

CEGEP VANIER COLLEGE

CENTRE FOR CONTINUING EDUCATION

Programming Algorithms and Patterns

420-930-VA

Teacher: Samir Chebbine

Lab 4

Aug 07, 2023

Lab 4: Generic Classes and Java Collections Framework

Complete all these following programs as explained during **Zoom Synchronous classes**. All *missing coding statements* were provided there with explanation. **Create and Submit** a Word file **Lab4ProgramminAlgorithmsandPatternsYourName.docx** which includes **output screenshots** for every Java Project. Submit the Java projects too.

1. Generic methods & Generic classes

- a) Create *GenericPointProject* using Eclipse IDE for demonstrating the use of generic method and generic classes as shown hereafter in Figure.

```
6
7 Point<String> strPoint =new Point<
8 System.out.println(strPoint);
9
10 Point<Number> Pie =new Point<
11 System.out.println(Pie);
12
13 //test printArray
14
15 Integer[] x = {2, 4, 9, 10};
16 String[] strName = {"Su","Khan","Robertson","Lee"};
17
18 strPoint.printArray(
19 strPoint.printArray(
```

<terminated> TestGenericPoint [Java Application]
Point [x=Anna, y=Banana]
Point [x=3.14, y=2.71]

2. Java Collection Framework (LinkedList, ArrayList)

- a) ListIterator interface

Create a Java Project named *CollectionExamplesProject* to demonstrate the use of ListIterator methods (*hasNext()*, *next()*, *hasPrevious()*, *previous()*) when traversing built-in *LinkedList* data structure in *TestLinkedListCollection.java* as shown hereafter in Figure.

```
6 public static void main(String[] args) {
7 List<String> namelist =
8 String [ ] names =
9
10 int index=0;
11 //Reading from Array names and filling
12 for (index=0; index< names.length; in
13 namelist.
14
15 //Traverse the built-in class Linkedi
16 System.out.println("\nPrinting elemen
17 ListIterator<String> it =
18 String strEle;
19 while(it.
20 {
21 strEle = it.next();
22 System.out.println(strEle);
23 if (strEle.equals("Bob"))
24 }
25 }
26 }
27 }
28 }
29 }
30 }
```

Printing elements of Linked list using get()
Ann
Bob
Carol

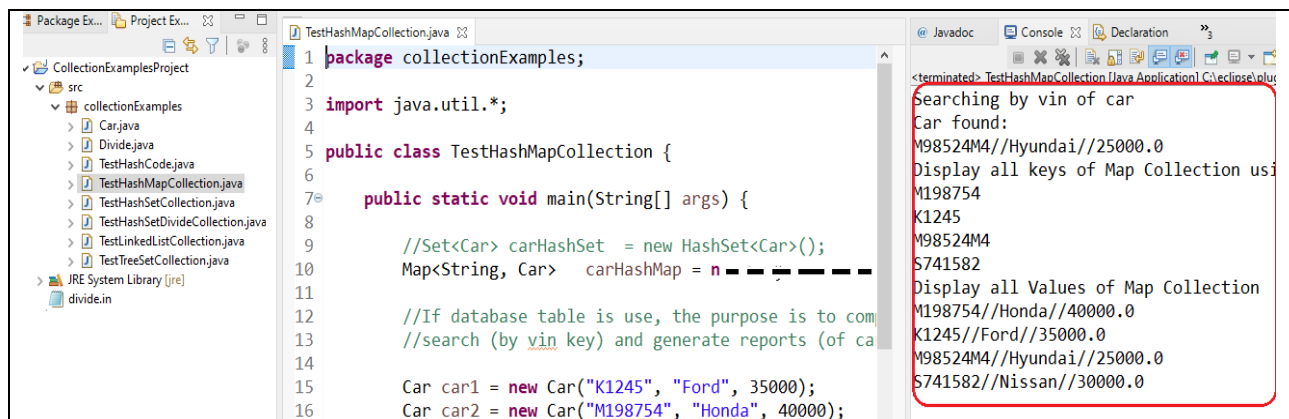
Printing elements of Linked list using Iterator
Ann
Bob
Carol

After Changing Traversing Linked list in Forward
Ann
Lee
Carol

Displaying Elements of Linked list in Backward (Rev)
Carol
Lee
Ann

- b) *TreeSet*, *HashSet*, *LinkedHashSet*, *TreeMap*, *HashMap*, *LinkedHashMap*

Create testing Java classes as done during Zoom class to demonstrate the use of *TreeSet*, *HashSet*, *LinkedHashSet*, *TreeMap*, *HashMap*, *LinkedHashMap* concrete classes as shown hereafter in Figure.



3. Using ArrayList class and its methods add(), size(), get()

- Create *BookArrayListCollectionProject* as shown in Figure 1, to store records of the file *Book.in* (use delimiter \t to read *Book.in*) onto an *ArrayList* of *Book* class type using the method *add()*.
- Use class *Book* (from Lab3) to represent a single record (*b_id*, *b_author*, *b_title*, *b_isbn*, *b_type*, *b_price*).
- Display number of elements of the *ArrayList* using the method *size()*.
- Print all elements of the *ArrayList* using the method *get()*.
- Print all elements of *ArrayList* using the method *next()* of *ListIterator* interface.
- Print all elements of *ArrayList* in reverse order using the method *previous()* of *ListIterator* interface.
- Add a record (13,"Joshawa Pierre","Python","1209845","BG",99.99) into *ArrayList* at index 2 using the method *add(int index, Book wrecord)*.

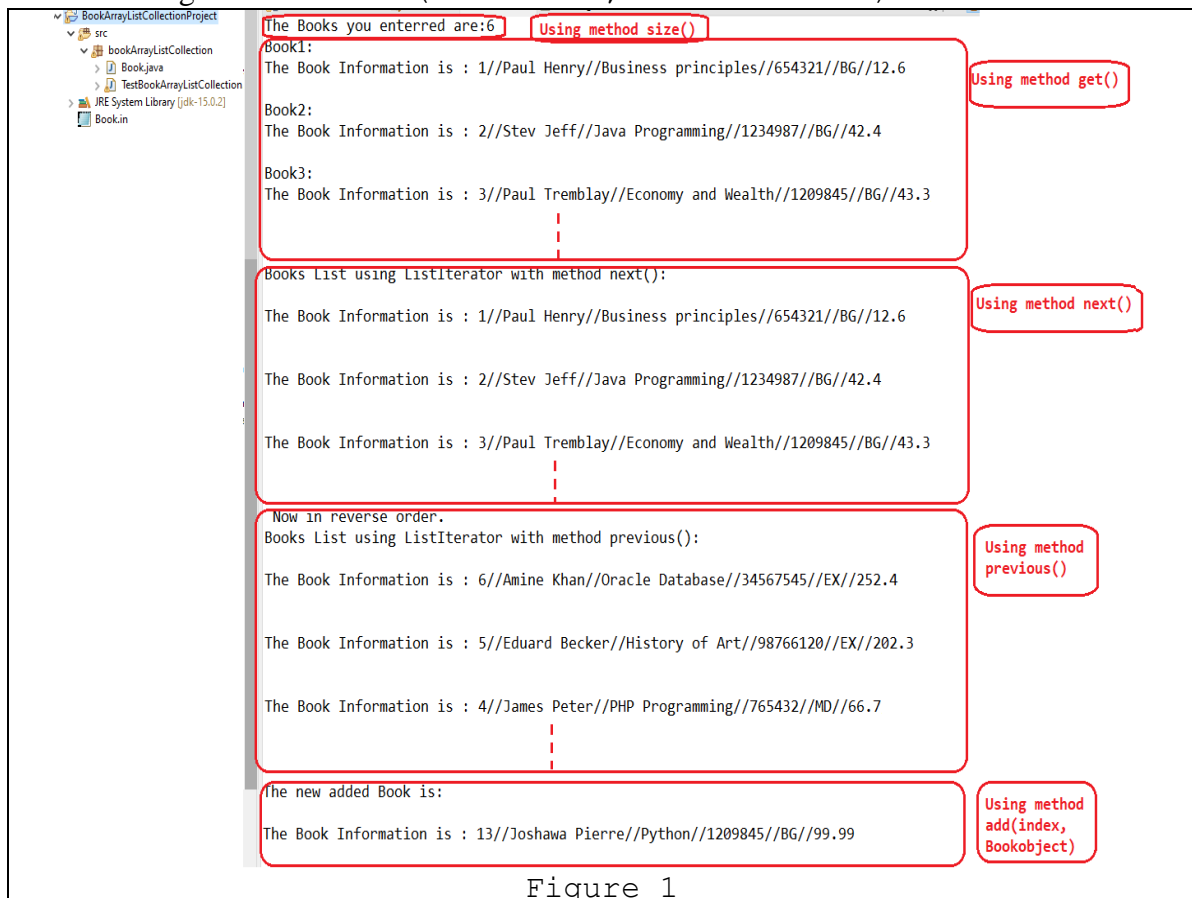


Figure 1

4. Using HashSet class and its methods add(), size()

- Create *CollectionTripProject* as shown in Figure 2, to store the records of the file *Trip.in* (use delimiter \t to read *Trip.in*) onto an *HashSet* using the method *add()*.
- Create a Java class *Trip*, to define data structure type, called *Trip*, which includes the following members:
 - a. The private data members: *emp_id* (Integer), *emp_name* (String), *emp_address* (String), *emp_gasprice* (double), *emp_distance* (int), *emp_costhotel* (double), and *emp_costfood* (double). This order represents the columns in the file *Trip.in*
 - b. Add Mutator (setter) methods in client class to *modify* the values of private members.
 - c. Add Accessor (getter) methods in client class to *access* the values of private members.
 - d. Add a method (*CalculateCostTrip()*) that calculates, and returns the cost of a trip ($\text{cost trip} = (\text{emp_distance} * \text{emp_gasprice}) + \text{emp_costhotel} + \text{emp_costfood}$)
- Add every record stored as an object into *HashSet* using the method *add(Trip wrecord)*
- Display the number of elements of the *HashSet* using the method *size()*.
- Print all elements of the *HashSet*.
- Print all elements of the *HashSet* using the method *next()* of *Iterator* interface.

```
Output - CollectionTripProject (run)
run:
The Employee Trip information you entered are: 6

Emp Id = 1, Emp Name = Stev Jeff, Emp Add = 112, New York Central Parkgas_price = 1.09, distance = 112, cost_hotel = 150.0, cost_food = 40.0, Total Cost = 312.08$
Emp Id = 2, Emp Name = Amine Khan, Emp Add = Paris Francegas_price = 1.11, distance = 50, cost_hotel = 75.0, cost_food = 50.0, Total Cost = 180.50$
Emp Id = 5, Emp Name = Paul Tremblay, Emp Add = Sidney, Australiagas_price = 1.15, distance = 20, cost_hotel = 69.99, cost_food = 35.5, Total Cost = 128.49$
Emp Id = 4, Emp Name = James Peter, Emp Add = Nairobi, Kenyagas_price = 0.99, distance = 300, cost_hotel = 245.0, cost_food = 70.0, Total Cost = 612.00$
Emp Id = 6, Emp Name = Paul Henry, Emp Add = Los_Angelos, USAgas_price = 0.98, distance = 95, cost_hotel = 315.0, cost_food = 85.0, Total Cost = 493.10$
Emp Id = 3, Emp Name = Eduard Becker, Emp Add = Helsinki, Swedengas_price = 1.01, distance = 200, cost_hotel = 110.5, cost_food = 80.0, Total Cost = 392.50$

Using Iterator interface, the Employee Trip information are:

Emp Id = 1, Emp Name = Stev Jeff, Emp Add = 112, New York Central Parkgas_price = 1.09, distance = 112, cost_hotel = 150.0, cost_food = 40.0, Total Cost = 312.08$
Emp Id = 2, Emp Name = Amine Khan, Emp Add = Paris Francegas_price = 1.11, distance = 50, cost_hotel = 75.0, cost_food = 50.0, Total Cost = 180.50$
Emp Id = 5, Emp Name = Paul Tremblay, Emp Add = Sidney, Australiagas_price = 1.15, distance = 20, cost_hotel = 69.99, cost_food = 35.5, Total Cost = 128.49$
Emp Id = 4, Emp Name = James Peter, Emp Add = Nairobi, Kenyagas_price = 0.99, distance = 300, cost_hotel = 245.0, cost_food = 70.0, Total Cost = 612.00$
Emp Id = 6, Emp Name = Paul Henry, Emp Add = Los_Angelos, USAgas_price = 0.98, distance = 95, cost_hotel = 315.0, cost_food = 85.0, Total Cost = 493.10$
Emp Id = 3, Emp Name = Eduard Becker, Emp Add = Helsinki, Swedengas_price = 1.01, distance = 200, cost_hotel = 110.5, cost_food = 80.0, Total Cost = 392.50$

After adding new Client, the List is:

Emp Id = 1, Emp Name = Stev Jeff, Emp Add = 112, New York Central Parkgas_price = 1.09, distance = 112, cost_hotel = 150.0, cost_food = 40.0, Total Cost = 312.08$
Emp Id = 2, Emp Name = Amine Khan, Emp Add = Paris Francegas_price = 1.11, distance = 50, cost_hotel = 75.0, cost_food = 50.0, Total Cost = 180.50$
Emp Id = 5, Emp Name = Paul Tremblay, Emp Add = Sidney, Australiagas_price = 1.15, distance = 20, cost_hotel = 69.99, cost_food = 35.5, Total Cost = 128.49$
Emp Id = 4, Emp Name = James Peter, Emp Add = Nairobi, Kenyagas_price = 0.99, distance = 300, cost_hotel = 245.0, cost_food = 70.0, Total Cost = 612.00$
Emp Id = 6, Emp Name = Paul Henry, Emp Add = Los_Angelos, USAgas_price = 0.98, distance = 95, cost_hotel = 315.0, cost_food = 85.0, Total Cost = 493.10$
Emp Id = 3, Emp Name = Eduard Becker, Emp Add = Helsinki, Swedengas_price = 1.01, distance = 200, cost_hotel = 110.5, cost_food = 80.0, Total Cost = 392.50$
BUILD SUCCESSFUL (total time: 0 seconds)

After adding new Client, the List is:

Emp Id = 5, Emp Name = Paul Tremblay, Emp Add = Sidney, Australiagas_price = 1.15, distance = 20, cost_hotel = 69.99, cost_food = 35.5, Total Cost = 128.49$
Emp Id = 1, Emp Name = Stev Jeff, Emp Add = 112, New York Central Parkgas_price = 1.09, distance = 112, cost_hotel = 150.0, cost_food = 40.0, Total Cost = 312.08$
Emp Id = 2, Emp Name = Amine Khan, Emp Add = Paris Francegas_price = 1.11, distance = 50, cost_hotel = 75.0, cost_food = 50.0, Total Cost = 180.50$
Emp Id = 3, Emp Name = Eduard Becker, Emp Add = Helsinki, Swedengas_price = 1.01, distance = 200, cost_hotel = 110.5, cost_food = 80.0, Total Cost = 392.50$
Emp Id = 4, Emp Name = James Peter, Emp Add = Nairobi, Kenyagas_price = 0.99, distance = 300, cost_hotel = 245.0, cost_food = 70.0, Total Cost = 612.00$
Emp Id = 6, Emp Name = Paul Henry, Emp Add = Los_Angelos, USAgas_price = 0.98, distance = 95, cost_hotel = 315.0, cost_food = 85.0, Total Cost = 493.10$

The Trip Employee information added to the TreeSet with respect to wemp_id, the TreeSet is:

Emp Id = 6, Emp Name = Paul Henry, Emp Add = Los_Angelos, USAgas_price = 0.98, distance = 95, cost_hotel = 315.0, cost_food = 85.0, Total Cost = 493.10$
Emp Id = 5, Emp Name = Paul Tremblay, Emp Add = Sidney, Australiagas_price = 1.15, distance = 20, cost_hotel = 69.99, cost_food = 35.5, Total Cost = 128.49$
Emp Id = 4, Emp Name = James Peter, Emp Add = Nairobi, Kenyagas_price = 0.99, distance = 300, cost_hotel = 245.0, cost_food = 70.0, Total Cost = 612.00$
Emp Id = 3, Emp Name = Eduard Becker, Emp Add = Helsinki, Swedengas_price = 1.01, distance = 200, cost_hotel = 110.5, cost_food = 80.0, Total Cost = 392.50$
Emp Id = 2, Emp Name = Amine Khan, Emp Add = Paris Francegas_price = 1.11, distance = 50, cost_hotel = 75.0, cost_food = 50.0, Total Cost = 180.50$
Emp Id = 1, Emp Name = Stev Jeff, Emp Add = 112, New York Central Parkgas_price = 1.09, distance = 112, cost_hotel = 150.0, cost_food = 40.0, Total Cost = 312.08$
BUILD SUCCESSFUL (total time: 0 seconds)

The Trip Employee information added to the TreeSet with respect to wemp_id, the TreeSet is:

Emp Id = 4, Emp Name = James Peter, Emp Add = Nairobi, Kenyagas_price = 0.99, distance = 300, cost_hotel = 245.0, cost_food = 70.0, Total Cost = 612.00$
Emp Id = 6, Emp Name = Paul Henry, Emp Add = Los_Angelos, USAgas_price = 0.98, distance = 95, cost_hotel = 315.0, cost_food = 85.0, Total Cost = 493.10$
Emp Id = 3, Emp Name = Eduard Becker, Emp Add = Helsinki, Swedengas_price = 1.01, distance = 200, cost_hotel = 110.5, cost_food = 80.0, Total Cost = 392.50$
Emp Id = 1, Emp Name = Stev Jeff, Emp Add = 112, New York Central Parkgas_price = 1.09, distance = 112, cost_hotel = 150.0, cost_food = 40.0, Total Cost = 312.08$
Emp Id = 2, Emp Name = Amine Khan, Emp Add = Paris Francegas_price = 1.11, distance = 50, cost_hotel = 75.0, cost_food = 50.0, Total Cost = 180.50$
Emp Id = 5, Emp Name = Paul Tremblay, Emp Add = Sidney, Australiagas_price = 1.15, distance = 20, cost_hotel = 69.99, cost_food = 35.5, Total Cost = 128.49$
BUILD SUCCESSFUL (total time: 0 seconds)
```

Figure 2

- Add a record (2,"Amine Khan", "Paris France", 1.11, 50, 75.00, 50.00) into the *HashSet* using the method `add(Trip wrecord)` .
- Explain *why* the record was added to the set despite that the set does not accept *duplicated* values as shown in Figure 2.
- Add statements to prevent end user to enter duplicated information related to Trip Employee objects when the *name of employee* is already in the set as shown in Figure 2.
- Add statements to store every record stored as an object into *LinkedHashSet* in order to display the information in the *same order* as found in the input file as shown in Figure 3.

```

The Trip Employee information added to the LinkedHashSet (Notice it keeps the order found in the input file, the LinkedHashSet is:

Emp Id = 1, Emp Name = Stev Jeff, Emp Add = 112, New York Central Parkgas_price = 1.09, distance = 112, cost_hotel = 150.0, cost_food = 40.0, Total Cost = 312.08$
Emp Id = 2, Emp Name = Amine Khan, Emp Add = Paris Francegas_price = 1.11, distance = 50, cost_hotel = 75.0, cost_food = 50.0, Total Cost = 180.50$
Emp Id = 3, Emp Name = Eduard Becker, Emp Add = Helsinki, Swedengas_price = 1.01, distance = 200, cost_hotel = 110.5, cost_food = 80.0, Total Cost = 392.50$
Emp Id = 4, Emp Name = James Peter, Emp Add = Nairobi, Kenyagas_price = 0.99, distance = 300, cost_hotel = 245.0, cost_food = 70.0, Total Cost = 612.00$
Emp Id = 5, Emp Name = Paul Tremblay, Emp Add = Sidney, Australiagas_price = 1.15, distance = 20, cost_hotel = 69.99, cost_food = 35.5, Total Cost = 128.49$
Emp Id = 6, Emp Name = Paul Henry, Emp Add = Los Anglos, USAgas_price = 0.98, distance = 95, cost_hotel = 315.0, cost_food = 85.0, Total Cost = 499.10$

BUILD SUCCESSFUL (total time: 0 seconds)

```

Figure 3

5. Using *HashMap* class and the methods `put()`, `size()`, `get()`, `keyset()`, `values()`

- Create *EmployeeHashMapCollectionProject* as shown in Figure 4, to store records of the file *Employee.in* onto a *HashMap* <key, value> using the method `put()` where key represents the *emp_id* and value represents the record of *Employee* class type.
- Use class *Employee* (from Lab3) to represent a single record (*emp_id*, *emp_name*, *emp_salary*).
- Display number of elements of the *HashMap* using the method `size()` .
- Search for a given *emp_id* into the *HashMap* entered from console and display its information using the method `get()` as shown hereafter.
- Iterate through the keys of *HashMap*, and printing each one of them.
- Iterate through the records stored in the collection and printing each one of them using *Collection* class.

```

EmployeeHashMapCollectionProject
├── src
│   ├── employeeHashMapCollection
│   │   ├── Employee.java
│   │   └── TestEmployeeHashMap.java
│   └── JRE System Library [jdk-15.0.2]
└── Employee.in

Displaying the components of Employee Hash Map list stored
from input file Employee.in

The Trip Employee you entered in the Map are:7

Searching for Employee with emp_id entered from console
Please enter Employee Id: 310
Employee [emp_id=310, emp_name=Emmanuel Jessica, emp_salary=55000.0]

Here are the keys:
210
370
340
310
810
110
910

Here are the records of Employee Information in the HashMap:
Employee [emp_id=210, emp_name=Rollen Bob, emp_salary=98000.0]
Employee [emp_id=370, emp_name=Siroko Ayo, emp_salary=95000.0]
Employee [emp_id=340, emp_name=Mendy Anais, emp_salary=99000.0]
Employee [emp_id=310, emp_name=Emmanuel Jessica, emp_salary=55000.0]
Employee [emp_id=810, emp_name=Fanel Yun, emp_salary=30000.0]
Employee [emp_id=110, emp_name=Rantson Ali, emp_salary=60000.0]
Employee [emp_id=910, emp_name=Simson Sarah, emp_salary=50000.0]

```

Figure 4