Department of Computer Technology and Information Systems

CTIS221 – Object Oriented Programming

FALL 2019 - 2020

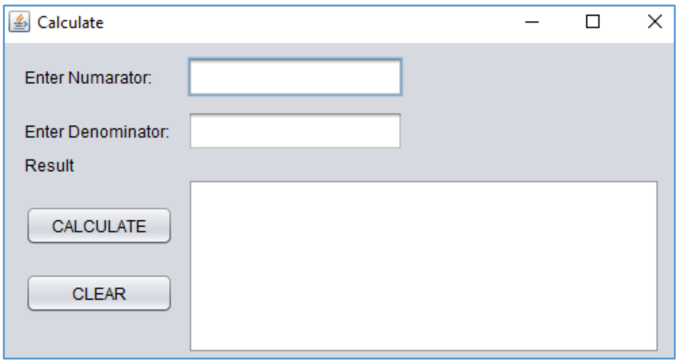
Lab Guide 14 - Week 10-2

OBJECTIVE: Exception + TextFiles + GUI

Instructor : Burcu LİMAN
Assistant : Burcu Alper, Leyla SEZER

Q1. Create a class for calculation and write a static “**calculate**” method that divides numerator to denominator and returns the result.

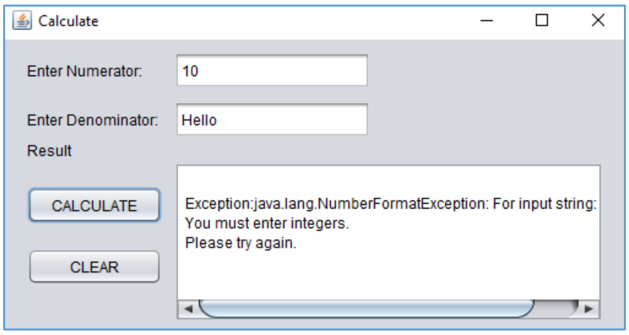
- Create a **Calculate** frame as follows:



- There are 3 labels, 2 text fields, 2 buttons and 1 text area;
  - If user fills the fields and clicks on calculate button, calculate the result of the operation by calling the **calculate ()** method.
  - If the user makes a mistake, the program catches and handles the exception.
  - If the user enters a zero denominator, an ArithmeticException occurs.
  - If the user enters a string as a denominator, NumberFormatException occurs.
  - If user clicks on “Clear” button all the fields should be cleared.
- For each exception, the user should be informed about the mistake and asked to try again as in the following figures:

  a) *Write try, catch in action of Calculate button.*

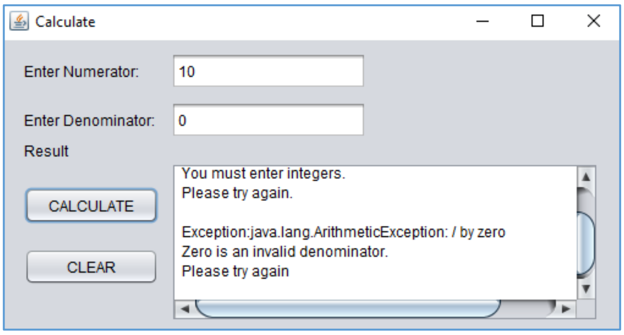
  b) *Write try, catch inside the method.*



Enter Numerator: 10

Enter Denominator: Hello

```
Exception:java.lang.NumberFormatException: For input string:
You must enter integers.
Please try again.
```

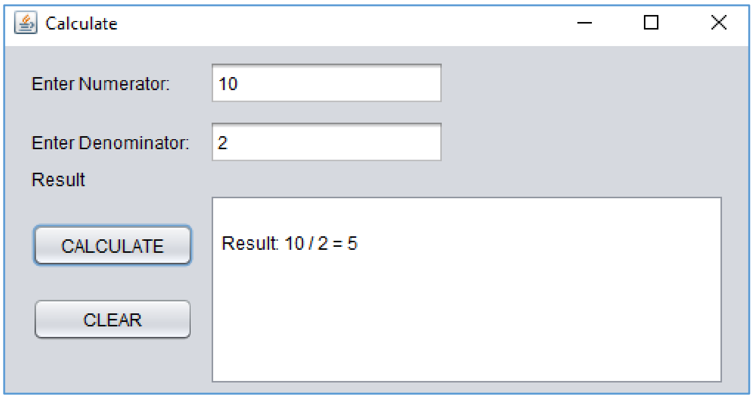


Enter Numerator: 10

Enter Denominator: 0

```
You must enter integers.
Please try again.

Exception:java.lang.ArithmeticException: / by zero
Zero is an invalid denominator.
Please try again
```

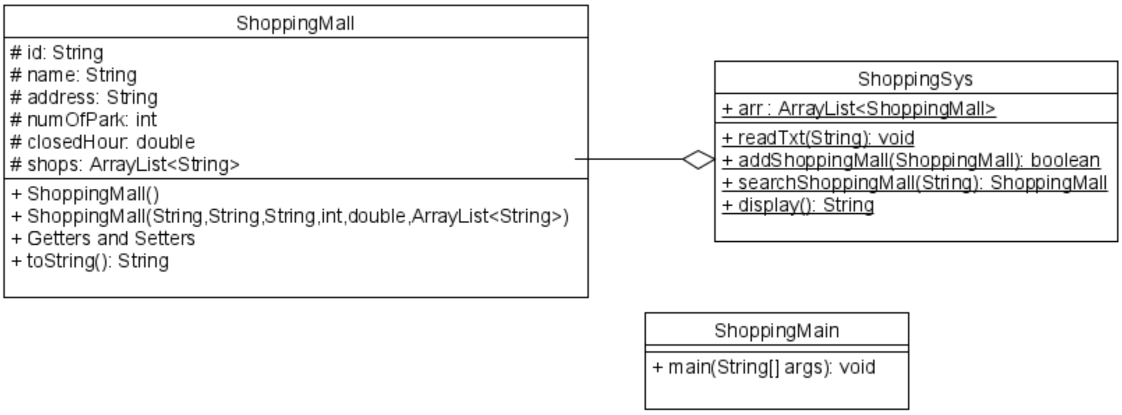


Enter Numerator: 10

Enter Denominator: 2

```
Result: 10 / 2 = 5
```

Q2. Write a Java program that gets ShoppingMall and displays the ShoppingMall information. Create ShoppingMall**,** ShoppingSys classes as shown in the following Class Diagram.


ShoppingMall
# id: String
# name: String
# address: String
# numOfPark: int
# closedHour: double
# shops: ArrayList<String>
+ ShoppingMall()
+ ShoppingMall(String,String,String,int,double,ArrayList<String>)
+ Getters and Setters
+ toString(): String
ShoppingSys
+ arr : ArrayList<ShoppingMall>
+ readTxt(String): void
+ addShoppingMall(ShoppingMall): boolean
+ searchShoppingMall(String): ShoppingMall
+ display(): String
ShoppingMain
+ main(String[] args): void

Create a **ShoppingMall** class, with the following instructions;

- Write data members; **id, name, address, numOfPark, closedHour and ArrayList<String> shops.**
- Write **non-default constructor.**
- Write **getId** method that returns the id of a **ShoppingMall**.
- Write **toString** method to show the content of the **ShoppingMall** object.

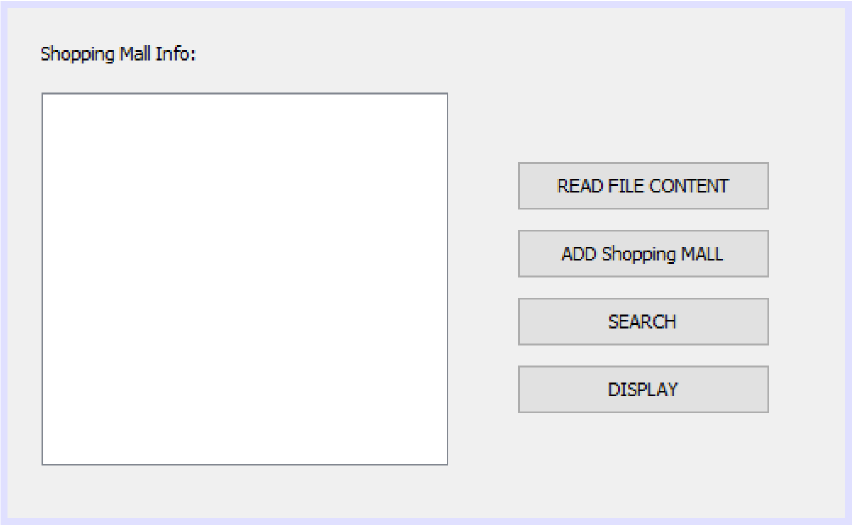
Create a **ShoppingSys** class, with the following instructions;

- Write a static method **readTxt**(…) that gets the file name as a parameter and puts the file content into the arraylist.
- Write a static method **addShoppingMall**(..) that gets a **ShoppingMall** objectFirst, method checks if the id already exist in the arraylist, if it is exist returns false, otherwise adds the object to the arraylist and returns true.
- Write a static method **searchShoppingMall** (…) that gets the id of the **ShoppingMall** then search that **ShoppingMall** from the list and returns that **ShoppingMall** object.
- Write a static method **display**() that returns the whole content of arraylist as a string.

```
shopping.txt
111 CEPA Ankara 1025 10 Penti H&M Ipekyol
222 PANORA Ankara 1256 11.30 Tefal Roman Karaca
333 KENTPLAZA Konya 678 9.30 D&R Accessories Atasun
444 ISTINYEPARK ISTANBUL 2258 12 Micheal&Kors Beymen Vakko
555 CADDE ESKISEHIR 456 10 LCW M&S Panco
```

Create a **ShoppingFrame** class, with the following components;

- 1 label “Shopping Info:”
- 1 textArea and 4 buttons “READ FILE CONTENT”, “ADD Shopping Mall”, “SEARCH”, “DISPLAY”

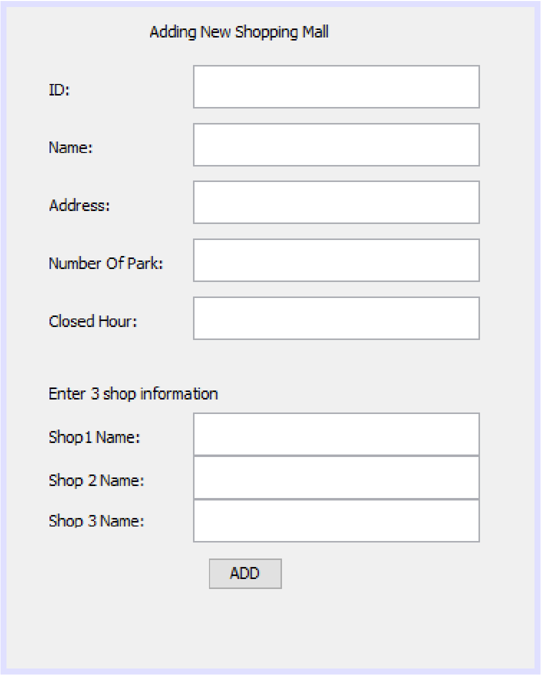


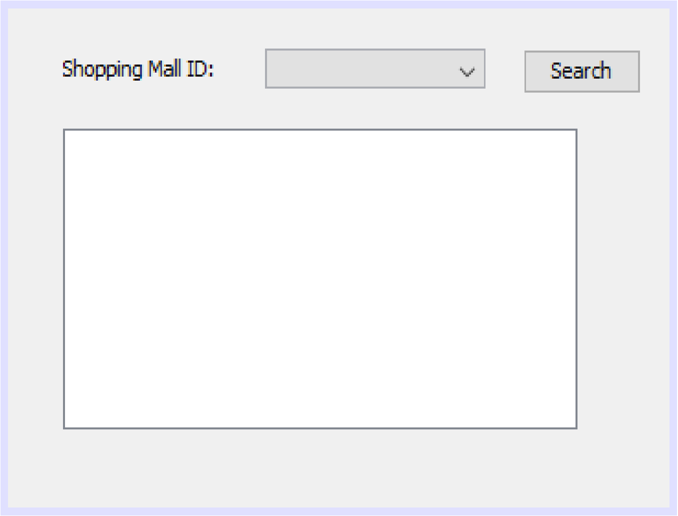
Create an **AddShoppingMallFrame** class, with the following components;

- 1 label “Adding New Shopping Mall:” at the top of the page.
- 8 labels; “ID”,”Name:”,”Address:”,”Number of Park:” ,“Closed Hour:” and Three Shop Names.
- 8 textFields and 1 button “ADD”.

Create an **SearchShoppingMallFrame** class, with the following components;

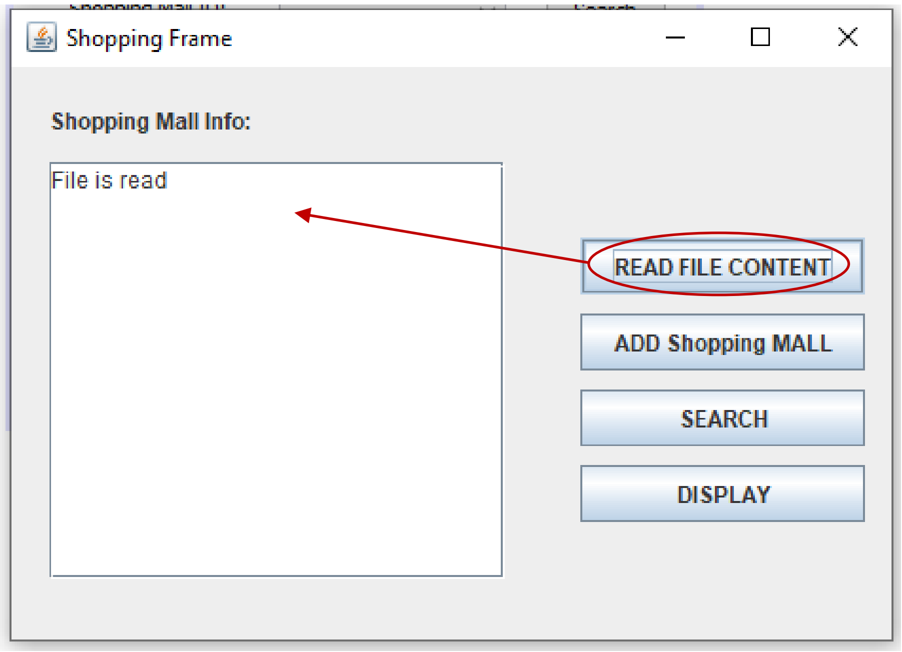
- 1 label “Shopping Mall ID:” at the top of the page.
- 1 ComboBox,
- 1 textarea and 1 button “Search”.



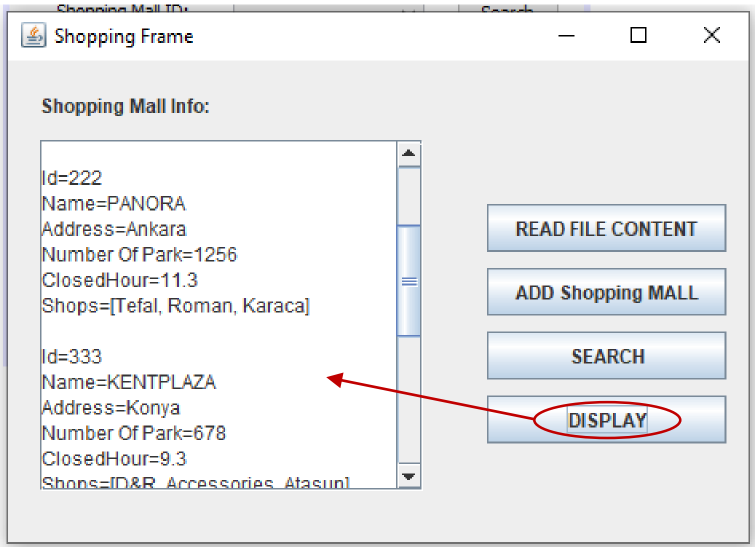


Create a **Main** class that has the static main method, inside create a ShoppingFrame object and sets its visibility to true.

When the user clicks on the “**<u>READ FILE CONTENT</u>**” button, content of the “shopping.txt” will be read and “File is read” message shown in the textArea. Then “**<u>DISPLAY”</u>** button shows the content of ArrayList.



```
File is read
```

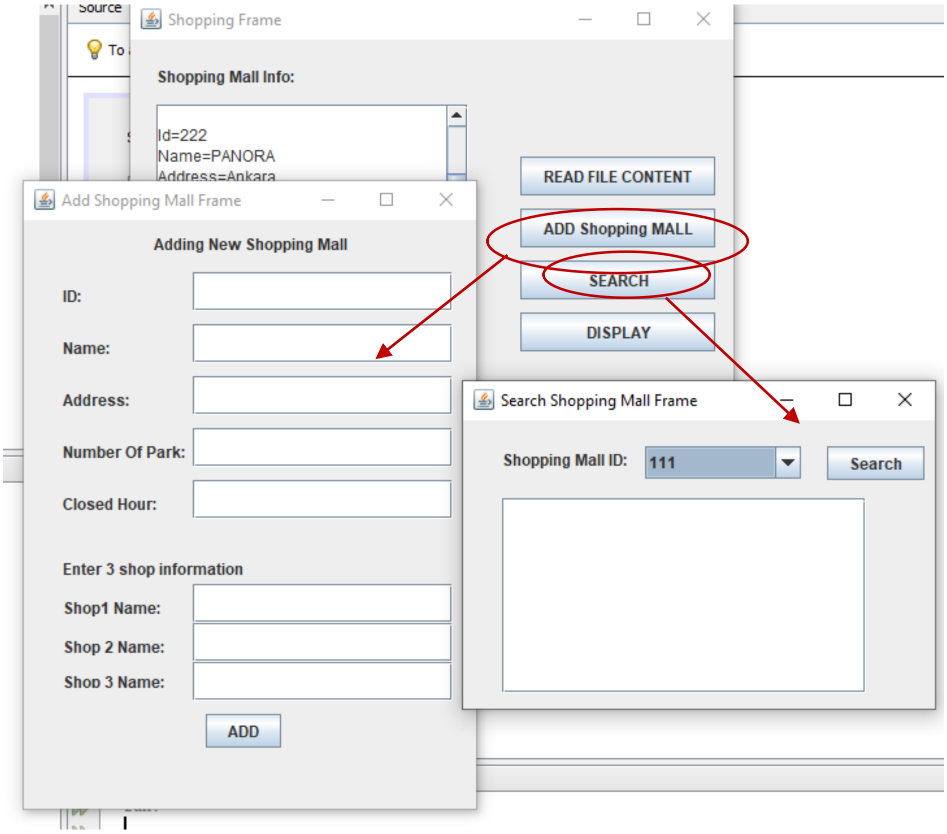


```
Id=222
Name=PANORA
Address=Ankara
Number Of Park=1256
ClosedHour=11.3
Shops=[Tefal, Roman, Karaca]

Id=333
Name=KENTPLAZA
Address=Konya
Number Of Park=678
ClosedHour=9.3
Shops=[D&R, Accessories, Atasun]
```

When the user clicks on the "**ADD**" button, AddShoppingMallFrame will be shown, when the user clicks on the "**SEARCH**" button, **SearchShoppingMallFrame** will be shown, The combobox of the search frame will be filled by the ids of the ShoppingMalls, with the given code of defaultcomboboxmodel.

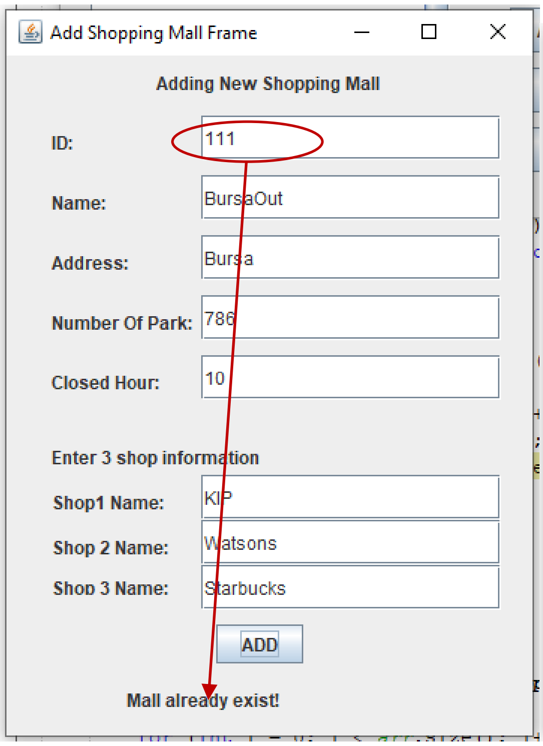
```
ArrayList arr = new ArrayList();
search.getCmbID().removeAllItems();
for (int i = 0; i < ShoppingSys.arr.size(); i++) {
  ShoppingMall sm = ShoppingSys.arr.get(i);
  dcm.addElement(sm.getId());
}
search.getCmbID().setModel(dcm);
search.setVisible(true);
```



```
Id=222
Name=PANORA
Address=Ankara
```

Shopping Mall ID: 111

In the **AddShoppingMallFrame**, get all the information, invoke the addShoppingMall(…) method and depending on the return value, give the message at the bottom of page “Mall already exist!” or the message “Mall Added!”



Add Shopping Mall Frame

Adding New Shopping Mall

ID: 111

Name: BursaOut

Address: Bursa

Number Of Park: 786

Closed Hour: 10

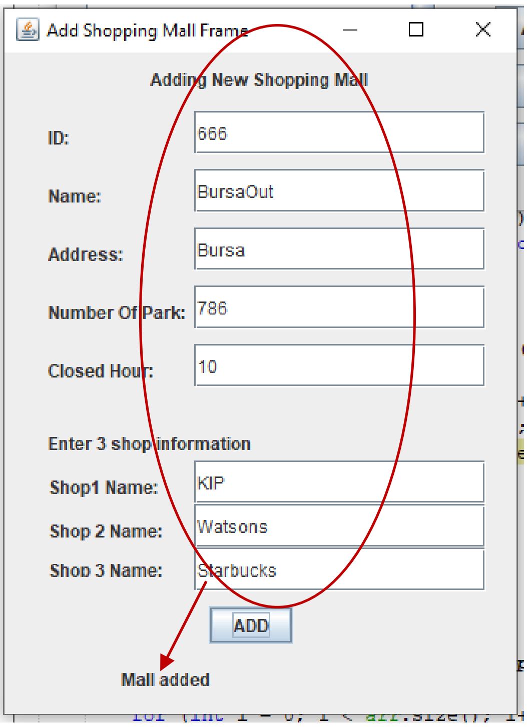
Enter 3 shop information

Shop1 Name: KIP

Shop 2 Name: Watsons

Shop 3 Name: Starbucks

Mall already exist!



Add Shopping Mall Frame

Adding New Shopping Mall

ID: 666

Name: BursaOut

Address: Bursa

Number Of Park: 786

Closed Hour: 10

Enter 3 shop information

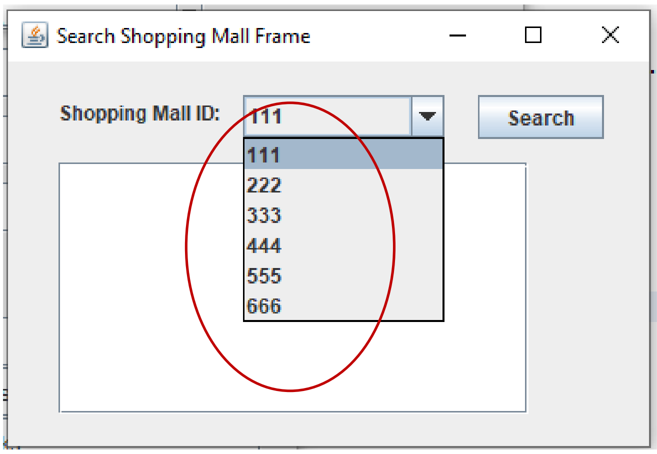
Shop1 Name: KIP

Shop 2 Name: Watsons

Shop 3 Name: Starbucks

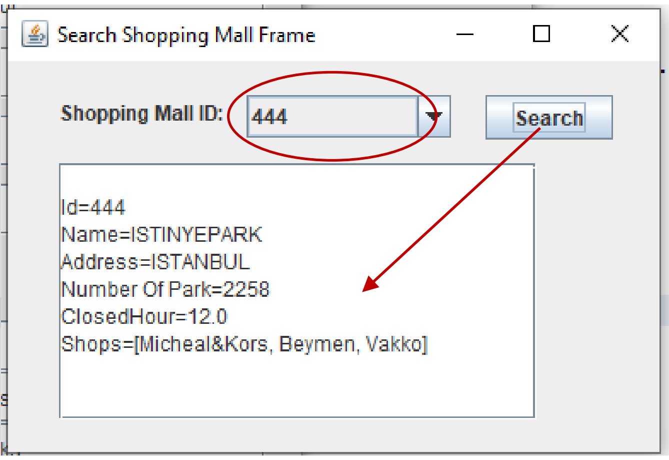
Mall added

In the **SearchShoppingMallFrame**, the combobox will be filled by the IDs of the Malls **(Use DefaultComboBoxModel)**, when user selects the id and clicks the “Search” button, searchShoppingMall(…) will be invoked and the result will be displayed in the text area.



Search Shopping Mall Frame

Shopping Mall ID: 111

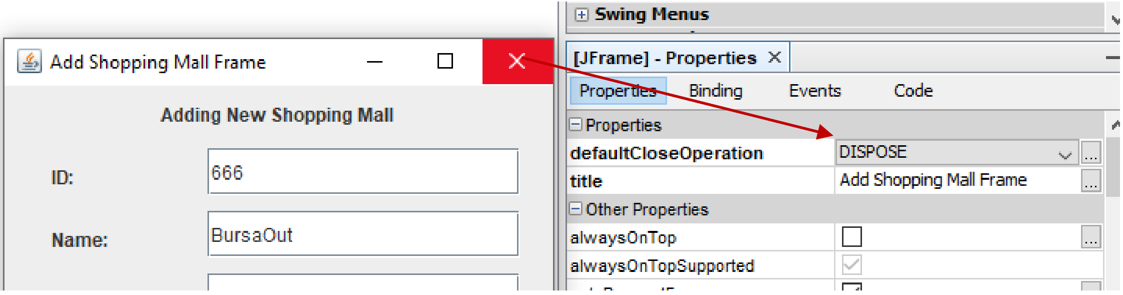


Search Shopping Mall Frame

Shopping Mall ID: 444

```
Id=444
Name=ISTINYEPARK
Address=ISTANBUL
Number Of Park=2258
ClosedHour=12.0
Shops=[Micheal&Kors, Beymen, Vakko]
```

Do not forget the make AddShoppingMallFrame and SearchShoppingMallFrame DISPOSE on close.



Add Shopping Mall Frame

Adding New Shopping Mall

ID: 666

Name: BursaOut

defaultCloseOperation: DISPOSE

title: Add Shopping Mall Frame

alwaysOnTop: □

alwaysOnTopSupported: ☑