Afshan ahmed khan

IT-2k19-03

```java
import java.util.*;
import java.util.LinkedList;
import java.util.Queue;

public class queue {
  public static void main(String[] args) {
    Queue obj = new LinkedList();
    Scanner sc = new Scanner(System.in);

    for (int i = 0; i <= 5; i++) {
      System.out.print("Enter elements of queue ");
      obj.add(sc.next());
    }

    System.out.println("enter any integer element you want to remove");
    Object a = sc.next();
    obj.remove(a);
    System.out.println("enter any String element you want to remove");
    String b = sc.next();
    obj.remove(b);
    System.out.println("printing through foreach loop");

    Iterator itr = obj.iterator();
    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```

```java
import java.util.*;
import java.util.LinkedList;
import java.util.Queue;

public class queue_foreach {
  public static void main(String[] args) {
    Queue obj = new LinkedList();
    Scanner sc = new Scanner(System.in);

    for (int i = 0; i <= 5; i++) {
      System.out.print("Enter elements of queue ");
      obj.add(sc.nextInt());
    }

    System.out.println("enter any integer element you want to remove");
    int a = sc.nextInt();
    obj.remove(a);
    System.out.println("enter any String element you want to remove");
    String b = sc.next();
    obj.remove(b);

    System.out.println("printing through foreach loop");
    for (Object i : obj) {
      System.out.println(i);
    }
  }
}
```



```
                                          nvim                              Q  :    _  □  ×

 2 q/queue.java  t//~/S/a/j//1/b/zsh                                                    x
15 afshan-pc at ~/SEM-6/advanced-java/java_assignment_progs/queue ❯ javac queue_foreach.java -Xlint:unchecke
14 d && java queue
13 queue_foreach.java:12: warning: [unchecked] unchecked call to add(E) as a member of the raw type Queue
12       │ obj.add(sc.next());
11       │ │ │ │ │^
10   where E is a type-variable:
 9   │ E extends Object declared in interface Queue
 8 1 warning
 7 Enter elements of queue 12
 6 Enter elements of queue afshan
 5 Enter elements of queue ahmed
 4 Enter elements of queue 87
 3 Enter elements of queue khan
 2 Enter elements of queue 21
 1 enter any integer element you want to remove
49 21
 1 enter any String element you want to remove
 2 khan
 3 printing through foreach loop
 4 12
 5 afshan
term://~/SEM-6/advanced-java/java_assignment_progs//185462:/bin/zsh                49,1              91%
```

```java
import java.util.Iterator;
import java.util.LinkedList;
import java.util.Queue;

public class queue_generic {

  public static void main(String[] args) {
    Queue<Integer> q = new LinkedList<>();

    for (int i = 0; i < 5; i++)
      q.add(i);

    int removedele = q.remove();
    System.out.println("removed element-"
        + removedele);

    int head = q.peek();
    System.out.println("head of queue-"
        + head);

    int size = q.size();
    System.out.println("Size of queue-"
        + size);

    System.out.println("Traversing through iterators");
    Iterator itr = q.iterator();
    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```

```java
import java.util.LinkedList;
import java.util.Queue;

public class queue_generic_foreach {

  public static void main(String[] args) {
    Queue<Integer> q = new LinkedList<>();

    for (int i = 0; i < 5; i++)
      q.add(i);

    int removedele = q.remove();
    System.out.println("removed element-"
        + removedele);

    int head = q.peek();
    System.out.println("head of queue-"
        + head);

    int size = q.size();
    System.out.println("Size of queue-"
        + size);

    System.out.println("print through foreach loop");
    for (Integer i : q) {
      System.out.println(i);
    }
  }
}
```
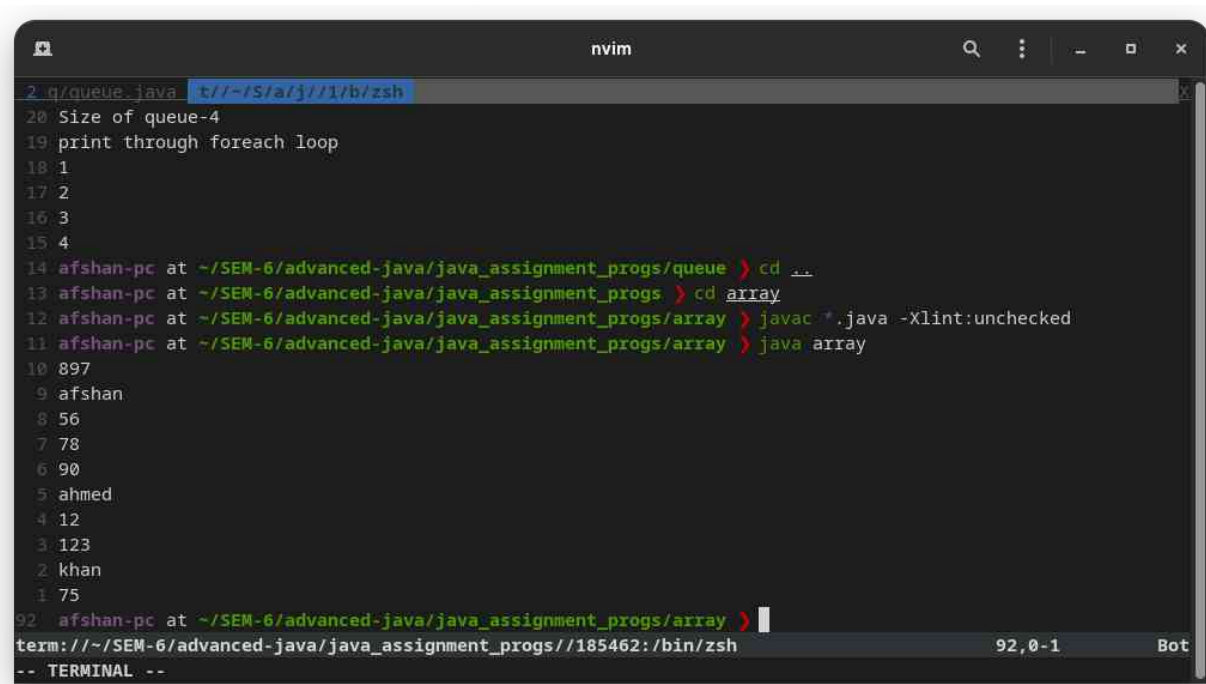
```java
import java.util.Arrays;
import java.util.Iterator;

public class array {
  public static void main(String args[]){
    Object[] myArray = {897,"afshan", 56, 78, 90, "ahmed", 12, 123,"khan", 75};
    Iterator iterator = Arrays.stream(myArray).iterator();
    while(iterator.hasNext()) {
      System.out.println(iterator.next());
    }
  }
}
```

```java
import java.util.Arrays;
import java.util.Iterator;

public class array_foreach {
  public static void main(String args[]){
    Object[] myArray = {897,"afshan", 56, 78, 90, "ahmed", 12, 123,"khan", 75};
    for (Object o : myArray) {
      System.out.println(o);
    }
  }
}
```
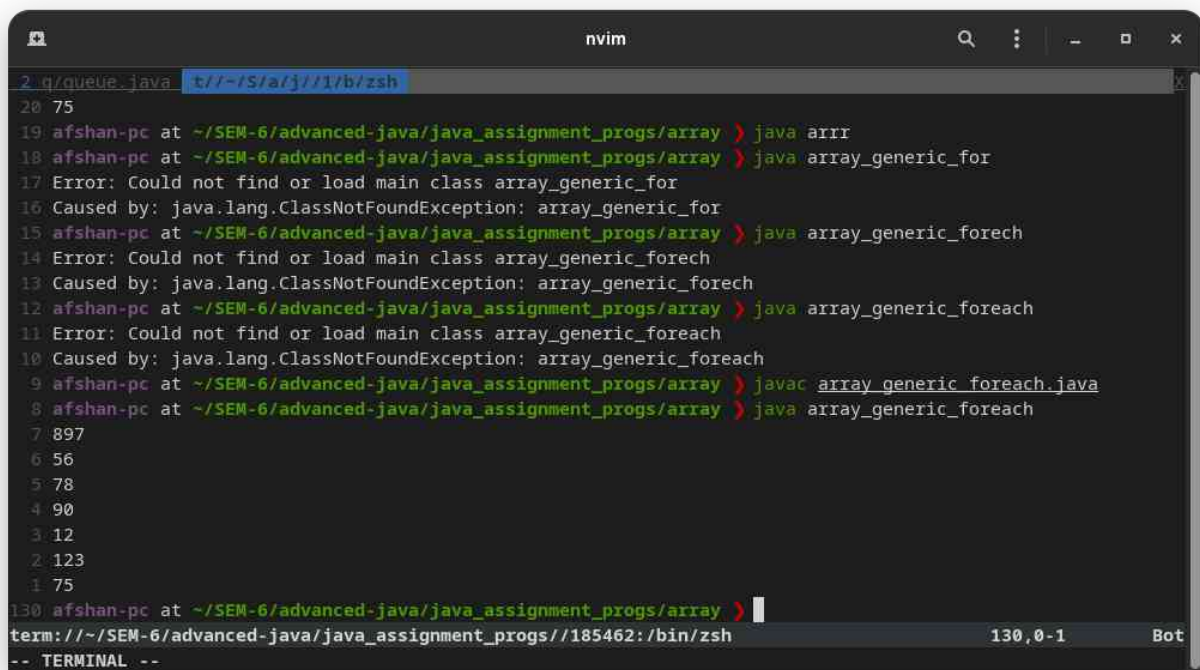
```java
import java.util.Arrays;
import java.util.Iterator;

public class array_generic {
  public static void main(String args[]){
    Integer[] myArray = {897, 56, 78, 90, 12, 123, 75};
    Iterator<Integer> iterator = Arrays.stream(myArray).iterator();
    while(iterator.hasNext()) {
      System.out.println(iterator.next());
    }
  }
}
```

```java
import java.util.Arrays;
import java.util.Iterator;

public class array_generic_foreach {
  public static void main(String args[]){
    Integer[] myArray = {897, 56, 78, 90, 12, 123, 75};
    for (Integer i : myArray) {
      System.out.println(i);
    }
  }
}
```

```java
import java.util.*;

class arraylist {
  public static void main(String args[]) {
    ArrayList<String> list = new ArrayList<String>();// Creating arraylist
    list.add("rohan"); // Adding object in arraylist
    list.add("mohan");
    list.add("sohan");
    list.add("pawan");
    System.out.println("Size of array BEFORE REMOVING " + list.size());

    System.out.println(list.remove("rohan"));
    // Traversing list through Iterator
    Iterator itr = list.iterator();
    System.out.println("Now size of array " + list.size());

    System.out.println("hash code is" + list.hashCode());

    System.out.println("Traversing list through iterator");
    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```
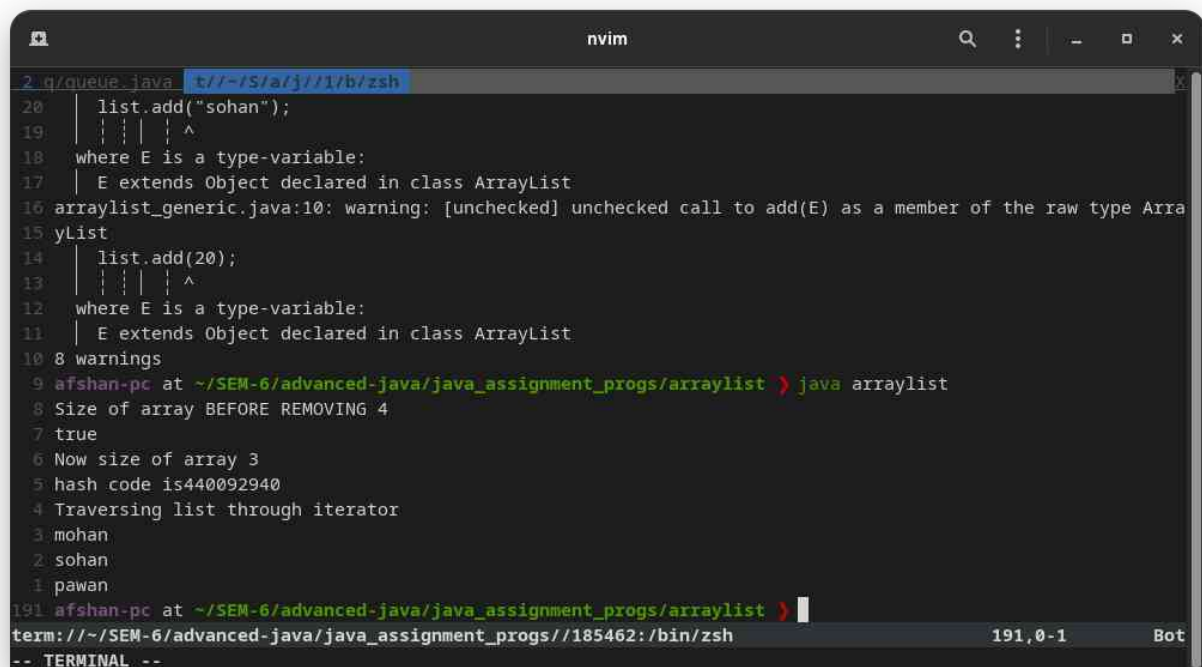
```java
import java.util.*;

public class arraylist_generic {
  public static void main(String args[]) {
    ArrayList list = new ArrayList();

    list.add("rohan");
    list.add("mohan");
    list.add("sohan");
    list.add(20);

    // Traversing list through Iterator
    System.out.println("Traversing list through Iterator");
    Iterator itr = list.iterator();
    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }

}
```

```java
import java.util.*;

class arraylist_generic_foreach {
  public static void main(String args[]) {
    ArrayList list = new ArrayList();

    list.add("rohan");
    list.add("mohan");
    list.add("sohan");
    list.add(20);

    System.out.println("Travering list through foreach loop");
    // Travering list through foreach loop
    for (Object str : list)
      System.out.println(str);
    System.out.println("The clone of list is " + list.clone());
    System.out.println("is this list empty T/F :- " + list.isEmpty());
    System.out.println("The size of the list is " + list.size());
  }

}
```

```java
import java.util.*;
class arraylist_foreach {
  public static void main(String[] args) {
    List<Integer> obj = new ArrayList<Integer>();
    obj.add(152);
    obj.add(343);
    obj.add(139);
    obj.add(123);
    obj.add(358);



    System.out.println("Traversing list through foreach loop");
    for (Integer str : obj) {
      System.out.println(str);
    }
    Collections.sort(obj);
    System.out.println("the assending order is" + obj);
  }
}
```

```java
import java.util.ArrayDeque;
import java.util.Deque;
import java.util.Iterator;
import java.lang.Object;

public class deque {
  public static void main(String[] args) {
    // Creating Deque and adding elements
    Deque<String> obj = new ArrayDeque<String>();
    obj.add("Rohan");
    obj.add("Mohan");
    obj.add("Rohan");
    obj.add("Ajay");
    obj.add("Mohan");

    System.out.println(obj);
    System.out.println("elements contain Rohan ? " + obj.contains("Rohan"));
    System.out.println("removing last Mohan" + obj.removeLastOccurrence("Mohan"));
    System.out.println(obj);
    System.out.println("after removing Rohan");
    obj.removeFirstOccurrence("Rohan");
    System.out.println(obj);
    obj.addFirst("first element");
    obj.addLast("last element");

    System.out.println("Traversing elements through iterator");
    Iterator itr = obj.iterator();
    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```

```java
import java.util.ArrayDeque;
import java.util.Deque;

public class deque_foreach {
  public static void main(String[] args) {

    Deque<Integer> deq = new ArrayDeque<>();
    deq.offer(5);
    deq.offer(6);
    deq.add(7);
    deq.offerFirst(1);

    for (int i : deq) {
      System.out.println(i);
    }

    deq.poll();
    System.out.println("after using poll " + deq);
    deq.pollFirst();
    System.out.println("after using poll first " + deq);
    deq.pollLast();
    System.out.println("after using polllast " + deq);
  }
}
```

```java
import java.util.LinkedList;
import java.util.Deque;
import java.util.Iterator;

public class deque_generic {
  public static void main(String[] args) {
    Deque obj = new LinkedList();
    for (int i = 0; i <= 5; i++) {
      int a = i * 3;
      obj.add(a);
    }
    obj.add("afshan");

    System.out.println("The queue is: " + obj);
    int num1 = (int)obj.removeFirst();
    System.out.println("The element deleted from the head is: " + num1);
    System.out.println("The queue after deletion is: " + obj);

    int head = (int)obj.peek();
    System.out.println("The head of the queue is: " + head);

    System.out.println("Traversing through iterator");
    Iterator itr = obj.iterator();
    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```

```java
import java.util.LinkedList;
import java.util.Deque;

public class deque_generic_foreach {
  public static void main(String[] args) {
    Deque obj = new LinkedList();
    for (int i = 0; i <= 5; i++) {
      int a = i * 3;
      obj.add(a);
    }
    obj.add("afshan");

    System.out.println("The queue is: " + obj);
    int num1 = (int)obj.removeFirst();
    System.out.println("The element deleted from the head is: " + num1);
    System.out.println("The queue after deletion is: " + obj);

    int head = (int)obj.peek();
    System.out.println("The head of the queue is: " + head);

    System.out.println("Traversing through foreach");
    for (Object o : obj) {
      System.out.println(o);
    }
  }
}
```



```
}
```

```java
import java.util.*;

public class arraydeque {
  public static void main(String[] args) {
    ArrayDeque<String> obj = new ArrayDeque<String>();
    obj.add("rohan");
    obj.add("mohan");
    obj.add("sohan");
    // Traversing elements
    System.out.println("Traversing through the iterator");
    Iterator<String> itr = obj.iterator();
    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```

```java
import java.util.*;

public class arraydeque_foreach {
  public static void main(String[] args) {
    ArrayDeque<String> obj = new ArrayDeque<String>();
    obj.add("rohan");
    obj.add("mohan");
    obj.add("sohan");
    // Traversing elements

    System.out.println("Traversing through the for each loop");
    for (String str : obj) {
      System.out.println(str);
    }
  }
}
```

```java
import java.util.*;

public class arraydeque_generic {
  public static void main(String[] args) {
    ArrayDeque obj = new ArrayDeque ();
    obj.add("rohan");
    obj.add("mohan");
    obj.add("sohan");
    obj.add(29);

    // Traversing elements
    System.out.println("Traversing through the iterator");
    Iterator<Object> itr = obj.iterator();
    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```
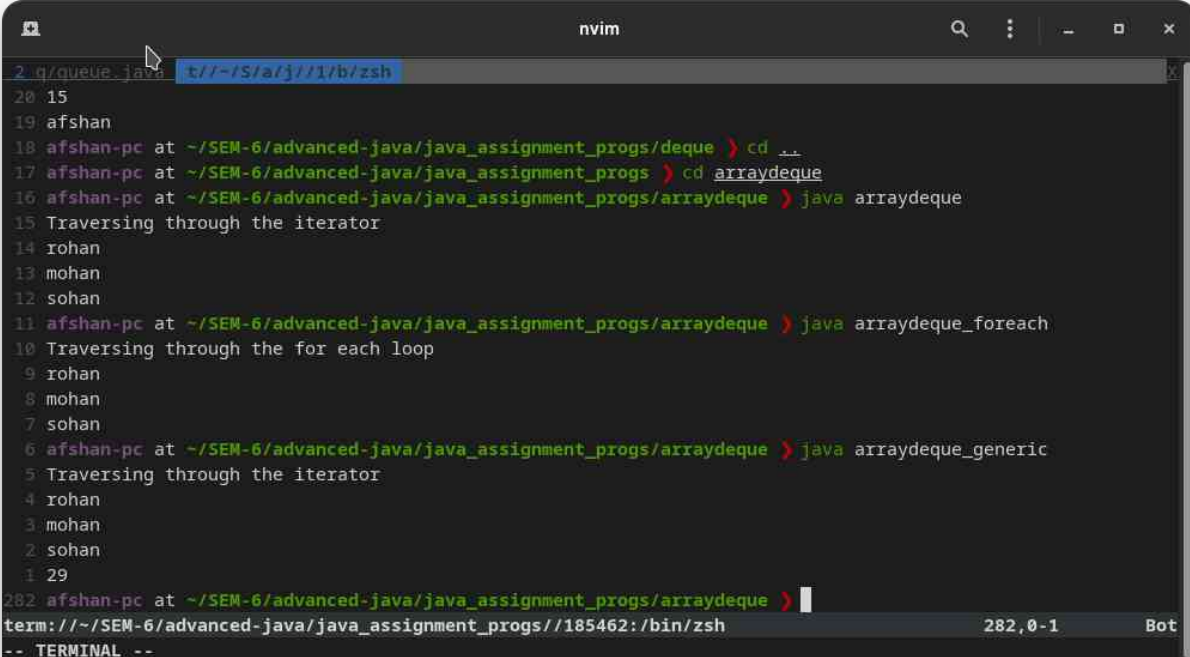
```java
import java.util.*;

public class arraydeque_generic_foreach {
  public static void main(String[] args) {
    ArrayDeque obj = new ArrayDeque();
    obj.add("rohan");
    obj.add("mohan");
    obj.add("sohan");
    obj.add(34);

    // Traversing elements
    System.out.println("Traversing through the for each loop");
    for (Object o : obj) {
      System.out.println(o);
    }
  }
}
```
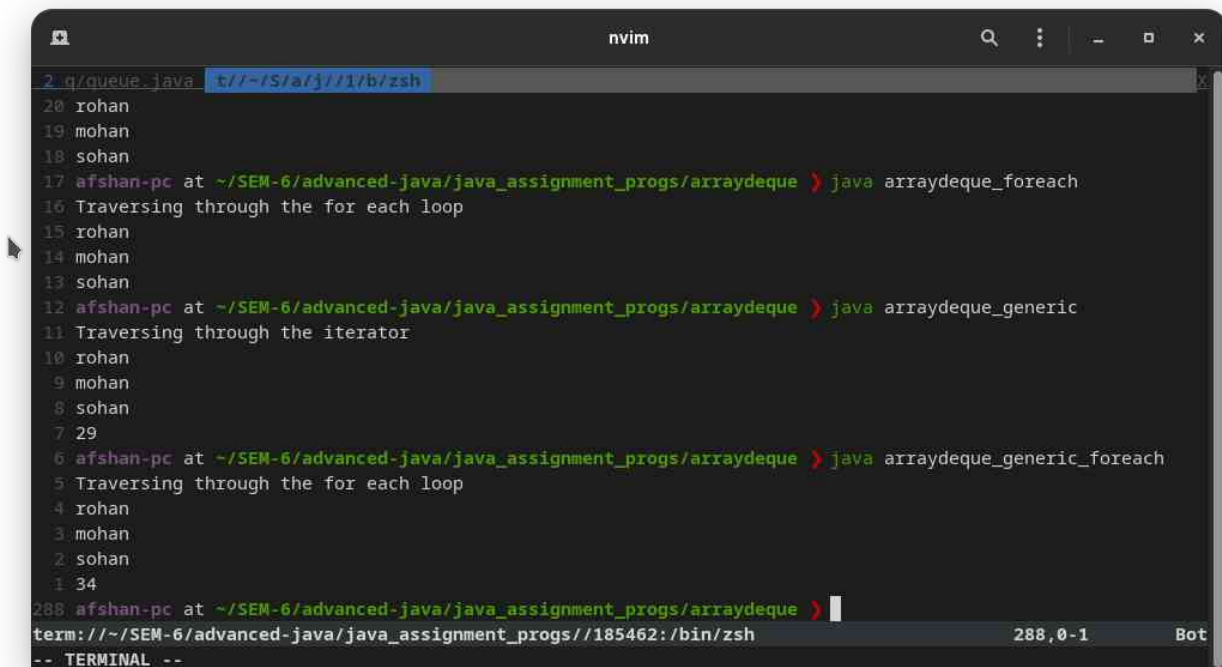
```java
import java.util.Iterator;
import java.util.PriorityQueue;

class priorityqueue {
    public static void main(String[] args) {

        // Creating a priority queue
        PriorityQueue numbers = new PriorityQueue();

        // Using the add() method
        numbers.add(4);
        numbers.add(2);
        numbers.add("ten");
        numbers.add("seven");

        // Using the offer() method
        numbers.offer(1);
        System.out.println("Updated PriorityQueue: " + numbers);

        Iterator itr = numbers.iterator();
        while (itr.hasNext()) {
            System.out.println(itr.next());
        }
    }
}
```

```
2 q/queue.java    t//~/S/a/j//1/b/zsh
20    where E is a type-variable:
19      E extends Object declared in class PriorityQueue
18  priorityqueue.java:12: warning: [unchecked] unchecked call to add(E) as a member of the raw type Priority
17  Queue
16          numbers.add(2);
15                        ^
14    where E is a type-variable:
13      E extends Object declared in class PriorityQueue
12  priorityqueue.java:16: warning: [unchecked] unchecked call to offer(E) as a member of the raw type Priori
11  tyQueue
10          numbers.offer(1);
 9                        ^
 8    where E is a type-variable:
 7      E extends Object declared in class PriorityQueue
 6  3 warnings
 5  afshan-pc at ~/SEM-6/advanced-java/java_assignment_progs/priorityqueue ❯ java priorityqueue
 4  Updated PriorityQueue: [1, 4, 2]
 3  1
 2  4
 1  2
408 afshan-pc at ~/SEM-6/advanced-java/java_assignment_progs/priorityqueue ❯
term://~/SEM-6/advanced-java/java_assignment_progs//185462:/bin/zsh                   408,0-1        Bot
-- TERMINAL --
```

```java
import java.util.PriorityQueue;

public class priorityqueue_foreach {
    public static void main(String[] args) {

        // Creating a priority queue
        PriorityQueue numbers = new PriorityQueue();

        // Using the add() method
        numbers.add(4);
        numbers.add(2);
        numbers.add("ten");
        numbers.add("seven");

        // Using the offer() method
        numbers.offer(1);
        System.out.println("Updated PriorityQueue: " + numbers);

        for (Object o : numbers ) {
          System.out.println(o);
        }
    }
}
```

```java
import java.util.Iterator;
import java.util.PriorityQueue;

class priorityqueue_generic_foreach{
    public static void main(String[] args) {

        // Creating a priority queue
        PriorityQueue<Integer> numbers = new PriorityQueue<>();

        // Using the add() method
        numbers.add(4);
        numbers.add(2);
        System.out.println("PriorityQueue: " + numbers);

        // Using the offer() method
        numbers.offer(1);
        System.out.println("Updated PriorityQueue: " + numbers);

        for (int i : numbers) {
            System.out.println(i);
        }
    }
}
```

```java
import java.util.Iterator;
import java.util.PriorityQueue;

class priorityqueue_generic {
    public static void main(String[] args) {

        // Creating a priority queue
        PriorityQueue<Integer> numbers = new PriorityQueue<>();

        // Using the add() method
        numbers.add(4);
        numbers.add(2);
        numbers.add(5);

        // Using the offer() method
        numbers.offer(1);
        System.out.println("Updated PriorityQueue: " + numbers);

        Iterator<Integer> itr = numbers.iterator();
        while (itr.hasNext()) {
            System.out.println(itr.next());
        }
    }
}
```

```java
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;

public class hashmap {
  public static void main(String[] args) {
    HashMap<Integer,Character> alphabets = new HashMap<>();
    alphabets.put(1, 'A');
    alphabets.put(2, 'B');
    alphabets.put(3, 'C');
    alphabets.put(4, 'D');
    alphabets.put(5, 'E');
    //...

    Set<Entry<Integer,Character>> alphabets_entry = alphabets.entrySet();
    Iterator<Entry<Integer,Character>> itr = alphabets_entry.iterator();

    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```

```java
import java.util.HashMap;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;

public class hashmap_foreach {
  public static void main(String[] args) {
    HashMap<Integer,Character> alphabets = new HashMap<>();
    alphabets.put(1, 'A');
    alphabets.put(2, 'B');
    alphabets.put(3, 'C');
    alphabets.put(4, 'D');
    alphabets.put(5, 'E');
    //...

    Set<Entry<Integer,Character>> alphabets_entry = alphabets.entrySet();

    for (Entry e : alphabets_entry ) {
      System.out.println(e);
    }
  }
}
```
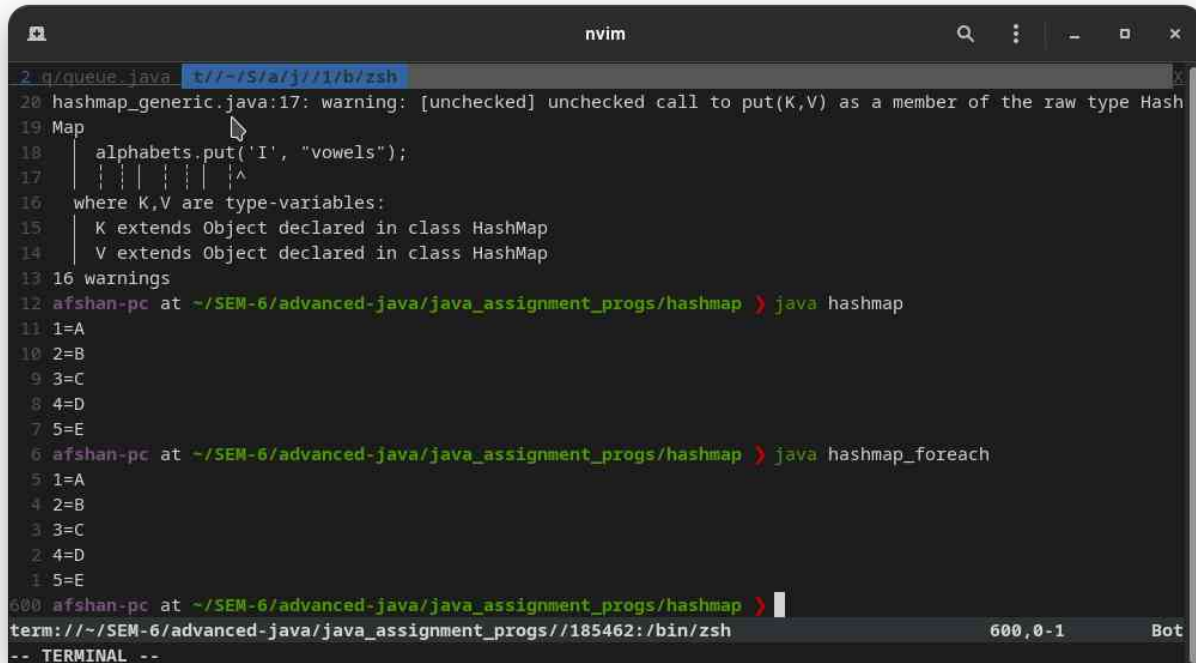
```java
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;

public class hashmap_generic {
  public static void main(String[] args) {
    HashMap alphabets = new HashMap();
    alphabets.put(1, 'A');
    alphabets.put(2, 'B');
    alphabets.put(3, 'C');
    alphabets.put(4, 'D');
    alphabets.put(5, 'E');
    alphabets.put('A', "vowels");
    alphabets.put('E', "vowels");
    alphabets.put('I', "vowels");


    Set alphabets_entry = alphabets.entrySet();
    Iterator itr = alphabets_entry.iterator();

    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```

```java
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;

public class hashmap_generic_foreach {
  public static void main(String[] args) {
    HashMap alphabets = new HashMap();
    alphabets.put(1, 'A');
    alphabets.put(2, 'B');
    alphabets.put(3, 'C');
    alphabets.put(4, 'D');
    alphabets.put(5, 'E');
    alphabets.put('A', "vowels");
    alphabets.put('E', "vowels");
    alphabets.put('I', "vowels");

    //...

    Set alphabets_entry = alphabets.entrySet();
    for (Object o : alphabets_entry) {
      System.out.println(o);
    }
  }
}
```

```java
import java.util.Hashtable;
import java.util.Iterator;
import java.util.Set;
import java.util.Map.Entry;

public class hashtable {
  public static void main(String[] args) {
    Hashtable<Integer,String> studentsList = new Hashtable<>();
    studentsList.put(1,"afshan");
    studentsList.put(2,"rahul");
    studentsList.put(3,"vinayak");
    studentsList.put(4, "xersex");

    Set<Entry<Integer,String>> entryset = studentsList.entrySet();
    Iterator<Entry<Integer,String>> itr = entryset.iterator();

    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```

```java
import java.util.Hashtable;
import java.util.Set;

public class hashtable_foreach {
  public static void main(String[] args) {
    Hashtable<Integer,String> studentsList = new Hashtable<>();
    studentsList.put(1,"afshan");
    studentsList.put(2,"rahul");
    studentsList.put(3,"vinayak");
    studentsList.put(4, "xersex");

    System.out.println(studentsList);
    Set set = studentsList.entrySet();
    for (Object o : set) {
      System.out.println(o);
    }
  }
}
```

```java
import java.util.Hashtable;
import java.util.Iterator;
import java.util.Set;
import java.util.Map.Entry;

public class hashtable_generic {
  public static void main(String[] args) {
    Hashtable studentsList = new Hashtable();
    studentsList.put(1,"afshan");
    studentsList.put(2,"rahul");
    studentsList.put(3,"vinayak");
    studentsList.put(4, "xersex");
    studentsList.put("Nil", "Transfer certificate given");
    studentsList.put(6, "zeus");

    Set entryset = studentsList.entrySet();
    Iterator itr = entryset.iterator();


    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```
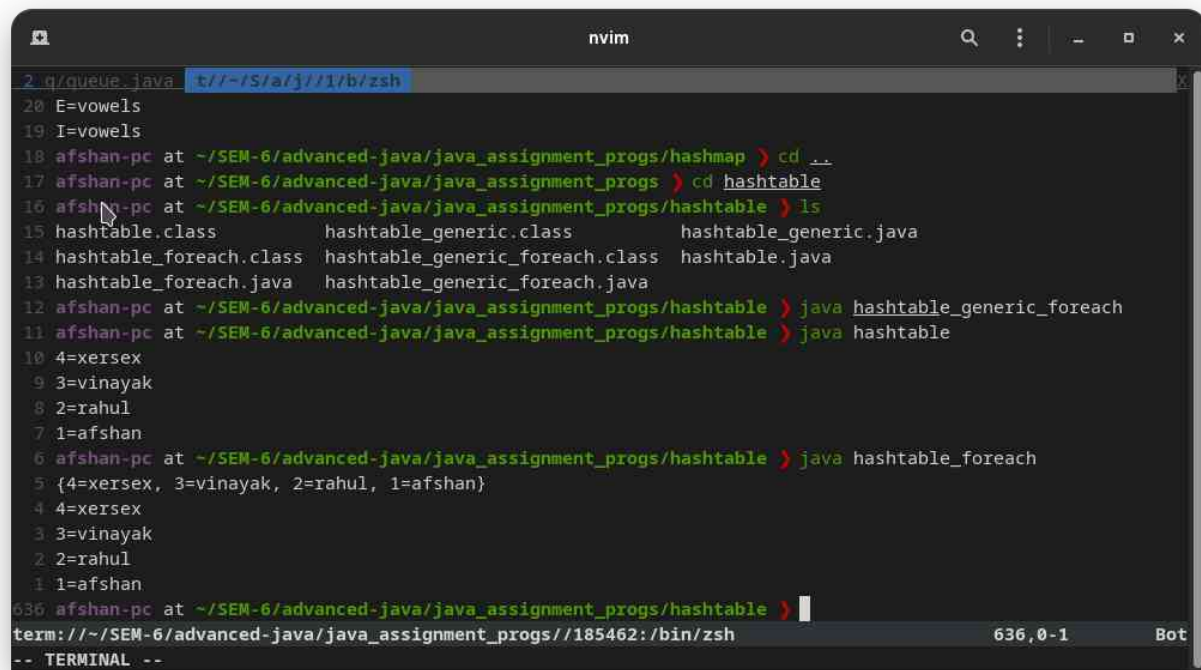
```java
import java.util.Hashtable;
import java.util.Iterator;
import java.util.Set;
import java.util.Map.Entry;

public class hashtable_generic_foreach {
  public static void main(String[] args) {
    Hashtable studentsList = new Hashtable();
    studentsList.put(1,"afshan");
    studentsList.put(2,"rahul");
    studentsList.put(3,"vinayak");
    studentsList.put(4, "xersex");
    studentsList.put("Nil", "Transfer certificate given");
    studentsList.put(6, "zeus");

    Set set = studentsList.entrySet();
    for (Object o : set) {
      System.out.println(o);
    }

  }
}
```

```java
import java.util.*;

public class hashset {
  public static void main(String[] args) {
    HashSet<String> obj = new HashSet<>();
    obj.add("rohan");
    obj.add("sohan");
    obj.add("mohan");
    obj.add("konan");
    // Traversing elements

    Iterator<String> itr = obj.iterator();

    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```

```java
import java.util.*;

public class hashset_foreach {
  public static void main(String[] args) {
    HashSet<String> obj = new HashSet<>();
    obj.add("rohan");
    obj.add("sohan");
    obj.add("mohan");
    obj.add("konan");
    // Traversing elements

    for (String s : obj) {
      System.out.println(s);
    }
  }
}
```

```java
import java.util.*;

public class hashset_generic {
  public static void main(String[] args) {
    HashSet obj = new HashSet();
    obj.add("rohan");
    obj.add("sohan");
    obj.add("mohan");
    obj.add("konan");
    obj.add(1);
    obj.add(2.8);
    // Traversing elements

    Iterator itr = obj.iterator();

    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```
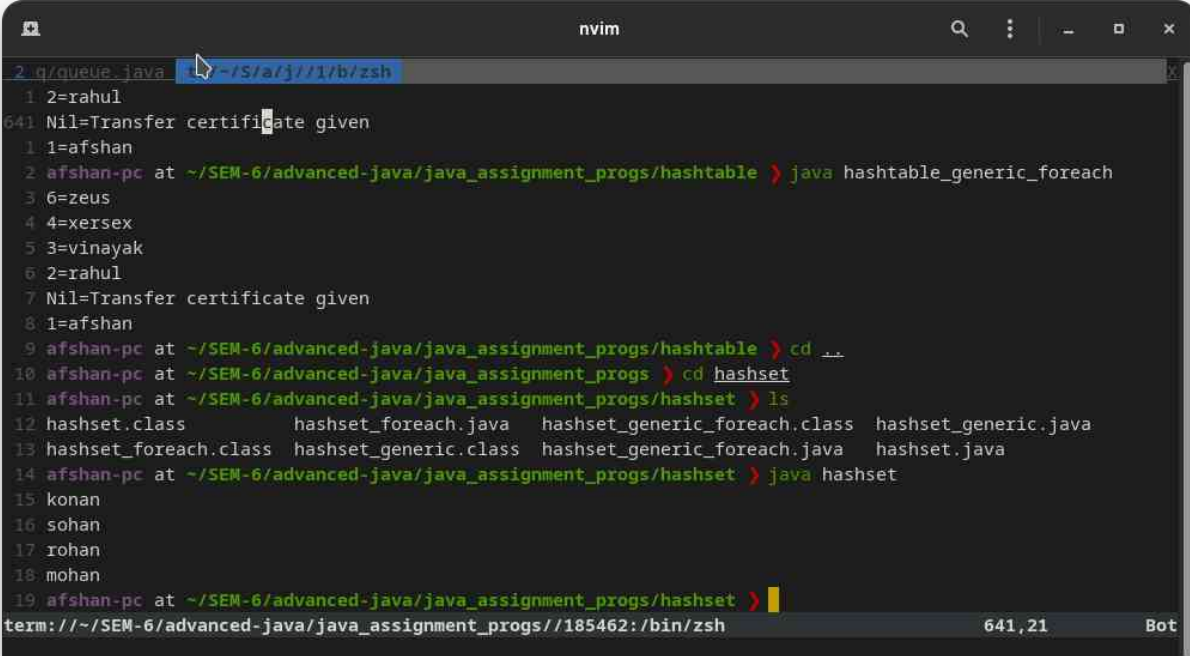
```java
import java.util.*;

public class hashset_generic_foreach {
  public static void main(String[] args) {
    HashSet obj = new HashSet();
    obj.add("rohan");
    obj.add("sohan");
    obj.add("mohan");
    obj.add("konan");
    obj.add(1);
    obj.add(2.8);

    // Traversing elements
    for (Object o : obj) {
      System.out.println(o);
    }
  }

}
```
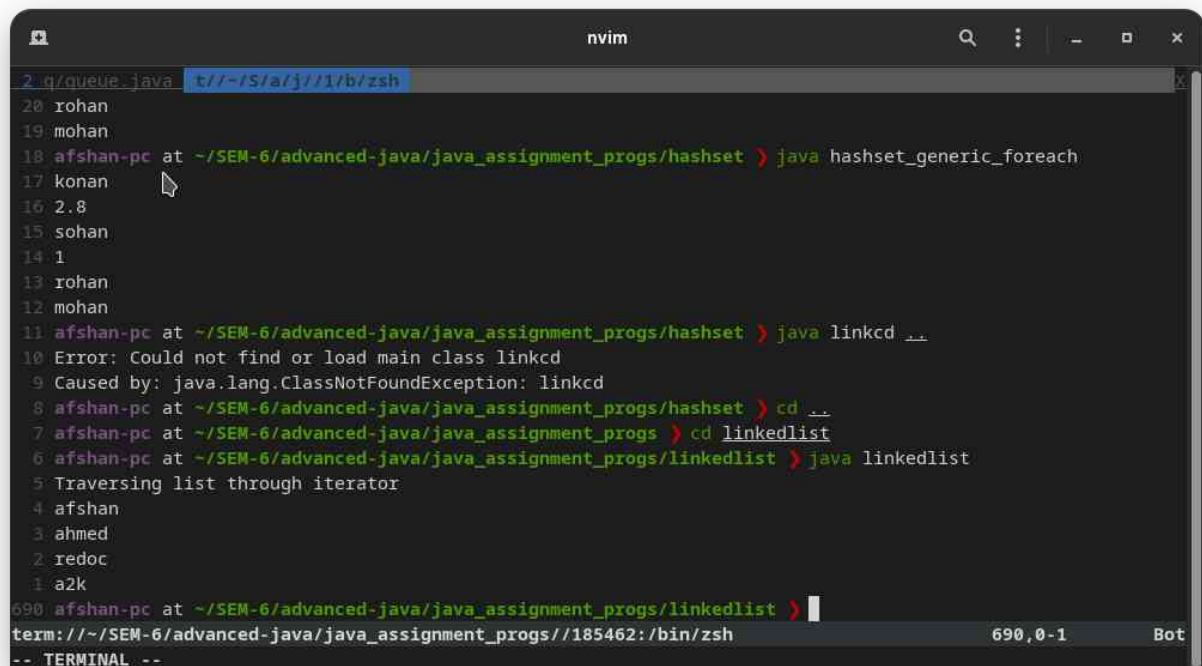
```java
import java.util.*;

public class linkedlist {
  public static void main(String args[]) {
    LinkedList<String> a1 = new LinkedList<String>();
    a1.add("afshan");
    a1.add("ahmed");
    a1.add("redoc");
    a1.add("a2k");

    Iterator<String> itr = a1.iterator();
    System.out.println("Traversing list through iterator");
    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```

```java
import java.util.*;

public class linkedlist_foreach {
  public static void main(String args[]) {
    LinkedList<String> a1 = new LinkedList<String>();
    a1.add("afshan");
    a1.add("ahmed");
    a1.add("a2k");
    a1.add("coder");

    System.out.println("Traversing list through foreach loop");
    for (String str : a1) {
      System.out.println(str);
    }
  }
}
```

```java
import java.util.Iterator;
import java.util.LinkedList;

public class linkedlist_generic {
  public static void main(String[] args) {
    LinkedList obj = new LinkedList();

    obj.add("afshn");
    obj.add("banna");
    obj.add("ahmed");
    obj.add(5);

    obj.remove("4.5");
    obj.remove(1);
    obj.addFirst('A');

    System.out.println("list " + obj);

    obj.removeFirst();
    System.out.println("list after removing first element " + obj);
    obj.removeLast();
    System.out.println("list after removing the first and last element " + obj);
    System.out.println("index of orange is " + obj.indexOf("orange"));

    Iterator itr = obj.iterator();
    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```
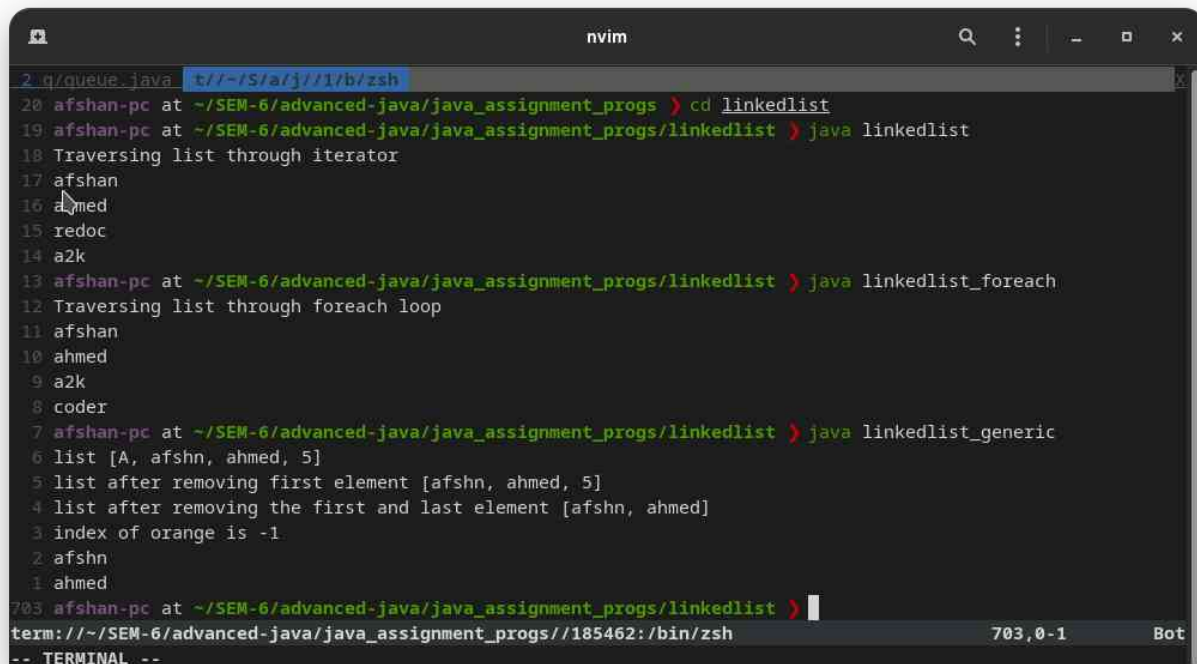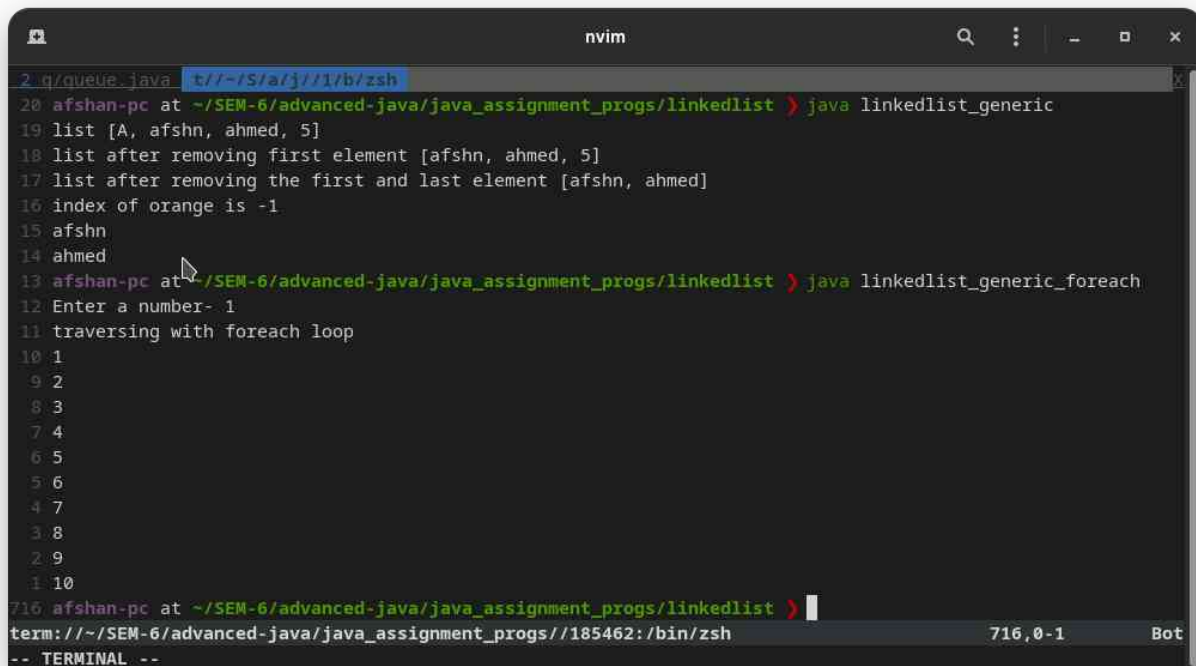
```java
import java.util.*;
import java.util.LinkedList;

public class linkedlist_generic_foreach {
  public static void main(String[] args) {

    LinkedList obj = new LinkedList<>();
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter a number- ");
    int a = sc.nextInt();
    for (int i = 1; i <= 10; i++) {
      obj.add(i * a);
    }
    System.out.println("traversing with foreach loop");
    for (Object in : obj) {
      System.out.println(in);
    }
    sc.close();
  }
}
```

```java
import java.util.Collections;
import java.util.Iterator;
import java.util.Vector;

public class vector {
  public static void main(String[] args) {
    Vector obj = new Vector();
    obj.add("afshan");
    obj.add("it-2k19-03");
    obj.add("iips davv");
    obj.insertElementAt(34, 3);
    obj.insertElementAt(14, 4);
    obj.insertElementAt(99, 5);
    obj.remove(1);
    obj.removeElement(34);
    System.out.println("the last element of vector list is " + obj.lastElement());
    System.out.println(obj);

    Iterator itr = obj.iterator();
    System.out.println("Traversing through iterators");
    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
    System.out.println("On replacing iips-davv ");
    obj.set(2, "International institute of proffessional studys");
    System.out.println(obj);

    System.out.println("the hash code of given vector is " + obj.hashCode());
  }
}
```

```java
import java.util.*;

public class vector_foreach {
  public static void main(String args[]) {
    // initial capacity=4
    Vector obj = new Vector(4);

    obj.add(2);
    obj.add("ramesh");
    obj.add("suresh");
    obj.add("mahesh");
    obj.add("kamlesh");
    obj.add(1);
    // Check size and capacity
    System.out.println("Size is: " + obj.size());
    System.out.println("Default capacity is: " + obj.capacity());

    obj.addElement("mohan");
    obj.addElement("sohan");
    obj.addElement("rohan");

    System.out.println("Size after insertion of more element: " + obj.size());
    System.out.println("Capacity after insertion is: " + obj.capacity());

    System.out.println("Elements are: " + obj);
    // Checking if student is present or not in this vector
    if (obj.contains("ramesh")) {
      System.out.println("ramesh is present at the index " + obj.indexOf("ramesh"));
    } else {
      System.out.println("ramesh is not present in the list.");
    }
    // Get the first element
    System.out.println("The first student of the vector list is = " + obj.firstElement());
    // Get the last element
    System.out.println("The last student of the vector list is = " + obj.lastElement());

    System.out.println("student vector list print through for each loop :-");
    for (Object i : obj) {
      System.out.println(i);
    }
  }
}
```

```
20 the hash code of given vector is -1405003172
19 afshan-pc at ~/SEM-6/advanced-java/java_assignment_progs/vector > java vector_foreach
18 Size is: 6
17 Default capacity is: 8
16 Size after insertion of more element: 9
15 Capacity after insertion is: 16
14 Elements are: [2, ramesh, suresh, mahesh, kamlesh, 1, mohan, sohan, rohan]
13 ramesh is present at the index 1
12 The first student of the vector list is = 2
11 The last student of the vector list is = rohan
10 student vector list print through for each loop :-
 9 2
 8 ramesh
 7 suresh
 6 mahesh
 5 kamlesh
 4 1
 3 mohan
 2 sohan
 1 rohan
748 afshan-pc at ~/SEM-6/advanced-java/java_assignment_progs/vector >
term://~/SEM-6/advanced-java/java_assignment_progs//185462:/bin/zsh                748,0-1        Bot
-- TERMINAL --
```
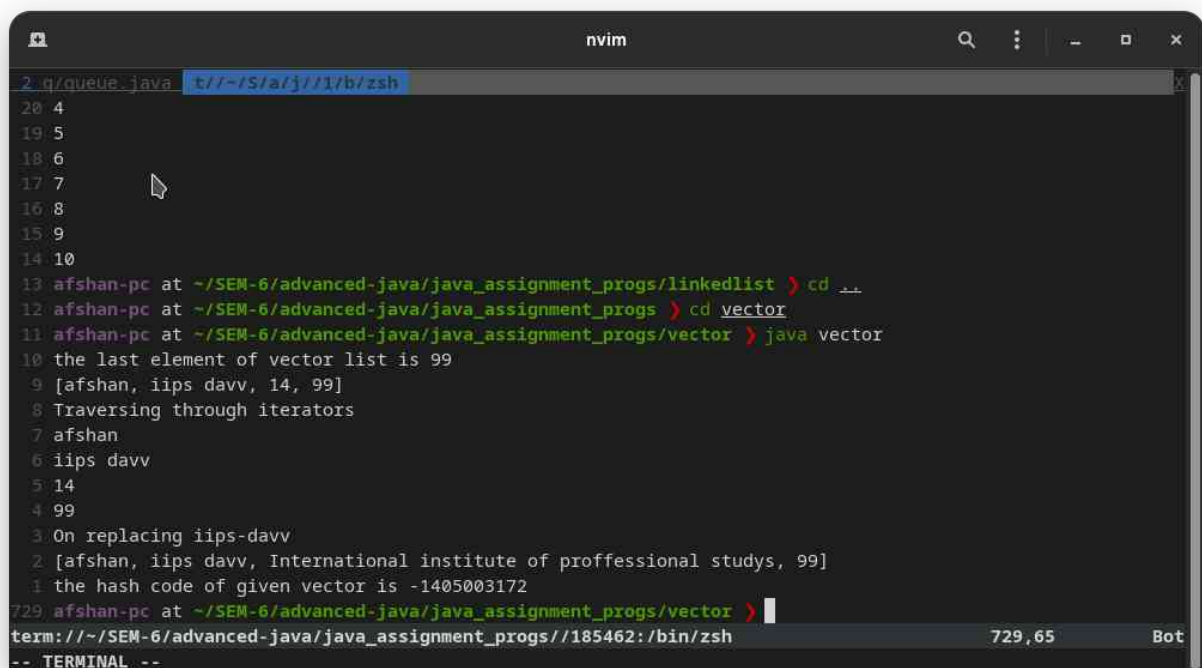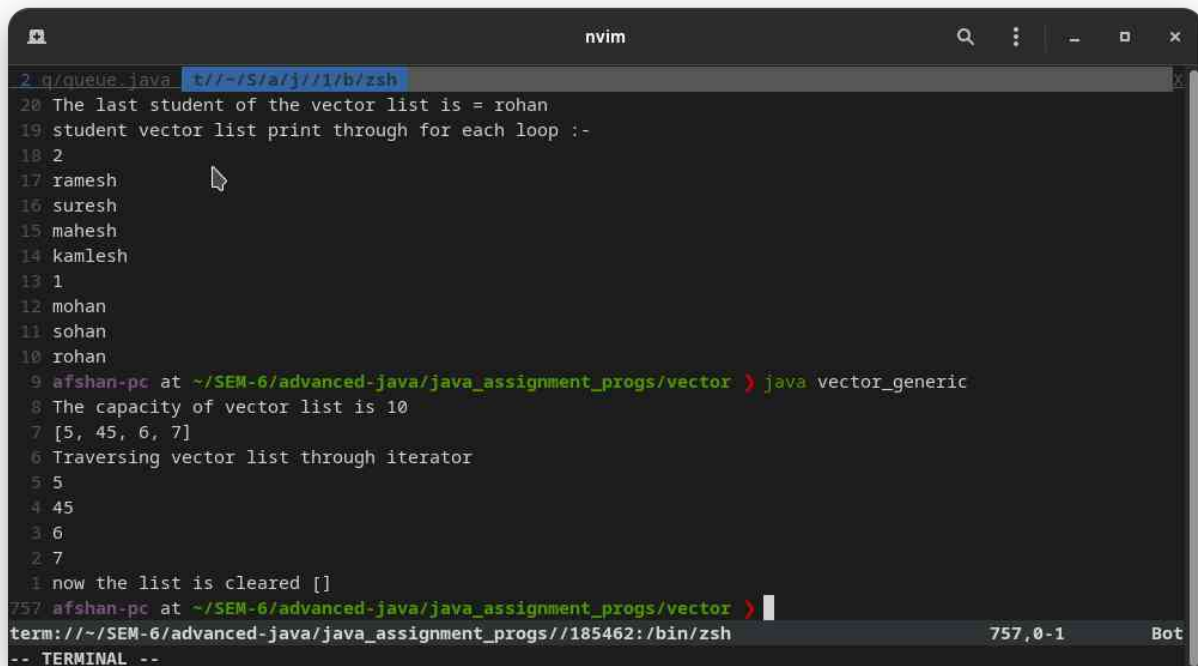
```java
import java.util.*;

public class vector_generic {

  public static void main(String[] args) {
    Vector<Integer> obj = new Vector<Integer>();
    obj.add(5);
    obj.add(6);
    obj.add(7);
    obj.add(1, 45);// in 1st place 45 is inserted
    System.out.println("The capacity of vector list is " + obj.capacity());

    Iterator<Integer> itr = obj.iterator();
    System.out.println(obj);
    System.out.println("Traversing vector list through iterator");
    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
    obj.clear();
    System.out.println("now the list is cleared " + obj);
  }
}
```

```java
import java.util.*;

public class vector_generic_foreach {

  public static void main(String[] args) {
    Vector<Integer> obj = new Vector<Integer>();
    obj.add(5);
    obj.add(6);
    obj.add(7);
    obj.add(1, 45);// in 1st place 45 is inserted
    System.out.println("The capacity of vector list is " + obj.capacity());

    System.out.println("Traversing vector list through foreach loop");
    for (Integer in : obj) {
      System.out.println(in);
    }
    obj.clear();
    System.out.println("now the list is cleared " + obj);
  }
}
```

```java
import java.util.*;

public class stack {
  public static void main(String[] args) {
    // Creating a Stack
    Stack nos = new Stack();
    // pushing values in stack
    nos.push("This is the start");
    for (int i = 0; i <= 10; i++) {
      nos.push(i);
    }
    nos.push("This is the end");

    System.out.println("Traversing with Iterators");
    Iterator itr = nos.iterator();
    while (itr.hasNext()) {
      System.out.println(itr.next());
    }

    System.out.println("after removing the element at index 1");
    nos.removeElementAt(1);
    System.out.println(nos);

    // printing the stack
    System.out.println("Print Stack before pop:");
    System.out.println(nos);
    System.out.println("Position of 8 from the top is " + nos.search(8));
    System.out.println("Position of 10 from the top is " + nos.search(10));

    // returning the number at the top of the stack and removing it
    System.out.println("pop => " + nos.pop());
    System.out.println("pop => " + nos.pop());
    System.out.println("pop => " + nos.pop());

    // printing the stack
    System.out.println("Print Stack after pop:");
    System.out.println(nos);
    // accessing the number at the top of the stack but not removing it
    System.out.println("Number on top of the stack is => " + nos.peek());
    // checking if the stack is empty or not
    System.out.println("Is stack empty?  Ans:" + nos.empty());
  }
}
```

```
2 q/queue.java   t//~/S/a/j//1/b/zsh
11 afshan-pc at ~/SEM-6/advanced-java/java_assignment_progs ❯ cd stack
10 afshan-pc at ~/SEM-6/advanced-java/java_assignment_progs/stack ❯ java stack
 9 Traversing with Iterators
 8 This is the start
 7 0
 6 1
 5 2
 4 3
 3 4
 2 5
 1 6
778 7
 1 8
 2 9
 3 10
 4 This is the end
 5 after removing the element at index 1
 6 [This is the start, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, This is the end]
 7 Print Stack before pop:
 8 [This is the start, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, This is the end]
 9 Position of 8 from the top is 4
10 Position of 10 from the top is 2
11 pop => This is the end
12 pop => 10
13 pop => 9
14 Print Stack after pop:
15 [This is the start, 1, 2, 3, 4, 5, 6, 7, 8]
16 Number on top of the stack is => 8
```

```java
import java.util.Iterator;
import java.util.Stack;

public class stack_generic {
  public static void main(String[] args) {
    Stack<String> obj = new Stack<>();
    obj.push("afshan");
    obj.push("ahmed");
    obj.push("rohan");
    obj.push("mohan");
    obj.pop();
    System.out.println("size of stack is " + obj.size());
    System.out.println("capacity is " + obj.capacity());
    System.out.println("is stacj empty?" + obj.empty());
    System.out.println("speek of stack -" + obj.peek());
    System.out.println("last element of stack is - " + obj.lastElement());
    System.out.println("the index of rai is " + obj.search("rai"));

    Iterator<String> itr = obj.iterator();

    System.out.println("Traversing the stack through the iterator");
    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```

```java
import java.util.Stack;
import java.util.Vector;

public class stack_foreach {
  public static void main(String[] args) {
    Stack stk = new Stack();
    // pushing elements into Stack
    stk.push("tomato");
    stk.push("brinjal");
    stk.push("cauliflower");
    stk.push("chilli");
    stk.push(1);
    stk.push(2);

    boolean rsult = stk.empty();
    System.out.println("Is the stack empty or not? " + rsult);

    int x = stk.size();
    System.out.println("The stack size is: " + x);
    System.out.println("Stack: " + stk);
    // Access element from the top of the stack
    Object vegetable  = stk.peek();
    // prints stack
    System.out.println("Element at top: " + vegetable);

    System.out.println("By for each loop");
    for (Object i : stk) {
      System.out.println(i);
    }
  }

}
```

```java
import java.util.Stack;

public class stack_generic_foreach {
  public static void main(String[] args) {
    Stack<String> obj = new Stack<>();
    obj.push("afshan");
    obj.push("ahmed");
    obj.push("rohan");
    obj.push("mohan");
    obj.pop();
    System.out.println("size of stack is " + obj.size());
    System.out.println("capacity is " + obj.capacity());
    System.out.println("is stacj empty?" + obj.empty());
    System.out.println("speek of stack -" + obj.peek());
    System.out.println("last element of stack is - " + obj.lastElement());
    System.out.println("the index of rai is " + obj.search("rai"));

    System.out.println("Traversing the stack through the foreach");
    for (String s : obj) {
      System.out.println(s);
    }
  }
}
```
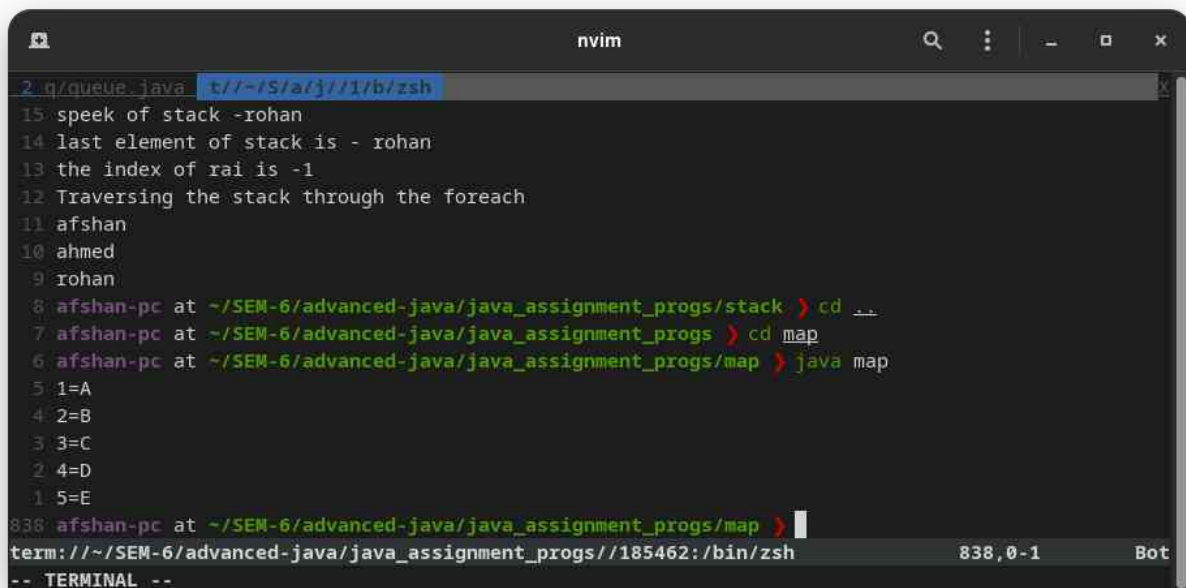
```java
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;

public class map {
  public static void main(String[] args) {
    Map<Integer,Character> alphabets = new HashMap<>();
    alphabets.put(1, 'A');
    alphabets.put(2, 'B');
    alphabets.put(3, 'C');
    alphabets.put(4, 'D');
    alphabets.put(5, 'E');
    //...

    Set<Entry<Integer,Character>> alphabets_entry = alphabets.entrySet();
    Iterator<Entry<Integer,Character>> itr = alphabets_entry.iterator();

    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```
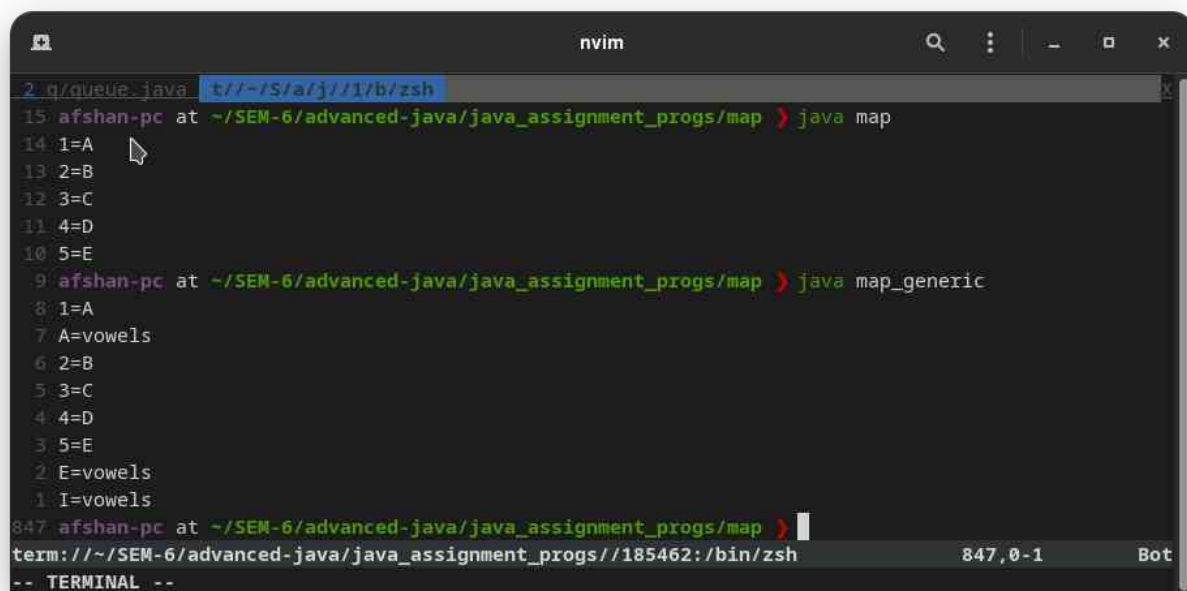
```java
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;

public class map_generic {
  public static void main(String[] args) {
    Map alphabets = new HashMap();
    alphabets.put(1, 'A');
    alphabets.put(2, 'B');
    alphabets.put(3, 'C');
    alphabets.put(4, 'D');
    alphabets.put(5, 'E');
    alphabets.put('A', "vowels");
    alphabets.put('E', "vowels");
    alphabets.put('I', "vowels");


    Set alphabets_entry = alphabets.entrySet();
    Iterator itr = alphabets_entry.iterator();

    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```
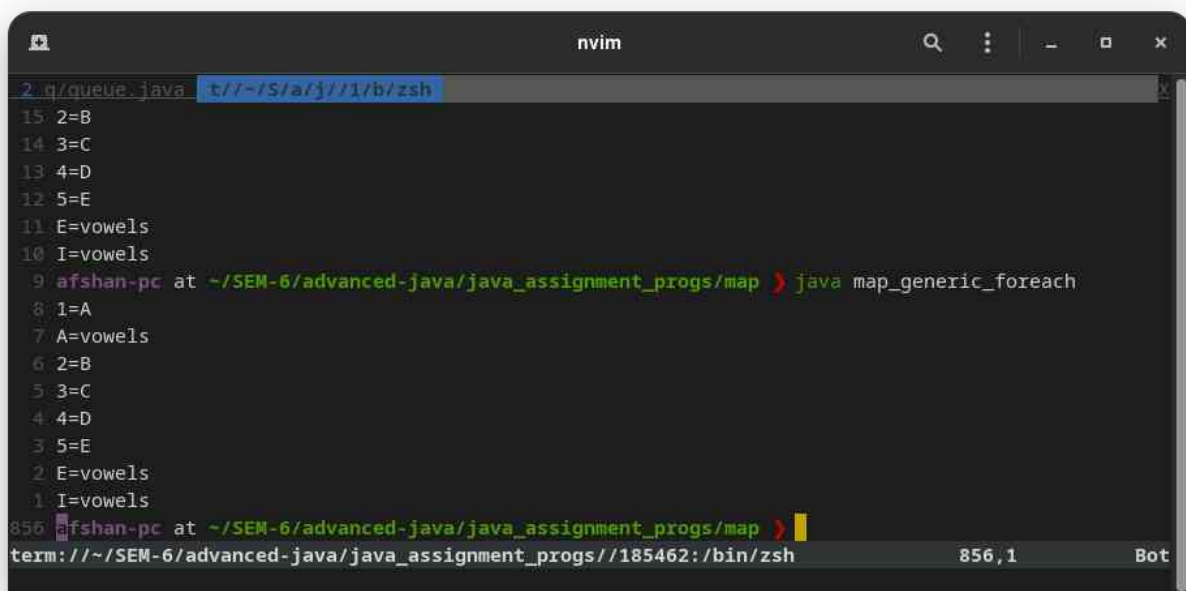
```java
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;

public class map_generic_foreach {
  public static void main(String[] args) {
    Map alphabets = new HashMap();
    alphabets.put(1, 'A');
    alphabets.put(2, 'B');
    alphabets.put(3, 'C');
    alphabets.put(4, 'D');
    alphabets.put(5, 'E');
    alphabets.put('A', "vowels");
    alphabets.put('E', "vowels");
    alphabets.put('I', "vowels");

    //...

    Set alphabets_entry = alphabets.entrySet();
    for (Object o : alphabets_entry) {
      System.out.println(o);
    }
  }
}
```

```java
import java.util.HashMap;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Set;

public class map_foreach {
  public static void main(String[] args) {
    Map<Integer,Character> alphabets = new HashMap<>();
    alphabets.put(1, 'A');
    alphabets.put(2, 'B');
    alphabets.put(3, 'C');
    alphabets.put(4, 'D');
    alphabets.put(5, 'E');
    //...

    Set<Entry<Integer,Character>> alphabets_entry = alphabets.entrySet();

    for (Entry e : alphabets_entry ) {
      System.out.println(e);
    }
  }
}
```