

Table of Contents

Model Detail	2
Class Model	2
Client	2
AuctionClient	3
WaitAndBid	6
IAuctionListener	7
Server	8
AuctionServer	8
AuctionServerImpl	9
WrappedObserver	12
AuctionServerImplFactory	13
Item	14
IAuctionServer	16

Model Documentation

Model Detail

This document provides a complete overview of all element details. For simpler and more focused reports, simply copy this initial template and turn off the sections not required.

Class Model

<i>Type:</i>	<u>Package</u>
<i>Status:</i>	Proposed. Version . Phase 1.0.
<i>Package:</i>	Model
<i>Detail:</i>	<i>Created on 2012-11-26. Last modified on 2012-11-26</i>
<i>GUID:</i>	{5B413F06-2BC5-49b6-87DB-E65CCEDD854A}

Client

<i>Type:</i>	<u>Package</u>
<i>Status:</i>	Proposed. Version 1.0. Phase 1.0.
<i>Package:</i>	Class Model
<i>Detail:</i>	<i>Created on 2012-12-02. Last modified on 2012-12-02</i>
<i>GUID:</i>	{D766C12C-154B-4f44-9AE4-DBDE86A7E9D1}

Client - (Class diagram)

<i>Created By:</i>	patryk.orwat on 2012-12-02
<i>Last Modified:</i>	2012-12-02
<i>Version:</i>	1.0. <i>Locked:</i> False
<i>GUID:</i>	{11A7A802-4E72-4680-8762-95520CB471A0}

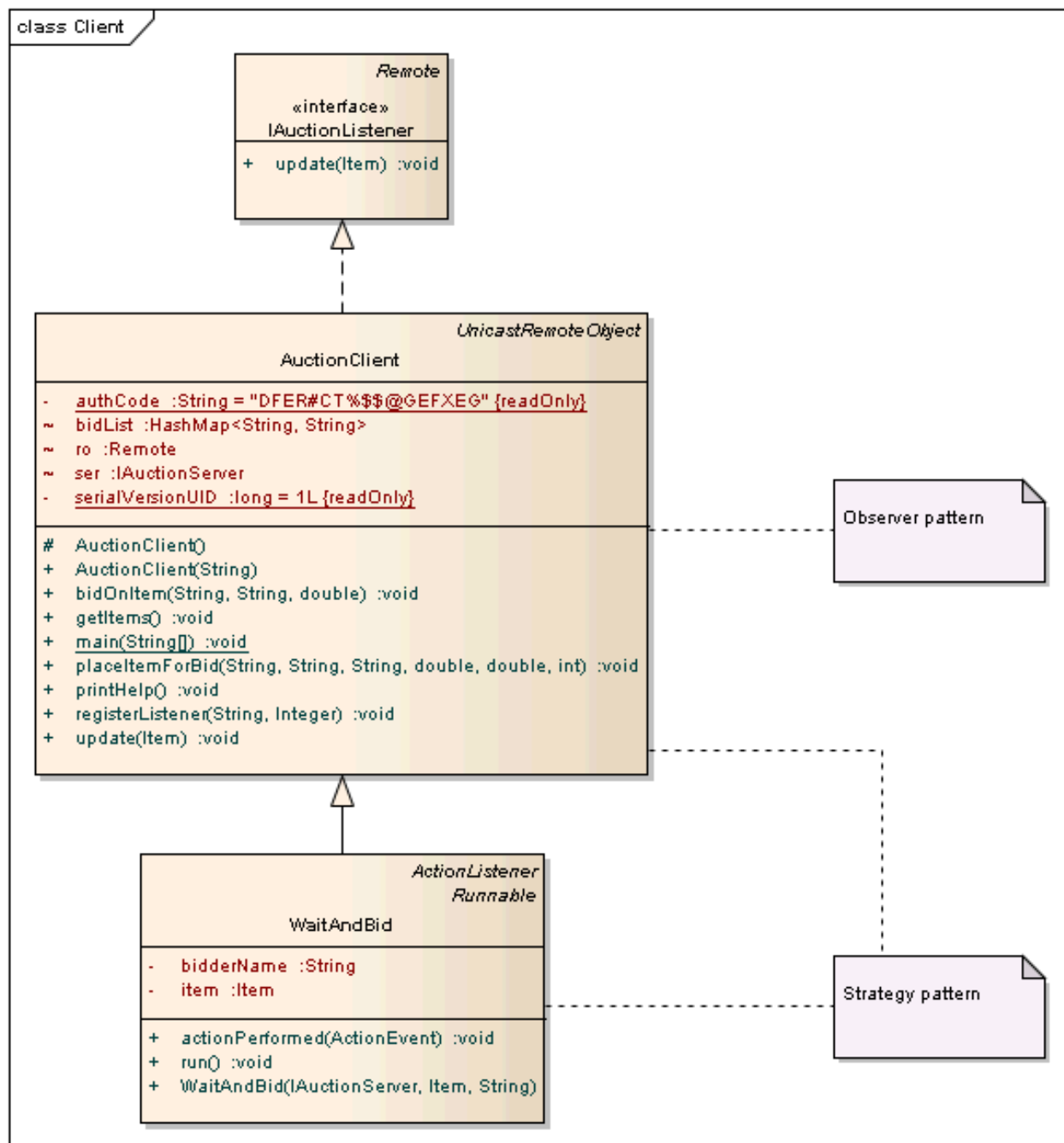


Figure: 1

AuctionClient

Type:

Class UnicastRemoteObject

Status:

Proposed. Version 1.0. Phase 1.0.

Package:

Client **Keywords:**

Detail:

Created on 2012-12-02. Last modified on 2012-12-02.

GUID:

{8738F704-D46C-4716-B69D-02ED531CEF1B}

Main client class which is responsible for user interaction and communication with server.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Generalization Source -> Destination	Public WaitAndBid	Public AuctionClient	
Association Source -> Destination	Public AuctionClient	Package ser IAuctionServer	
Realization Source -> Destination	Public AuctionClient	Public IAuctionListener	
NoteLink Source -> Destination	Public <anonymous>	Public AuctionClient	
NoteLink Source -> Destination	Public <anonymous>	Public AuctionClient	

Attributes

Attribute	Notes	Constraints and tags
authCode String Private Static Const		<i>Default:</i> "DFER#CT%\$\$@GEFXEG"
bidList HashMap<String, String> Package		<i>Default:</i>
ro Remote Package		<i>Default:</i>

Attribute	Notes	Constraints and tags
ser IAuctionServer Package		<i>Default:</i>
serialVersionUID long Private Static Const		<i>Default:</i> 1L

Operations

Method	Notes	Parameters
AuctionClient() Protected		
AuctionClient() Public	Creates client object and initializes server connection.	String [in] uri URI to server
bidOnItem() void Public	Bids an auction described by a name.	String [in] bidderName Name of the bidder String [in] itemName Name of the item double [in] bid Amount of money
getItems() void Public	Gets list of items from server and prints it on a screen.	
Static main() void Public	The main client application.	String[] [in] args System arguments. The first should be the server URI.
placeItemForBid() void Public	Adds a new auction.	String [in] ownerName Owner name String [in] itemName Item name String [in] itemDesc Item description double [in] startBid Start bid double [in] maxBid Maximum bid int [in] auctionTime Auction time
printHelp() void	Prints available commands which client can	

Method	Notes	Parameters
Public	run.	
registerListener() void Public	Registers client to observe an item. If the item is changed, update method if called.	String [in] itemName Name of the item Integer [in] strategy Strategy of automatic bidding
update() void Public	Update method which is called on server side when an item is changed. Shows on client side info about update and fires proper strategies.	Item [in] item The item which was changed

WaitAndBid

Type: **Class** **AuctionClient, ActionListener, Runnable**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Client *Keywords:*
Detail: Created on 2012-12-02. Last modified on 2012-12-02.
GUID: {714A4DB2-572A-4490-A4E2-0B7EBCE964D2}

Class responsible for creating bidding thread for WaitAndBid strategy

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public WaitAndBid	Private item Item	
Generalization Source -> Destination	Public WaitAndBid	Public AuctionClient	
NoteLink Source -> Destination	Public <anonymous>	Public WaitAndBid	

Attributes

Attribute	Notes	Constraints and tags
bidderName String Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
item Item Private		<i>Default:</i>

Operations

Method	Notes	Parameters
actionPerformed() void Public	Perform bid	ActionEvent [in] e e
run() void Public	Creates timer in new thread	
WaitAndBid() Public	Constructor of the class	IAuctionServer [in] s Item [in] item Item to bid String [in] bidderName Author of the bid

IAuctionListener

Type: **Interface** **Remote**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Client *Keywords:*
Detail: Created on 2012-12-02. Last modified on 2012-12-02.
GUID: {62EF2AF4-C10C-46a3-A6A9-1750DEE80891}

Interface implemented by AuctionClient to provide callbacks from server side. Required by RMI.

Connections

Connector	Source	Target	Notes
Realization Source -> Destination	Public AuctionClient	Public IAuctionListener	
Association Source -> Destination	Public WrappedObserver	Private ro IAuctionListener	

Operations

Method	Notes	Parameters
update() void Public	Callback function which is called on the server side to inform the client that an Item (item	Item [in] item item to be observable.

Method	Notes	Parameters
	param) has changed.	

Server

Type: **Package**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model
Detail: Created on 2012-12-02. Last modified on 2012-12-02
GUID: {D5D7151C-C648-42a4-9F30-9B4291F28E26}

Server - (Class diagram)

Created By: patryk.orwat on 2012-12-02
Last Modified: 2012-12-02
Version: 1.0. **Locked:** False
GUID: {87B431E0-D9AF-4071-96A3-10185F6AC7BA}

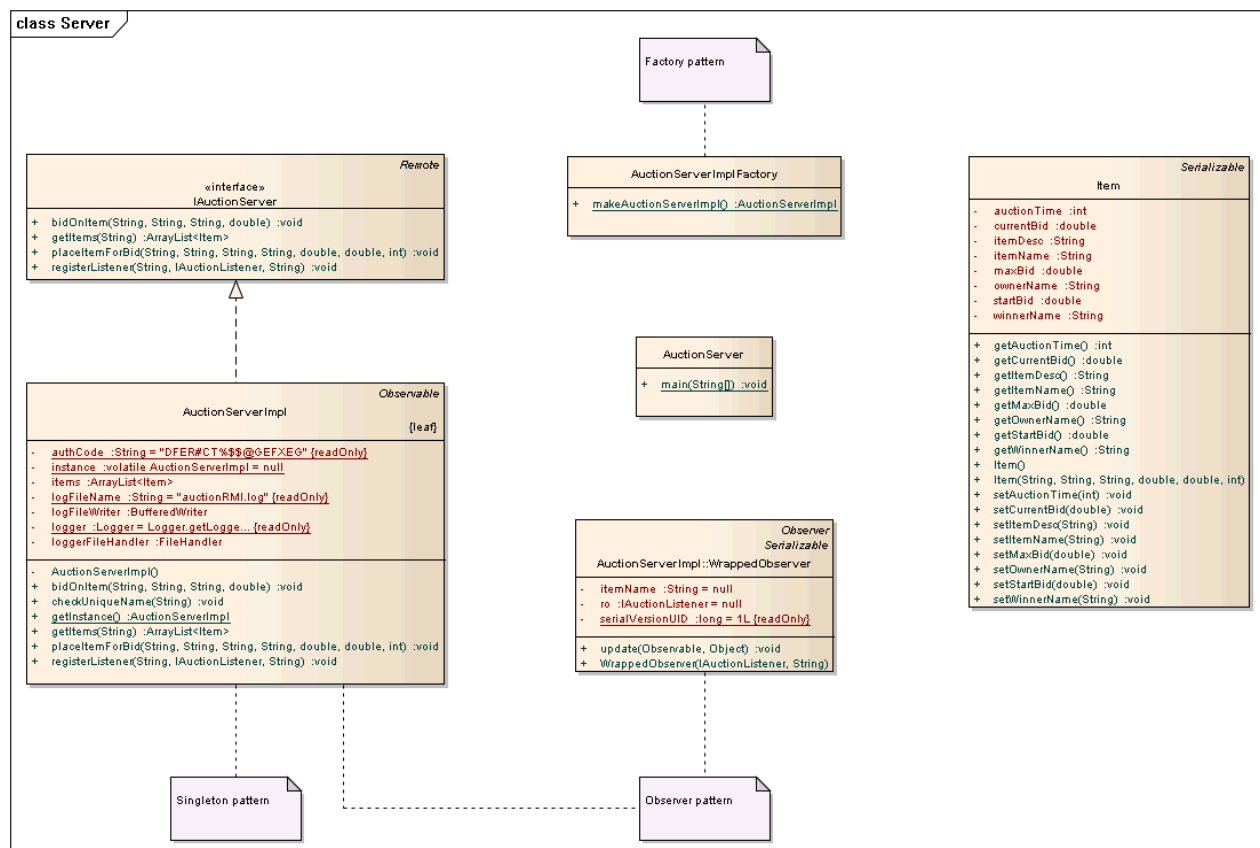


Figure: 2

AuctionServer

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Server **Keywords:**
Detail: Created on 2012-12-02. Last modified on 2012-12-02.
GUID: {842EB8D8-AFE5-4695-A893-38703FDB91C2}

Main class responsible for running the server.

Custom Properties

- isActive = False

Operations

Method	Notes	Parameters
Static main() void Public	Gets AuctionServerImpl object, makes a stub and registers it.	String[] [in] args system arguments (not used)

AuctionServerImpl

Type: **Class** **Observable**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Server **Keywords:**
Detail: Created on 2012-12-02. Last modified on 2012-12-02.
GUID: {60AC0BF5-0592-43d9-9CEA-CF250C890E42}

Implements server methods responsible for auction system.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Realization Source -> Destination	Public AuctionServerImpl	Public IAuctionServer	
NoteLink Source -> Destination	Public <anonymous>	Public AuctionServerImpl	
NoteLink Source -> Destination	Public <anonymous>	Public AuctionServerImpl	

Attributes

Attribute	Notes	Constraints and tags
-----------	-------	----------------------

Attribute	Notes	Constraints and tags
authCode String Private Static Const		<i>Default:</i> "DFER#CT%\$\$@GEFXEG"
instance AuctionServerImpl Private Static		<i>Default:</i> null
items ArrayList<Item> Private		<i>Default:</i>
logFileName String Private Static Const		<i>Default:</i> "auctionRMI.log"
logFileWriter BufferedWriter Private		<i>Default:</i>
logger Logger Private Static Const		<i>Default:</i> Logger.getLogger("AuctionRMI")
loggerFileHandler FileHandler Private		<i>Default:</i>

Attribute	Notes	Constraints and tags

Operations

Method	Notes	Parameters
AuctionServerImpl() Private	Private constructor of the class to prevent the creation of more than one object (singleton pattern).	
bidOnItem() void Public	Bids one Item. If an Item is changed, observers are notified.	String [in] authCode authorisation code String [in] bidderName name of the bidder String [in] itemName name of the item double [in] bid value of the bid
checkUniqueName() void Public	Checks if there is an item with the same itemName. If so, a RemoteException is thrown.	String [in] name the name to be check
Static getInstance() AuctionServerImpl Public	Singleton pattern implementation. source: http://en.wikipedia.org/wiki/Singleton_pattern "should not be used prior to J2SE 5.0" @return AuctionServerImpl the only object in the application	
getItems() ArrayList<Item> Public	Returns a list of items. @return ArrayList<Item> list of items	String [in] authCode authorisation code
placeItemForBid() void Public	Creates a new Item object and adds to the list of items.	String [in] authCode authorisation code String [in] ownerName owner name String [in] itemName item name String [in] itemDesc item description double [in] startBid start bid double [in] maxBid max bid int [in] auctionTime auction time
registerListener() void Public	Register a client to observe an Item (observer pattern).	String [in] authCode authorisation code IAuctionListener [in] al client object String [in] itemName

Method	Notes	Parameters
		Item name

WrappedObserver

Type: **Class** **Observer, Serializable**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Server **Keywords:**
Detail: Created on 2012-12-02. Last modified on 2012-12-02.
GUID: {31D50C4D-6FBC-4f9a-AAB4-15048460B0BE}

Represents an observer (client).

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public WrappedObserver	Private ro IAuctionListener	
NoteLink Source -> Destination	Public <anonymous>	Public WrappedObserver	

Attributes

Attribute	Notes	Constraints and tags
itemName String Private		<i>Default:</i> null
ro IAuctionListener Private		<i>Default:</i> null
serialVersionUID long Private Static Const		<i>Default:</i> 1L

Attribute	Notes	Constraints and tags

Operations

Method	Notes	Parameters
update() void Public		Observable [in] o Object [in] arg
WrappedObserver() Public		IAuctionListener [in] ro String [in] name

AuctionServerImplFactory

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Server *Keywords:*
Detail: Created on 2012-12-02. Last modified on 2012-12-02.
GUID: {3650FE13-A163-4edb-9AE8-1A61BC4066B4}

AuctionServerImpl's Factory.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
NoteLink Source -> Destination	Public <anonymous>	Public AuctionServerImplFact ory	

Operations

Method	Notes	Parameters
Static makeAuctionServerImpl() AuctionServerImpl Public	Returns an AuctionServerImpl object. AuctionServerImpl class uses singleton pattern and always the same object is returned. @return AuctionServerImpl	

Item

Type: **Class** **Serializable**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Server **Keywords:**
Detail: Created on 2012-12-02. Last modified on 2012-12-02.
GUID: {8EE5F479-6AD7-4ee1-9C83-FE801BF2D880}

Represents an auction item.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public WaitAndBid	Private item Item	

Attributes

Attribute	Notes	Constraints and tags
auctionTime int Private		<i>Default:</i>
currentBid double Private		<i>Default:</i>
itemDesc String Private		<i>Default:</i>
itemName String Private		<i>Default:</i>

Attribute	Notes	Constraints and tags
maxBid double Private		<i>Default:</i>
ownerName String Private		<i>Default:</i>
startBid double Private		<i>Default:</i>
winnerName String Private		<i>Default:</i>

Operations

Method	Notes	Parameters
getAuctionTime() int Public		
getCurrentBid() double Public		
getItemDesc() String Public		
getItemName() String Public		
getMaxBid() double Public		
getOwnerName() String Public		
getStartBid() double		

Method	Notes	Parameters
Public		
getWinnerName() String Public		
Item() Public		
Item() Public		String [in] ownerName String [in] itemName String [in] itemDesc double [in] startBid double [in] maxBid int [in] auctionTime
setAuctionTime() void Public		int [in] val
setCurrentBid() void Public		double [in] val
setItemDesc() void Public		String [in] val
setItemName() void Public		String [in] val
setMaxBid() void Public		double [in] val
setOwnerName() void Public		String [in] val
setStartBid() void Public		double [in] val
setWinnerName() void Public		String [in] val

IAuctionServer

Type: **Interface** **Remote**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Server *Keywords:*
Detail: Created on 2012-12-02. Last modified on 2012-12-02.
GUID: {AD4E9B6C-A989-4ace-B638-33C1CD5988A2}

Describes methods provided by AuctionServerImpl. Necessary for RMI.

Connections

Connector	Source	Target	Notes
Association Source -> Destination	Public AuctionClient	Package ser IAuctionServer	
Realization Source -> Destination	Public AuctionServerImpl	Public IAuctionServer	

Operations

Method	Notes	Parameters
bidOnItem() void Public	Bids one Item. If an Item is changed, observers are notified.	String [in] authCode authorisation code String [in] bidderName name of the bidder String [in] itemName name of the item double [in] bid value of the bid
getItems() ArrayList<Item> Public	Returns a list of items. @return ArrayList<Item> list of items	String [in] authCode authorisation code
placeItemForBid() void Public	Creates a new Item object and adds to the list of items.	String [in] authCode authorisation code String [in] ownerName owner name String [in] itemName item name String [in] itemDesc item description double [in] startBid start bid double [in] maxBid max bid int [in] auctionTime auction time
registerListener() void Public	Register a client to observe an Item (observer pattern).	String [in] authCode authorisation code IAuctionListener [in] al client object String [in] itemName Item name