Table of Contents

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

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

Client

Type:	**Package**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Class Model
Detail:	*Created on 2012-11-26. Last modified on 2012-11-26*
GUID:	{B4CD24C8-984F-44bb-8FBE-B225A3D9B2FF}

Client - *(Class diagram)*

Created By:	Jacek *on* 2012-11-26
Last Modified:	2012-11-26
Version:	1.0. *Locked:* False
GUID:	{3483899E-BB8B-4559-979A-4BCCF2B3D5A0}
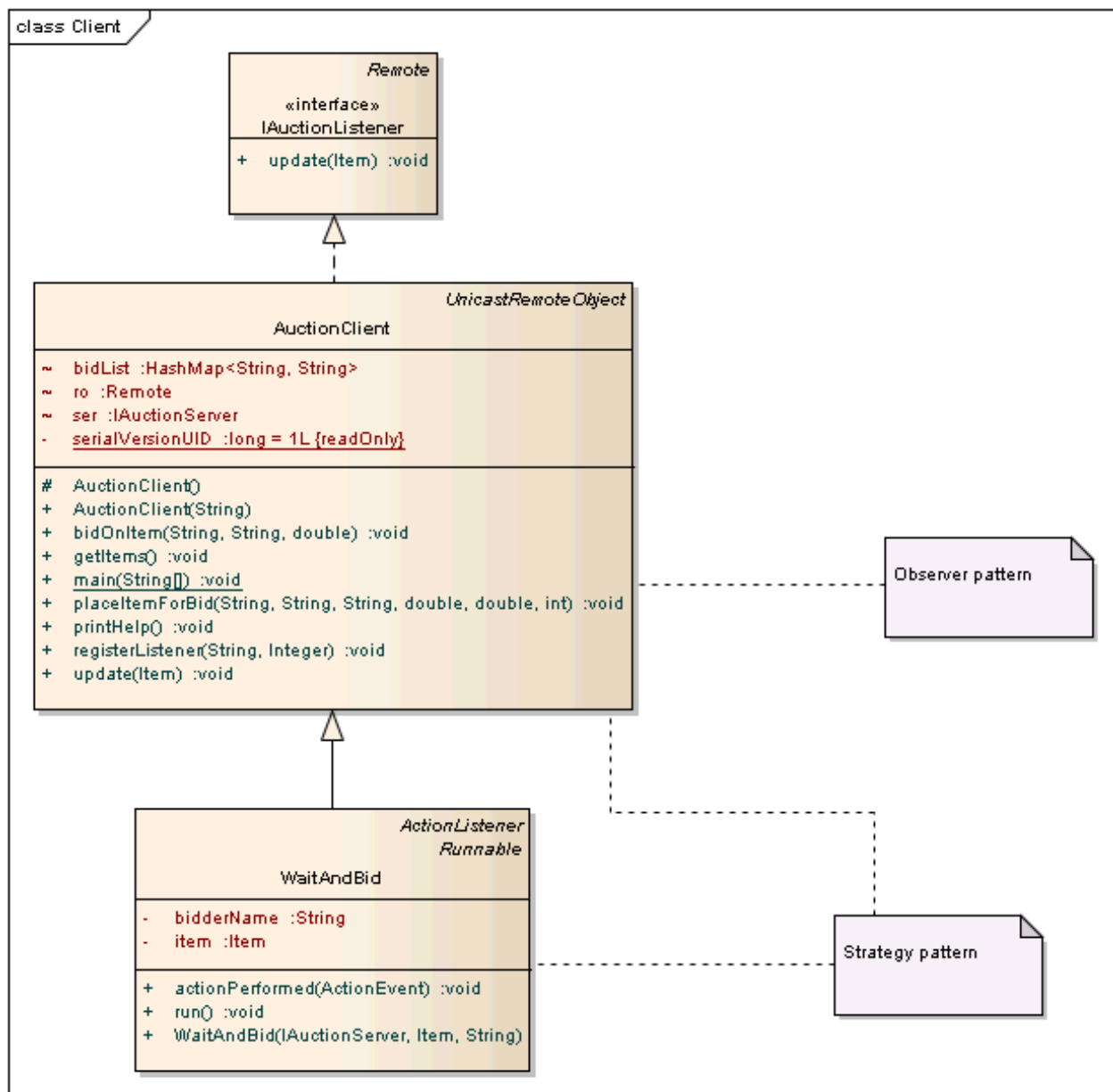
Figure: 1

AuctionClient

Type:	**Class UnicastRemoteObject**
Status:	Proposed. Version 1.0. Phase 1.0.
Package:	Client *Keywords:*
Detail:	*Created on 2012-11-26. Last modified on 2012-11-26.*
GUID:	{2D4A950A-0307-483a-9F26-0DC19FAEFB9A}

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

Custom Properties

Custom Properties

- isActive = False

Connections

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

Attributes

Attribute	Notes	Constraints and tags
bidList HashMap<String, String> Package		*Default:*
ro Remote Package		*Default:*
ser IAuctionServer Package		*Default:*
serialVersionUID long Private Static Const		*Default:* 1L

Attribute	Notes	Constraints and tags

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 Public	Prints available commands which client can 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 2012-11-26. Last modified on 2012-11-26.*
GUID:	{DB3A1431-4789-4909-8B2F-2D6E7BB3DA36}

Class responisible for creating biding 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 WaitAndBid	Public <anonymous>	

Attributes

Attribute	Notes	Constraints and tags
bidderName String Private		*Default:*
item Item Private		*Default:*

Operations

Method	Notes	Parameters
actionPerformed() void	Perform bid	**ActionEvent** [in] e

Method	Notes	Parameters
Public		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 2012-11-26. Last modified on 2012-11-26.
GUID: {2D7E7DEB-5ABE-4bd3-B6FA-FB674C302A79}

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 param) has changed.	**Item** [in] item item to be observable.

Server

Type: **Package**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Class Model
Detail: Created on 2012-11-26. Last modified on 2012-11-26
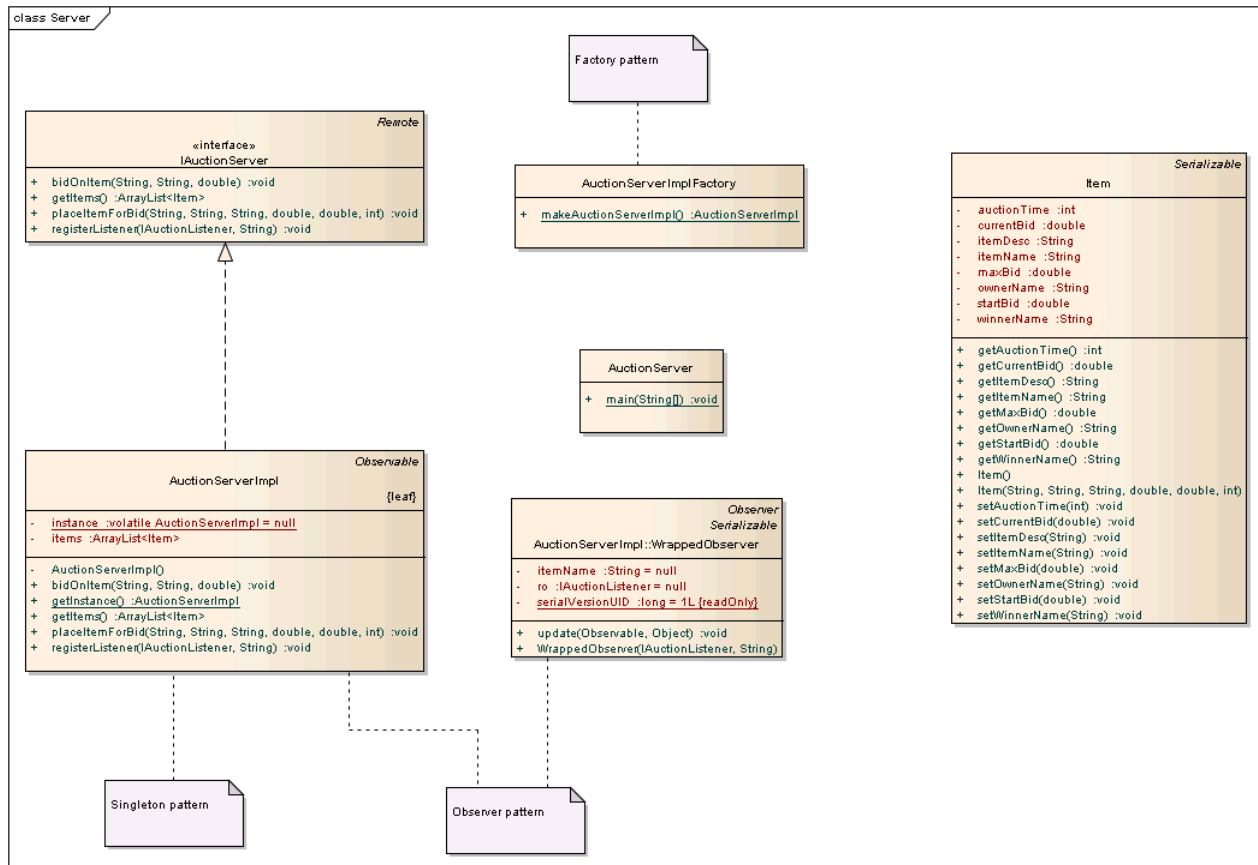*GUID:*   {EA6174B6-95CE-47c0-A6ED-4FB090A3A40C}

Server - *(Class diagram)*
Created By: Jacek *on* 2012-11-26
Last Modified: 2012-11-26
Version: 1.0. *Locked:* False
GUID: {47120561-CD20-4f86-B054-0095CADD7964}

Figure: 2

AuctionServer

Type: **Class**
Status: Proposed. Version 1.0. Phase 1.0.
Package: Server   *Keywords:*
Detail: *Created on 2012-11-26. Last modified on 2012-11-26.*
GUID: {437DB21A-C046-4d56-8A92-75E9F4806664}

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 2012-11-26. Last modified on 2012-11-26.*
GUID:	{7E9115A1-65C5-4d5d-9B6F-C205EBD75094}

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 AuctionServerImpl	Public <anonymous>	

Attributes

Attribute	Notes	Constraints and tags
instance AuctionServerImpl Private Static		*Default:* null

Attribute	Notes	Constraints and tags
items ArrayList<Item> Private		*Default:*

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 noitfied.	**String** [in] bidderName name of the bidder **String** [in] itemName name of the item **double** [in] bid value of the bid
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	
placeItemForBid() void Public	Creates a new Item object and adds to the list of items.	**String** [in] ownerName owner name **String** [in] itemName item name **String** [in] itemDesc iten description **double** [in] startBid start bid **double** [in] maxBid max bid **int** [in] auctionTime aution time
registerListener() void Public	Register a client to observe an Item (observer pattern).	**IAuctionListener** [in] al client object **String** [in] itemName Item name

WrappedObserver

Type:	**Class Observer, Serializable**	
Status:	Proposed. Version 1.0. Phase 1.0.	
Package:	Server *Keywords:*	
Detail:	*Created on 2012-11-26. Last modified on 2012-11-26.*	
GUID:	{818A44D1-2D16-473f-A6F5-F22756177FFE}	

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 WrappedObserver	Public <anonymous>	

Attributes

Attribute	Notes	Constraints and tags
itemName String Private		*Default:* null
ro IAuctionListener Private		*Default:* null
serialVersionUID long Private Static Const		*Default:* 1L

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 2012-11-26. Last modified on 2012-11-26.
GUID:	{82E5B19A-2A47-44a4-8165-334EF38961DC}

AuctionServerImpl's Factory.

Custom Properties

- isActive = False

Connections

Connector	Source	Target	Notes
NoteLink Source -> Destination	Public <anonymous>	Public AuctionServerImplFactory	

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 2012-11-26. Last modified on 2012-11-26.*
GUID: {8C95F9C9-542F-4e03-8E95-D891AC040B9B}

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		*Default:*
currentBid double Private		*Default:*
itemDesc String Private		*Default:*
itemName String Private		*Default:*
maxBid double		*Default:*

Attribute	Notes	Constraints and tags
Private		
ownerName String Private		*Default:*
startBid double Private		*Default:*
winnerName String Private		*Default:*

Operations

Method	Notes	Parameters
getAuctionTime() int Public		
getCurrentBid() double Public		
getItemDesc() String Public		
getItemName() String Public		
getMaxBid() double Public		
getOwnerName() String Public		
getStartBid() double Public		
getWinnerName() String Public		
Item() Public		
Item()		**String** [in] ownerName

Method	Notes	Parameters
Public		**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 2012-11-26. Last modified on 2012-11-26.*
GUID: {31691CE0-704A-4780-9FEA-8958A28DF660}

Describes methods provided by AuctionServerImpl. Necessary for RMI.

Connections

Connector	Source	Target	Notes
Association	Public	Package ser	

Connector	Source	Target	Notes
Source -> Destination	AuctionClient	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] 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	
placeItemForBid() void Public	Creates a new Item object and adds to the list of items.	**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).	**IAuctionListener** [in] al client object **String** [in] itemName Item name