otherwise.

player

```
public Player player()
```

Returns the value of the `player` record component.

Returns:

the value of the `player` record component

Package connect4

Enum Class GameResult

java.lang.Object
  java.lang.Enum<GameResult>
    connect4.GameResult

All Implemented Interfaces:

Serializable, Comparable<GameResult>, Constable

```
public enum GameResult
extends Enum<GameResult>
```

The result of a Game: a RED win, YELLOW win, or draw.

Nested Class Summary

Nested classes/interfaces inherited from class java.lang.Enum

Enum.EnumDesc<E extends Enum<E>>

Enum Constant Summary

Enum Constants

Enum Constant	Description
DRAW	Represents a draw.
RED_WIN	Represents a RED win.
YELLOW_WIN	Represents a YELLOW win.

Method Summary

All Methods | **Static Methods** | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method	Description
static GameResult	fromWinner(Player player)	Create a new GameResult, representing the given player's win.
static GameResult	valueOf(String name)	Returns the enum constant of this class with the specified name.
static GameResult[]	values()	Returns an array containing the constants of this enum class, in the order they are declared.
Optional<Player>	winner()	Get the winner for this GameResult, if any.

Methods inherited from class java.lang.Enum

clone, compareTo, describeConstable, equals, finalize, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

Enum Constant Details

RED_WIN

```
public static final GameResult RED_WIN
```

Represents a RED win.

YELLOW_WIN

```
public static final GameResult YELLOW_WIN
```

Represents a YELLOW win.

DRAW

```
public static final GameResult DRAW
```

Represents a draw.

Method Details

values

`public static GameResult[] values()`

Returns an array containing the constants of this enum class, in the order they are declared.

Returns:

an array containing the constants of this enum class, in the order they are declared

valueOf

`public static GameResult valueOf(String name)`

Returns the enum constant of this class with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this class. (Extraneous whitespace characters are not permitted.)

Parameters:

`name` - the name of the enum constant to be returned.

Returns:

the enum constant with the specified name

Throws:

`IllegalArgumentException` - if this enum class has no constant with the specified name

`NullPointerException` - if the argument is null

winner

`public Optional<Player> winner()`

Get the winner for this `GameResult`, if any.

If this result is a `RED` or `YELLOW` win, return the winning player. If this result is a draw, return `Optional.empty()`.

Returns:

Player who won, if any

fromWinner

`public static GameResult fromWinner(Player player)`

Create a new `GameResult`, representing the given player's win.

Parameters:

`player` - winning player

Returns:

`GameResult` representing `player`'s win

Package connect4

Enum Class Player

java.lang.Object
    java.lang.Enum<Player>
        connect4.Player

All Implemented Interfaces:

Serializable, Comparable<Player>, Constable

```
public enum Player
extends Enum<Player>
```

One of the two players in Connect 4.

Nested Class Summary

Nested classes/interfaces inherited from class java.lang.Enum

Enum.EnumDesc<E extends Enum<E>>

Enum Constant Summary

Enum Constants

Enum Constant	Description
RED	Player who uses the red chips, in Connect 4.
YELLOW	Player who uses the yellow chips, in Connect 4.

Method Summary

All Methods | **Static Methods** | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method	Description
Player	**opponent**()	Get this Player's opponent.
static **Player**	**valueOf**(**String** name)	Returns the enum constant of this class with the specified name.
static **Player**[]	**values**()	Returns an array containing the constants of this enum class, in the order they are declared.

Methods inherited from class java.lang.Enum

clone, compareTo, describeConstable, equals, finalize, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object

getClass, notify, notifyAll, wait, wait, wait

Enum Constant Details

RED

```
public static final Player RED
```

Player who uses the red chips, in Connect 4.

YELLOW

```
public static final Player YELLOW
```

Player who uses the yellow chips, in Connect 4.

Method Details

values

```
public static Player[] values()
```

Returns an array containing the constants of this enum class, in the order they are declared.

Returns:

an array containing the constants of this enum class, in the order they are declared

valueOf

```
public static Player valueOf(String name)
```

Returns the enum constant of this class with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this class. (Extraneous whitespace characters are not permitted.)

Parameters:

`name` - the name of the enum constant to be returned.

Returns:

the enum constant with the specified name

Throws:

`IllegalArgumentException` - if this enum class has no constant with the specified name

`NullPointerException` - if the argument is null

opponent

```
public Player opponent()
```

Get this `Player`'s opponent.

`RED`'s opponent is `YELLOW` and vice versa.

Returns:

this `Player`'s opponent