OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

dataBaseC

Class Ticket

java.lang.Object
    dataBaseC.Ticket

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

Data structure to hold the ticket and the data associated with it. Used by waiter and chef.

Author:

cms549

Field Summary

Fields

Modifier and Type	Field and Description
`int`	**`amountOfDishes`**
`int`	**`amountOfDishesFinished`**
`int`	**`amountOfDishesStarted`**
`int`	**`amountOfDishesUnstarted`**
`boolean`	**`hotFood`** Used to keep track of hot food tables for waiters
`java.util.ArrayList<Dish>`	**`listOfDishes`** List of dishes on the ticket
`double`	**`price`** total price of ticket
`boolean`	**`priority`** Used to mark the ticket as a priority ticket to the chef
`boolean`	**`recentlySat`** Used to keep track of recently sat tables for waiters
`char`	**`status`** status of ticket: u=unstarted, s=semi started, S=started, f=finished
`int`	**`tableNumber`** table number this order belongs to
`long`	**`ticketNumber`** unique id of ticket
`long`	**`waiterID`** id of waiter this ticket belongs to
`java.lang.String`	**`waiterName`** name of waiter this ticket is under

Constructor Summary

Constructors

Constructor and Description
`Ticket()` Creates empty ticket
`Ticket(java.lang.String waiterName, int tableNum, long waiterID)` Creates a new empty ticket with the following

Method Summary

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

Modifier and Type	Method and Description
`void`	**`addDishToTicket`**`(`**`Dish`**` d)` Adds dish to ticket and also updates price and status of ticket
`static` **`Ticket`**	**`fromString`**`(java.lang.String tick)` Takes the string representation of this ticket and make a new ticket object for it
`boolean`	**`removeDishFromTicket`**`(int indexOfDishInTickList)` Removes the dish at index i from the ticket and decrements the price
`java.lang.String`	**`toStringForChef`**`()` Creates a string representation of this ticket that Chef will use to see dishes.
`java.lang.String`	**`toStringForDBC`**`()` Creates a string representation of this ticket that DB C will use to record expenses.
`char`	**`updateStatusOfTicket`**`()` Looks through the dishes of the ticket and updates the status of the ticket accordingly

Methods inherited from class java.lang.Object

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

Field Detail

tableNumber

`public int tableNumber`

table number this order belongs to

waiterName

`public java.lang.String waiterName`

name of waiter this ticket is under

waiterID

`public long waiterID`

id of waiter this ticket belongs to

status

`public char status`

status of ticket: u=unstarted, s=semi started, S=started, f=finished

listOfDishes

`public java.util.ArrayList<Dish> listOfDishes`

List of dishes on the ticket

ticketNumber

`public long ticketNumber`

unique id of ticket

price

`public double price`

total price of ticket

amountOfDishesUnstarted

```
public int amountOfDishesUnstarted
```

amountOfDishesStarted

```
public int amountOfDishesStarted
```

amountOfDishesFinished

```
public int amountOfDishesFinished
```

amountOfDishes

```
public int amountOfDishes
```

recentlySat

```
public boolean recentlySat
```

Used to keep track of recently sat tables for waiters

hotFood

```
public boolean hotFood
```

Used to keep track of hot food tables for waiters

priority

```
public boolean priority
```

Used to mark the ticket as a priority ticket to the chef

Constructor Detail

Ticket

```
public Ticket(java.lang.String waiterName,
              int tableNum,
              long waiterID)
```

Creates a new empty ticket with the following

Parameters:

`waiterName` - - name of waiter for this ticket

`tableNum` - - table number the ticket is under

`waiterID` - - id of the waiter

Ticket

```
public Ticket()
```

Creates empty ticket

Method Detail

addDishToTicket

```
public void addDishToTicket(Dish d)
```

Adds dish to ticket and also updates price and status of ticket

Parameters:

`d -`

removeDishFromTicket

```
public boolean removeDishFromTicket(int indexOfDishInTickList)
```

Removes the dish at index i from the ticket and decrements the price

Parameters:

`indexOfDishInTickList -`

Returns:

updateStatusOfTicket

```
public char updateStatusOfTicket()
```

Looks through the dishes of the ticket and updates the status of the ticket accordingly

Returns:

`old status of ticket`

toStringForDBC

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

Creates a string representation of this ticket that DB C will use to record expenses. Includes the waiter name and id, the table number, and the price of the ticket.

Returns:

`the string format of the ticket Format: Waiter Name:John Waiter ID:123 Table Number:14 Price:$5.00`

toStringForChef

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

Creates a string representation of this ticket that Chef will use to see dishes. Includes the priority waiter's name, id, the table number, and the list of dishes MARKS DISHES THAT ARE GETTING SENT AS SENT

Returns:

```
the string format of the ticket Format is as follows:
P:WAITERNAME:WAITERID:TABLENUMBER:;DISHNAME1-,COMMENT11,COMMENT12;DISHNAME2-,COMMENT21;DISHNAME3-,COMMENT31,COMMENT32,COMMENT33
N:WAITERNAME:WAITERID:TABLENUMBER:;DISHNAME1-,COMMENT11,COMMENT12;DISHNAME2-,COMMENT21;DISHNAME3-,COMMENT31,COMMENT32,COMMENT33
```

fromString

```
public static Ticket fromString(java.lang.String tick)
```

Takes the string representation of this ticket and make a new ticket object for it

Returns:

```
a new Ticket Format is as follows:
P:WAITERNAME:WAITERID:TABLENUMBER:;DISHNAME1-,COMMENT11,COMMENT12;DISHNAME2-,COMMENT21;DISHNAME3-,COMMENT31,COMMENT32,COMMENT33
N:WAITERNAME:WAITERID:TABLENUMBER:;DISHNAME1-,COMMENT11,COMMENT12;DISHNAME2-,COMMENT21;DISHNAME3-,COMMENT31,COMMENT32,COMMENT33
```

OVERVIEW PACKAGE CLASS USE TREE DEPRECATED INDEX HELP

PREV CLASS NEXT CLASS FRAMES NO FRAMES ALL CLASSES

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD