

Packages

Package	Description
hk.edu.polyu.comp.comp3211.monopoly	
hk.edu.polyu.comp.comp3211.monopoly.controller	
hk.edu.polyu.comp.comp3211.monopoly.model	
hk.edu.polyu.comp.comp3211.monopoly.model.squares	
hk.edu.polyu.comp.comp3211.monopoly.view	

Package hk.edu.polyu.comp.comp3211.monopoly

Class Summary	
Class	Description
Main	

Package hk.edu.polyu.comp.comp3211.monopoly.controller

Interface Summary

Interface	Description
IBase	

Class Summary

Class	Description
EndGame	
Game	
StartMenu	

Package hk.edu.polyu.comp.comp3211.monopoly.model

Class Summary	
Class	Description
Board	A board, containing players, squares, and game status
Player	A player and its status in the game

Package hk.edu.polyu.comp.comp3211.monopoly.model.squares

Interface Summary

Interface	Description
ISquare	Any square of the board

Class Summary

Class	Description
Chance	The Chance square of the board
Free	The Free-Parking square of the board
Go	The Go square of the board
Jail	The In-Jail/Just-Visiting square of the board
Oops	The Go-to-Jail square of the board
Property	The Property squares of the board
Tax	The Income-Tax square of the board

Package hk.edu.polyu.comp.comp3211.monopoly.view

Class Summary

Class	Description
Printer	

Package [hk.edu.polyu.comp.comp3211.monopoly](#)

Class Main

java.lang.Object
hk.edu.polyu.comp.comp3211.monopoly.Main

```
public class Main
extends java.lang.Object
```

Constructor Summary

Constructors	
Constructor	Description
Main ()	

Method Summary

All Methods	Static Methods	Concrete Methods
Modifier and Type	Method	Description
static java.util.Scanner	GetScanner ()	If a method needs to get user input, use this static method to get the scanner
static void	main(java.lang.String[] args)	
static void	setUI(IBase ui)	Switch user interface

Methods inherited from class java.lang.Object

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

Constructor Detail

Main
public Main()

Method Detail

main

```
public static void main(java.lang.String[] args)
```

GetScanner

```
public static java.util.Scanner GetScanner()
```

If a method needs to get user input, use this static method to get the scanner

Returns:

The scanner

setUI

```
public static void setUI(IBase ui)
```

Switch user interface

Parameters:

`ui` - the new interface

Package [hk.edu.polyu.comp.comp3211.monopoly.controller](#)

Interface IBase

All Known Implementing Classes:

[EndGame](#), [Game](#), [StartMenu](#)

```
public interface IBase
```

Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type	Method	Description
void	<code>terminal</code> ()	Prompt , parse and process user commands

Method Detail

<code>terminal</code>
<pre>void terminal()</pre>
Prompt , parse and process user commands

Package `hk.edu.polyu.comp.comp3211.monopoly.controller`

Class EndGame

`java.lang.Object`
`hk.edu.polyu.comp.comp3211.monopoly.controller.EndGame`

All Implemented Interfaces:

`IBase`

```
public class EndGame
extends java.lang.Object
implements IBase
```

Constructor Summary

Constructors	
Constructor	Description
<code>EndGame ()</code>	

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
void	<code>terminal ()</code>	Print the game over message, and ask if the player want to restart, load or quit.

Methods inherited from class java.lang.Object

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

Constructor Detail

EndGame
<pre>public EndGame ()</pre>

Method Detail

terminal

```
public void terminal()
```

Print the game over message, and ask if the player want to restart, load or quit.

Specified by:

`terminal` in interface `IBase`

Package `hk.edu.polyu.comp.comp3211.monopoly.controller`

Class Game

`java.lang.Object`
`hk.edu.polyu.comp.comp3211.monopoly.controller.Game`

All Implemented Interfaces:

`IBase`

```
public class Game
extends java.lang.Object
implements IBase
```

Constructor Summary

Constructors	
Constructor	Description
<code>Game ()</code>	

Method Summary

All Methods		Instance Methods	Concrete Methods
Modifier	Method	Description	
and Type			
void	<code>terminal ()</code>	First print game board, current round and player; If he is in jail, refer to the document Otherwise, prompt the user to roll the dice(user needs to input something); Then player should move forward; then call the execute method of the arriving square.	

Methods inherited from class java.lang.Object

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

Constructor Detail

Game
<pre>public Game ()</pre>

Method Detail

terminal

```
public void terminal()
```

First print game board, current round and player; If he is in jail, refer to the document Otherwise, prompt the user to roll the dice(user needs to input something); Then player should move forward; then call the execute method of the arriving square. Note that user can save, load or quit at this time. At the beginning of each round, check if the game ends. If so, print the game result and switch to the Endgame Interface

Specified by:

```
terminal in interface IBase
```

Package [hk.edu.polyu.comp.comp3211.monopoly.controller](#)

Class StartMenu

java.lang.Object
 hk.edu.polyu.comp.comp3211.monopoly.controller.StartMenu

All Implemented Interfaces:

[IBase](#)

```
public class StartMenu
extends java.lang.Object
implements IBase
```

Constructor Summary

Constructors	
Constructor	Description
StartMenu()	

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
void	terminal()	Welcome the user.

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

StartMenu
<pre>public StartMenu()</pre>

Method Detail

terminal

```
public void terminal()
```

Welcome the user. If there exists a save file in the directory, prompt to the user of the option to load the game. User can try to start a game; load a game if any; and exit in this interface.

Specified by:

```
terminal in interface IBase
```

Package [hk.edu.polyu.comp.comp3211.monopoly.model](#)

Class Board

java.lang.Object
hk.edu.polyu.comp.comp3211.monopoly.model.Board

```
public class Board
extends java.lang.Object
```

A board, containing players, squares, and game status

Constructor Summary

Constructors	
Constructor	Description
Board (int num)	Initialize the board with fixed number of squares and customized number of players (2-6)

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
int	getP_index ()	Get the index of current active player
Player []	getPlayers ()	Get all players in the board
int	getRound ()	Get the index of current round
ISquare []	getSquares ()	Get all squares in the board
void	load (java.lang.String name)	Load the board from a local file
void	save (java.lang.String name)	Save the board to a local file
void	setP_index (int p_index)	Set active player index to a custom number
void	setRound (int round)	Set round index to a custom number

Methods inherited from class java.lang.Object

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

Constructor Detail

Board

public Board(int num)

Initialize the board with fixed number of squares and customized number of players (2-6)

Parameters:

num - number of players in the board

Method Detail

getPlayers

public Player [] getPlayers()

Get all players in the board

Returns:

array of players

getSquares

public ISquare [] getSquares()

Get all squares in the board

Returns:

array of squares

getRound

public int getRound()

Get the index of current round

Returns:

round index

setRound

```
public void setRound(int round)
```

Set round index to a custom number

Parameters:

`round` - dest round index

getP_index

```
public int getP_index()
```

Get the index of current active player

Returns:

round index

setP_index

```
public void setP_index(int p_index)
```

Set active player index to a custom number

Parameters:

`p_index` - dest player index

save

```
public void save(java.lang.String name)
```

Save the board to a local file

Parameters:

`name` - the path (name) of the local file

load

```
public void load(java.lang.String name)
```

Load the board from a local file

Parameters:

`name` - the path (name) of the local file

Package [hk.edu.polyu.comp.comp3211.monopoly.model](#)

Class Player

java.lang.Object
hk.edu.polyu.comp.comp3211.monopoly.model.Player

```
public class Player
extends java.lang.Object
```

A player and its status in the game

Constructor Summary

Constructors	
Constructor	Description
<code>Player ()</code>	initialize a player and scan input from user

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
void	<code>bankrupt ()</code>	The player is bankrupt
int	<code>getInJail ()</code>	Get the "IN JAIL" status of the player
int	<code>getMoney ()</code>	Get the balance of the player
java.lang.String	<code>getName ()</code>	Get the name of the player
int	<code>getPosition ()</code>	Get the position of the player
void	<code>goToJail ()</code>	The player goes to jail
boolean	<code>isBankrupted ()</code>	Judge whether the player is bankrupted
void	<code>move(int step)</code>	Advance by the number of steps, note 20th square and 1st square is connected If player goes past the starting position, give him salary(with notice).
int	<code>rollDice ()</code>	Roll a dice which is uniformly distributed from 1-4
void	<code>setInJail(int inJail)</code>	Set the "IN JAIL" status to a custom number
void	<code>setMoney(int money)</code>	Set balance of the player to a custom number
void	<code>setName(java.lang.String name)</code>	Set name of the player to a custom string
void	<code>setPosition(int position)</code>	

Set position of the player to a custom number

Methods inherited from class java.lang.Object

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

Constructor Detail

Player

public Player()

initialize a player and scan input from user

Method Detail

getName

public java.lang.String getName()

Get the name of the player

Returns:

name of the player

setName

public void setName(java.lang.String name)

Set name of the player to a custom string

Parameters:

name - dest. name

getMoney

public int getMoney()

Get the balance of the player

Returns:
balance of the player

setMoney

```
public void setMoney(int money)
```

Set balance of the player to a custom number

Parameters:
`money` - dest. balance

getPosition

```
public int getPosition()
```

Get the position of the player

Returns:
position of the player

setPosition

```
public void setPosition(int position)
```

Set position of the player to a custom number

Parameters:
`position` - dest. position

move

```
public void move(int step)
```

Advance by the number of steps, note 20th square and 1st square is connected If player goes past the starting position, give him salary(with notice).

Parameters:
`step` - number of steps

goToJail

```
public void goToJail()
```

The player goes to jail

bankrupt

```
public void bankrupt()
```

The player is bankrupt

getInJail

```
public int getInJail()
```

Get the "IN JAIL" status of the player

Returns:
"IN JAIL" status

setInJail

```
public void setInJail(int inJail)
```

Set the "IN JAIL" status to a custom number

Parameters:
`inJail` - dest. "IN JAIL" status

isBankrupted

```
public boolean isBankrupted()
```

Judge whether the player is bankrupted

Returns:
true if bankrupted, else false

rollDice

```
public int rollDice()
```

Roll a dice which is uniformly distributed from 1-4

Returns:

the result (ranged from 1-4)

Package `hk.edu.polyu.comp.comp3211.monopoly.model.squares`

Interface ISquare

All Known Implementing Classes:

`Chance`, `Free`, `Go`, `Jail`, `Oops`, `Property`, `Tax`

`public interface` **ISquare**

Any square of the board

Method Summary

All Methods	Instance Methods	Abstract Methods
Modifier and Type	Method	Description
void	<code>execute</code> (<code>Player</code> player)	Generate an effect to a player

Method Detail

execute

```
void execute(Player player)
```

Generate an effect to a player

Parameters:

`player` - dest. player

Package `hk.edu.polyu.comp.comp3211.monopoly.model.squares`

Class Chance

`java.lang.Object`
`hk.edu.polyu.comp.comp3211.monopoly.model.squares.Chance`

All Implemented Interfaces:

`ISquare`

```
public class Chance
  extends java.lang.Object
  implements ISquare
```

The Chance square of the board

Constructor Summary

Constructors	
Constructor	Description
<code>Chance ()</code>	

Method Summary

All Methods		
Instance Methods		
Concrete Methods		
Modifier and Type	Method	Description
void	<code>execute(Player player)</code>	Generate an effect to a player

Methods inherited from class java.lang.Object

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

Constructor Detail

Chance
<pre>public Chance ()</pre>

Method Detail

execute

```
public void execute(Player player)
```

Generate an effect to a player

Chance Effect:

- The player either gains a random amount (n*10) up to 200;
- or loses a random amount (n*10) up to 300.

Specified by:

```
execute in interface ISquare
```

Parameters:

```
player - dest. player
```

Package [hk.edu.polyu.comp.comp3211.monopoly.model.squares](#)

Class Free

java.lang.Object
hk.edu.polyu.comp.comp3211.monopoly.model.squares.Free

All Implemented Interfaces:

ISquare

```
public class Free
extends java.lang.Object
implements ISquare
```

The Free-Parking square of the board

Constructor Summary

Constructors	
Constructor	Description
Free()	

Method Summary

All Methods		
Instance Methods		
Concrete Methods		
Modifier and Type	Method	Description
void	execute (Player player)	Generate an effect to a player

Methods inherited from class java.lang.Object

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

Constructor Detail

Free
<pre>public Free()</pre>

Method Detail

execute

```
public void execute(Player player)
```

Generate an effect to a player

Free-Parking Effect:

- None.

Specified by:

```
execute in interface ISquare
```

Parameters:

```
player - dest. player
```

Package [hk.edu.polyu.comp.comp3211.monopoly.model.squares](#)

Class Go

java.lang.Object
hk.edu.polyu.comp.comp3211.monopoly.model.squares.Go

All Implemented Interfaces:

ISquare

```
public class Go
extends java.lang.Object
implements ISquare
```

The Go square of the board

Constructor Summary

Constructors	
Constructor	Description
Go ()	

Method Summary

All Methods		
Instance Methods		
Concrete Methods		
Modifier and Type	Method	Description
void	execute(Player player)	Generate an effect to a player

Methods inherited from class java.lang.Object

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

Constructor Detail

```
Go

public Go ()
```

Method Detail

execute

```
public void execute(Player player)
```

Generate an effect to a player

Go Effect:

- The player passes through (not necessarily lands on):
gets \$1500 salary;
- All players starts from this square, and at this time:
this square has no effect.

Specified by:

execute in interface ISquare

Parameters:

player - dest. player

Package [hk.edu.polyu.comp.comp3211.monopoly.model.squares](#)

Class Jail

java.lang.Object
hk.edu.polyu.comp.comp3211.monopoly.model.squares.Jail

All Implemented Interfaces:

ISquare

```
public class Jail
extends java.lang.Object
implements ISquare
```

The In-Jail/Just-Visiting square of the board

Constructor Summary

Constructors	
Constructor	Description
Jail()	

Method Summary

All Methods		
Instance Methods		
Concrete Methods		
Modifier and Type	Method	Description
void	execute(Player player)	Generate an effect to a player

Methods inherited from class java.lang.Object

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

Constructor Detail

Jail
public Jail()

Method Detail

execute

```
public void execute(Player player)
```

Generate an effect to a player

In-Jail/Just-Visiting Effect:

- if the player just passes:
this square has no effect
- if the player is at in-jail status:
cannot make a move, and can only get out by:
 1. throwing doubles (two dice with same results) within next 3 turns
(and if so, immediately moves by the throw);
 2. paying \$150 before rolling the dice within the next 2 turns;
 3. if still not out within 3 turns, paying \$150 (must) to get out
(and if `cond. 2` or `cond. 3` is met, the player can throw once and move).

Specified by:

```
execute in interface ISquare
```

Parameters:

`player` - dest. player

Package [hk.edu.polyu.comp.comp3211.monopoly.model.squares](#)

Class Oops

java.lang.Object
hk.edu.polyu.comp.comp3211.monopoly.model.squares.Oops

All Implemented Interfaces:

ISquare

```
public class Oops
extends java.lang.Object
implements ISquare
```

The Go-to-Jail square of the board

Constructor Summary

Constructors	
Constructor	Description
<code>Oops ()</code>	

Method Summary

All Methods		
Instance Methods		
Concrete Methods		
Modifier and Type	Method	Description
void	<code>execute(Player player)</code>	Generate an effect to a player

Methods inherited from class java.lang.Object		
<code>clone, equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>		

Constructor Detail

Oops	
<pre>public Oops ()</pre>	

Method Detail

execute

```
public void execute(Player player)
```

Generate an effect to a player

Go-to-Jail Effect:

- The player immediately goes to Jail.

Specified by:

```
execute in interface ISquare
```

Parameters:

```
player - dest. player
```

Package `hk.edu.polyu.comp.comp3211.monopoly.model.squares`

Class Property

`java.lang.Object`
`hk.edu.polyu.comp.comp3211.monopoly.model.squares.Property`

All Implemented Interfaces:

`ISquare`

```
public class Property
extends java.lang.Object
implements ISquare
```

The Property squares of the board

Constructor Summary

Constructors	
Constructor	Description
<code>Property(java.lang.String s, int p, int r)</code>	Initialize a property with params:

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
<code>void</code>	<code>execute(Player player)</code>	Generate an effect to a player
<code>java.lang.String</code>	<code>getName()</code>	Get the name of the property
<code>Player</code>	<code>getOwner()</code>	Get the owner of the property
<code>void</code>	<code>setOwner(Player owner)</code>	Set the owner to a custom player

Methods inherited from class java.lang.Object

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

Constructor Detail

Property

```
public Property(java.lang.String s,  
               int p,  
               int r)
```

Initialize a property with params:

Parameters:

- s - name of the property
- p - selling price of the property
- r - rental price of the property

Method Detail

execute

```
public void execute(Player player)
```

Generate an effect to a player

Property Effect:

- if property is not owned by any player:
the visiting player can buy the property at the selling price or do nothing;
- if property is owned by a player:
then if visiting player is the owner, nothing happens, otherwise the visiting player must pay the owner a rent.

Specified by:

```
execute in interface ISquare
```

Parameters:

player - dest. player

getName

```
public java.lang.String getName()
```

Get the name of the property

Returns:

name of the property

getOwner

public **Player** getOwner()

Get the owner of the property

Returns:

owner of the property

setOwner

public void setOwner(**Player** owner)

Set the owner to a custom player

Parameters:

owner - dest. player

Package [hk.edu.polyu.comp.comp3211.monopoly.model.squares](#)

Class Tax

java.lang.Object
hk.edu.polyu.comp.comp3211.monopoly.model.squares.Tax

All Implemented Interfaces:

ISquare

```
public class Tax
extends java.lang.Object
implements ISquare
```

The Income-Tax square of the board

Constructor Summary

Constructors	
Constructor	Description
<code>Tax ()</code>	

Method Summary

All Methods		
Instance Methods		
Concrete Methods		
Modifier and Type	Method	Description
void	<code>execute(Player player)</code>	Generate an effect to a player

Methods inherited from class java.lang.Object

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

Constructor Detail

Tax
<code>public Tax ()</code>

Method Detail

execute

```
public void execute(Player player)
```

Generate an effect to a player

Income-Tax Effect:

- the player pays 10% of the balance (in form 10*n) as tax.

Specified by:

```
execute in interface ISquare
```

Parameters:

```
player - dest. player
```


Package [hk.edu.polyu.comp.comp3211.monopoly.view](#)

Class Printer

java.lang.Object
hk.edu.polyu.comp.comp3211.monopoly.view.Printer

```
public class Printer
extends java.lang.Object
```

Constructor Summary

Constructors	
Constructor	Description
Printer (Board board)	

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method	Description
void	Print ()	Print out current game board

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

Printer
<pre>public Printer(Board board)</pre>

Method Detail

Print

```
public void Print()
```

Print out current game board