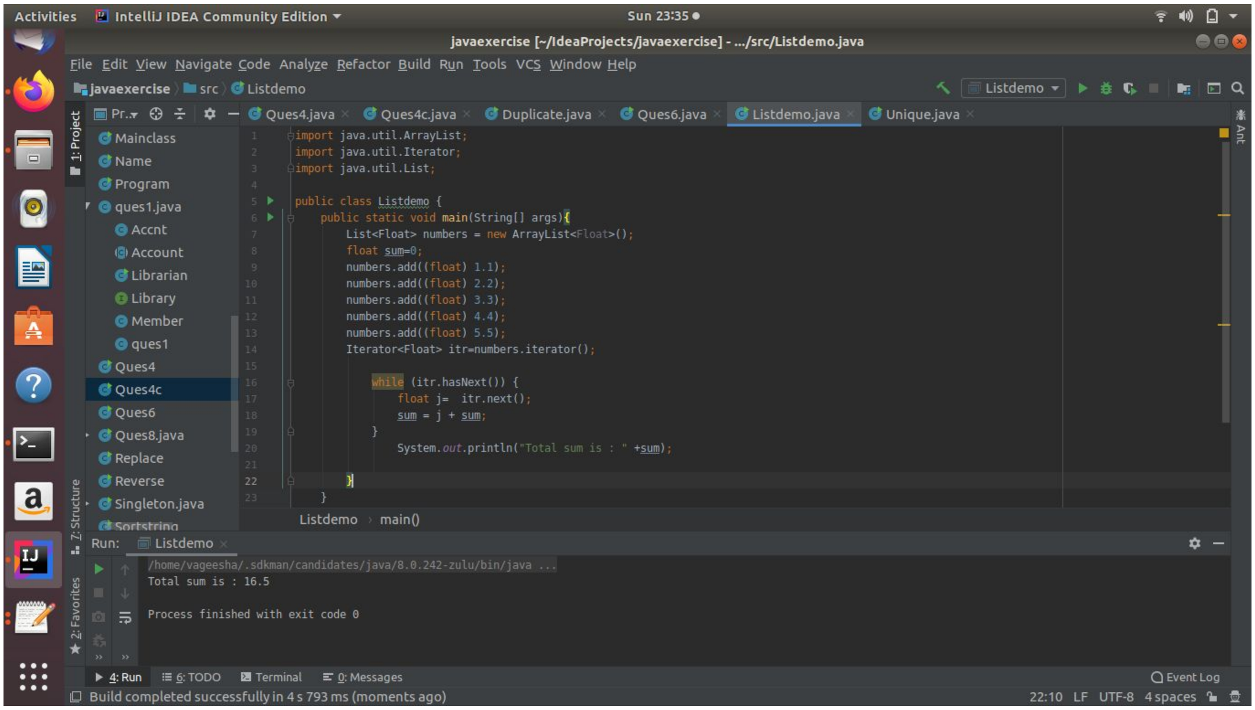
1. **Write Java code to define List . Insert 5 floating point numbers in List, and using an iterator, find the sum of the numbers in List.**

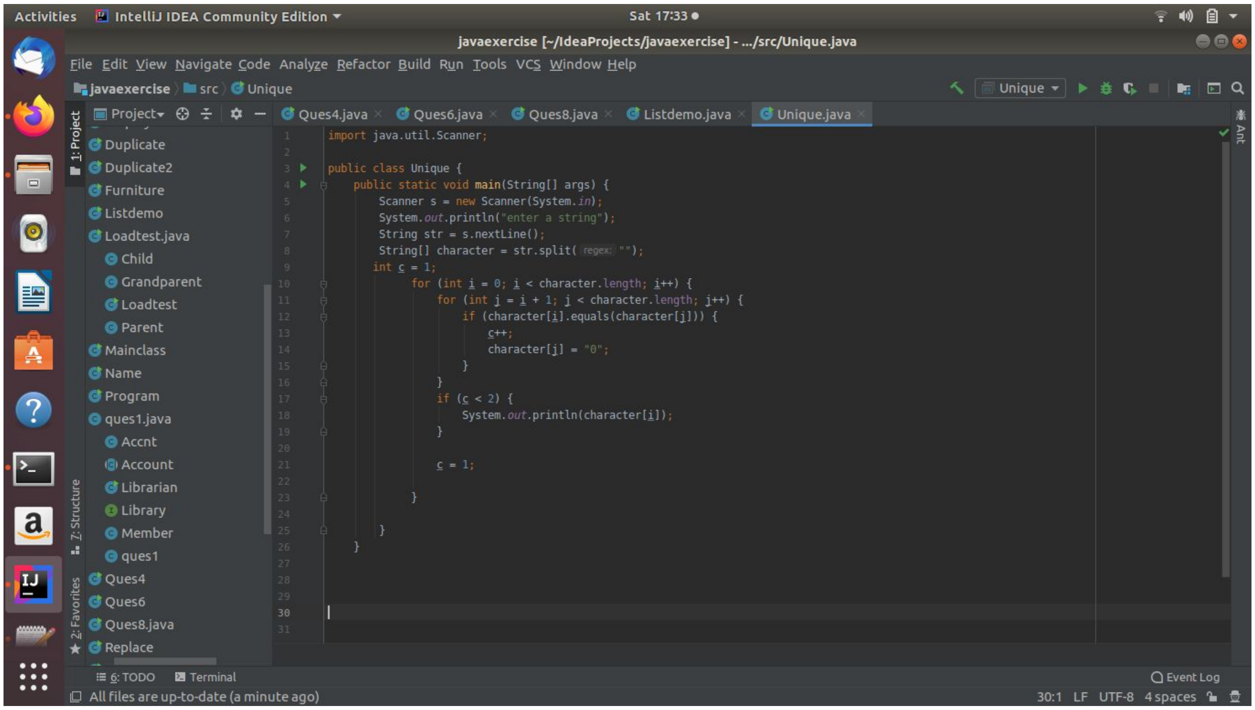

Activities
IntelliJ IDEA Community Edition
Sun 23:35
javaexercise [~/IdeaProjects/javaexercise] - .../src/Listdemo.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
javaexercise 〉 src 〉 Listdemo
Ques4.java Ques4c.java Duplicate.java Ques6.java Listdemo.java Unique.java
Mainclass
Name
Program
ques1.java
Accnt
Account
Librarian
Library
Member
ques1
Ques4
Ques4c
Ques6
Ques8.java
Replace
Reverse
Singleton.java
Sortstring
Listdemo 〉 main()
Run: Listdemo
4: Run
6: TODO
Terminal
0: Messages
Event Log
22:10 LF UTF-8 4 spaces

```java
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

public class Listdemo {
    public static void main(String[] args){
        List<Float> numbers = new ArrayList<Float>();
        float sum=0;
        numbers.add((float) 1.1);
        numbers.add((float) 2.2);
        numbers.add((float) 3.3);
        numbers.add((float) 4.4);
        numbers.add((float) 5.5);
        Iterator<Float> itr=numbers.iterator();

          while (itr.hasNext()) {
              float j=  itr.next();
              sum = j + sum;
          }
              System.out.println("Total sum is : " +sum);

        }
    }
```

```
/home/vageesha/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Total sum is : 16.5

Process finished with exit code 0
```

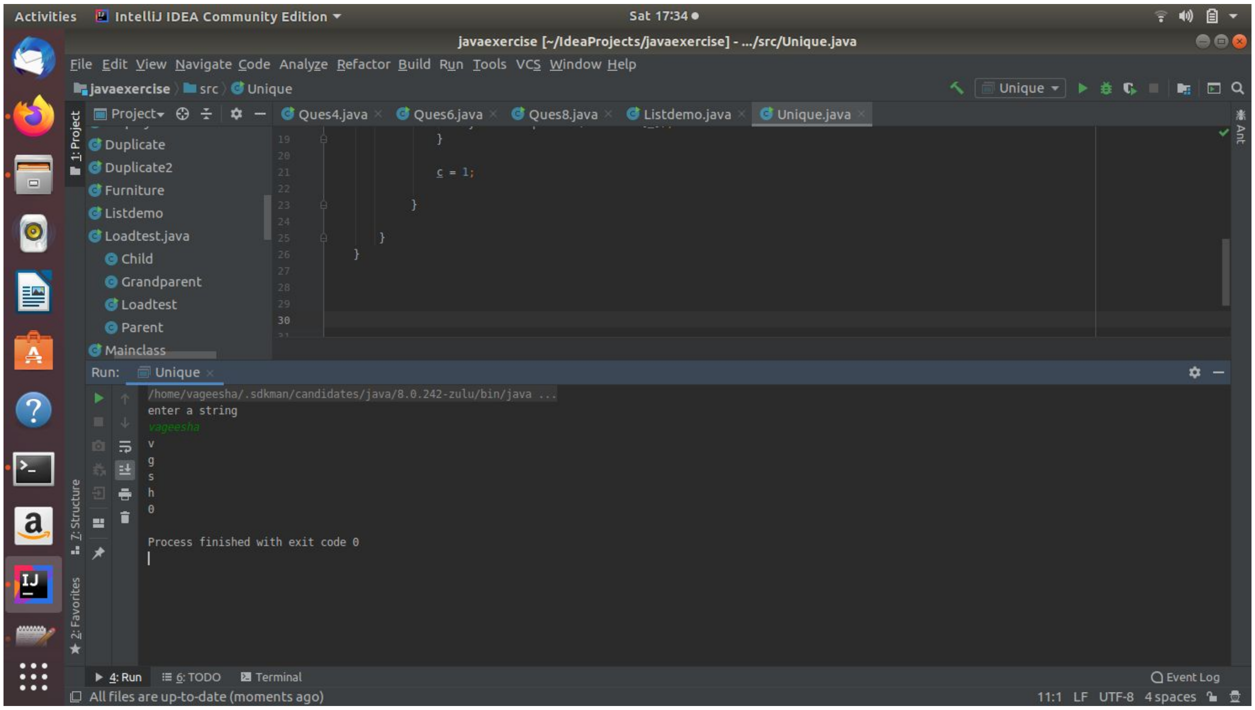
Build completed successfully in 4 s 793 ms (moments ago)

2. **Write a method that takes a string and returns the number of unique characters in the string.**


Activities
IntelliJ IDEA Community Edition
Sat 17:33
javaexercise [~/IdeaProjects/javaexercise] - .../src/Unique.java
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
javaexercise 〉 src 〉 Unique
Ques4.java Ques6.java Ques8.java Listdemo.java Unique.java
Project
Duplicate
Duplicate2
Furniture
Listdemo
Loadtest.java
Child
Grandparent
Loadtest
Parent
Mainclass
Name
Program
ques1.java
Accnt
Account
Librarian
Library
Member
ques1
Ques4
Ques6
Ques8.java
Replace
6: TODO
Terminal
Event Log
30:1 LF UTF-8 4 spaces

```java
import java.util.Scanner;

public class Unique {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        System.out.println("enter a string");
        String str = s.nextLine();
        String[] character = str.split( regex: "");
       int c = 1;
          for (int i = 0; i < character.length; i++) {
              for (int j = i + 1; j < character.length; j++) {
                  if (character[i].equals(character[j])) {
                      c++;
                      character[j] = "0";
                  }
              }
              if (c < 2) {
                  System.out.println(character[i]);
              }

              c = 1;

          }

       }
    }
```

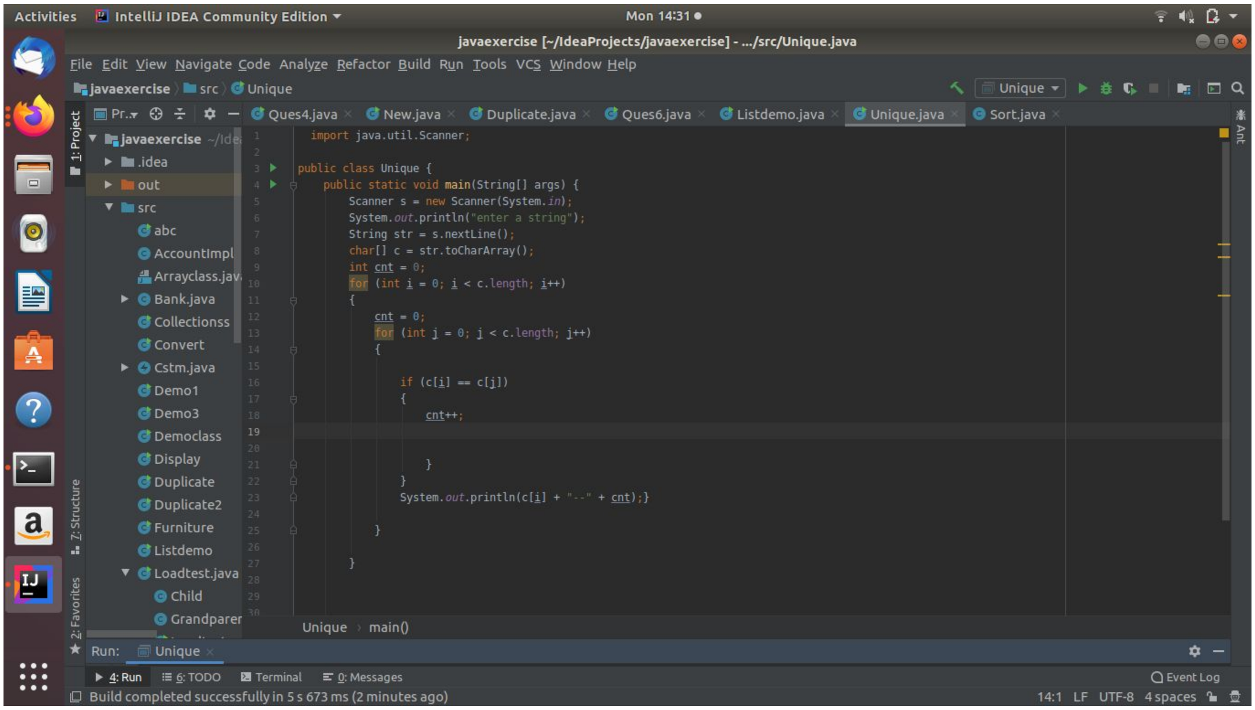
All files are up-to-date (a minute ago)



```
                    }

                    c = 1;

                }

            }
        }
```

```
/home/vageesha/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
enter a string
vageesha
v
g
s
h
0

Process finished with exit code 0
```

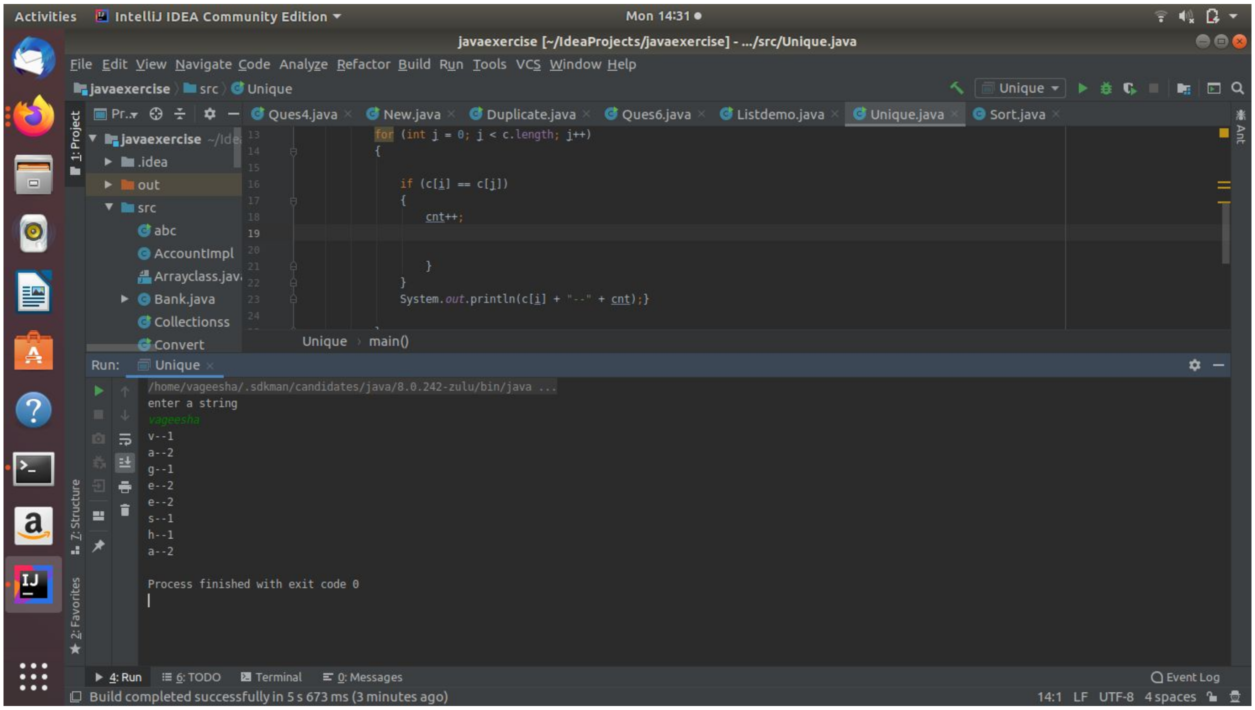
3. **Write a method that takes a string and print the number of occurrence of each characters in the string.**



javaexercise [~/IdeaProjects/javaexercise] - .../src/Unique.java

```java
 import java.util.Scanner;

public class Unique {
   public static void main(String[] args) {
       Scanner s = new Scanner(System.in);
       System.out.println("enter a string");
       String str = s.nextLine();
       char[] c = str.toCharArray();
       int cnt = 0;
       for (int i = 0; i < c.length; i++)
       {
           cnt = 0;
           for (int j = 0; j < c.length; j++)
           {

               if (c[i] == c[j])
               {
                   cnt++;


               }
           }
           System.out.println(c[i] + "--" + cnt);}

       }

   }
```

Build completed successfully in 5 s 673 ms (2 minutes ago)



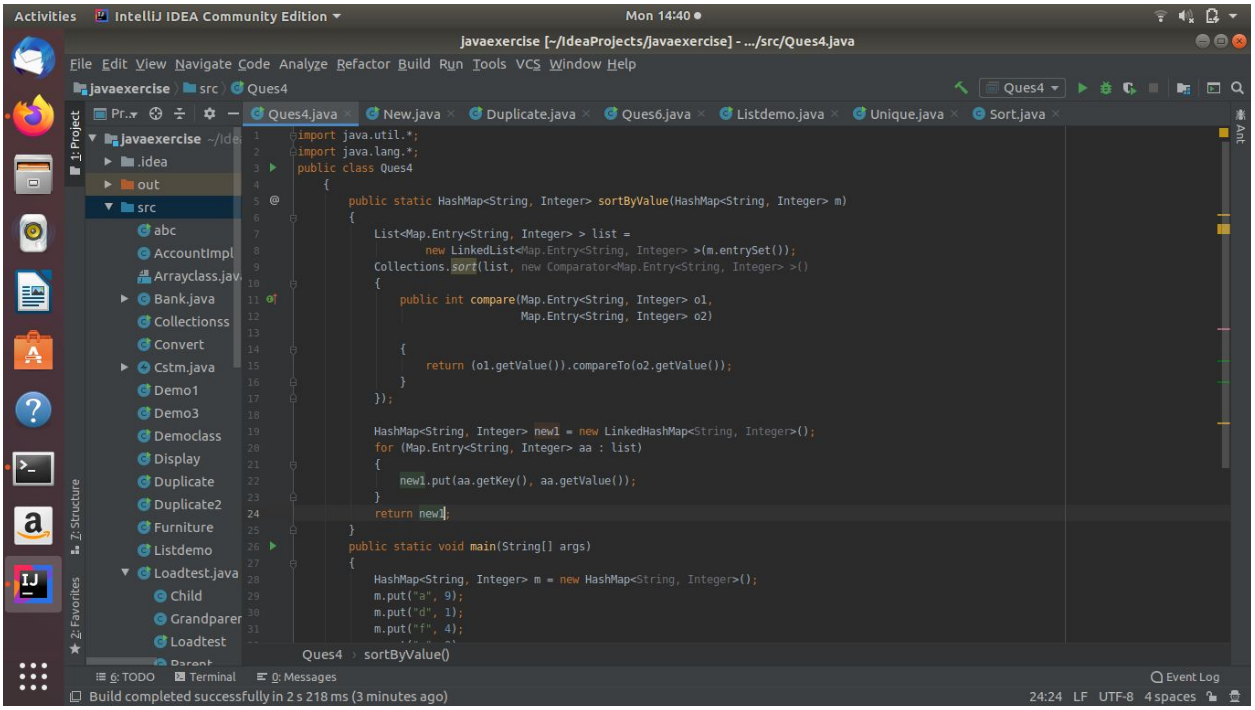
javaexercise [~/IdeaProjects/javaexercise] - .../src/Unique.java

```java
           for (int j = 0; j < c.length; j++)
           {

               if (c[i] == c[j])
               {
                   cnt++;


               }
           }
           System.out.println(c[i] + "--" + cnt);}
```

```
/home/vageesha/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
enter a string
vageesha
v--1
a--2
g--1
e--2
e--2
s--1
h--1
a--2

Process finished with exit code 0
```

Build completed successfully in 5 s 673 ms (3 minutes ago)

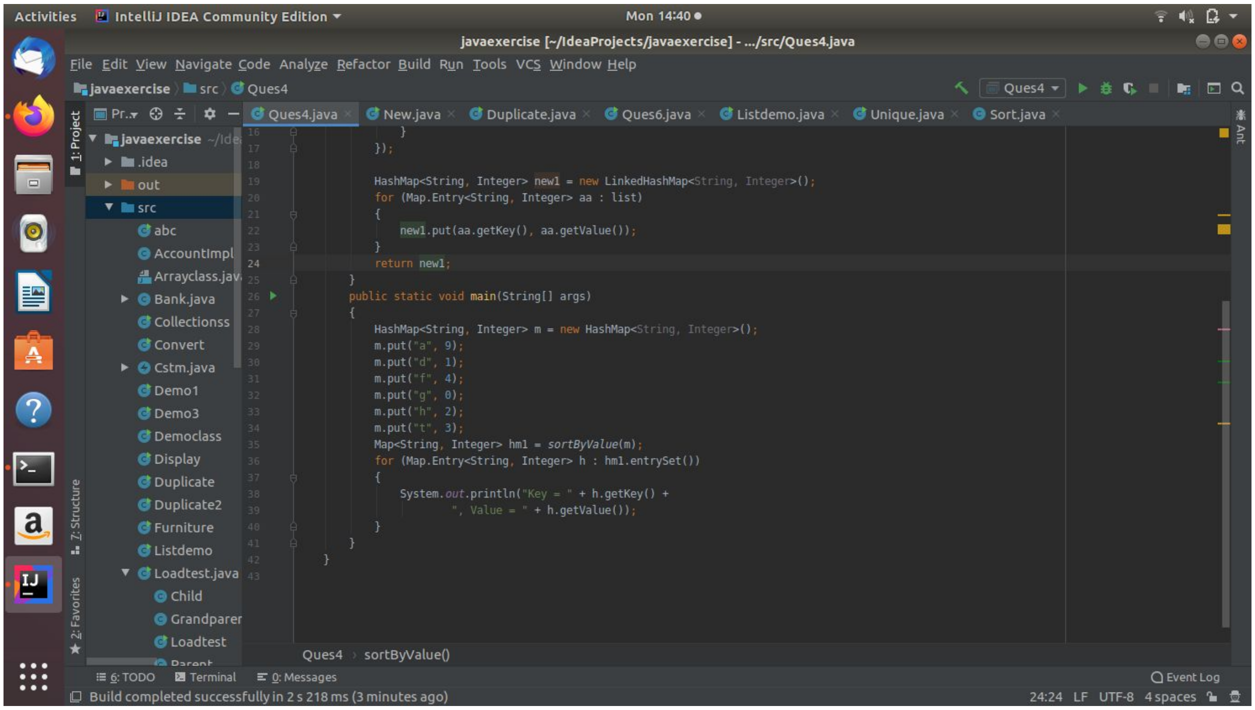
4. Write a program to sort HashMap by value.



javaexercise [~/IdeaProjects/javaexercise] - .../src/Ques4.java

```java
import java.util.*;
import java.lang.*;
public class Ques4
    {
        public static HashMap<String, Integer> sortByValue(HashMap<String, Integer> m)
        {
            List<Map.Entry<String, Integer> > list =
                    new LinkedList<Map.Entry<String, Integer> >(m.entrySet());
            Collections.sort(list, new Comparator<Map.Entry<String, Integer> >()
            {
                public int compare(Map.Entry<String, Integer> o1,
                                   Map.Entry<String, Integer> o2)

                {
                    return (o1.getValue()).compareTo(o2.getValue());
                }
            });

            HashMap<String, Integer> new1 = new LinkedHashMap<String, Integer>();
            for (Map.Entry<String, Integer> aa : list)
            {
                new1.put(aa.getKey(), aa.getValue());
            }
            return new1;
        }
        public static void main(String[] args)
        {
            HashMap<String, Integer> m = new HashMap<String, Integer>();
            m.put("a", 9);
            m.put("d", 1);
            m.put("f", 4);
```

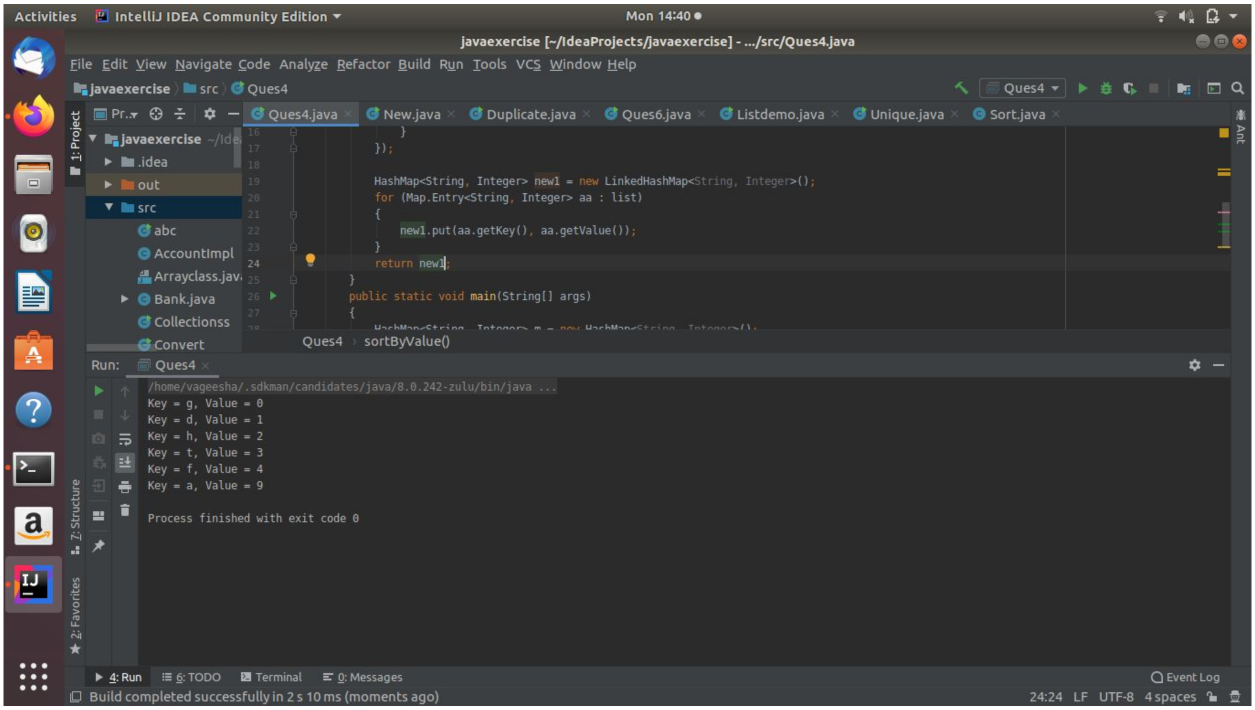
Build completed successfully in 2 s 218 ms (3 minutes ago)



javaexercise [~/IdeaProjects/javaexercise] - .../src/Ques4.java

```java
                }
            });

            HashMap<String, Integer> new1 = new LinkedHashMap<String, Integer>();
            for (Map.Entry<String, Integer> aa : list)
            {
                new1.put(aa.getKey(), aa.getValue());
            }
            return new1;
        }
        public static void main(String[] args)
        {
            HashMap<String, Integer> m = new HashMap<String, Integer>();
            m.put("a", 9);
            m.put("d", 1);
            m.put("f", 4);
            m.put("g", 0);
            m.put("h", 2);
            m.put("t", 3);
            Map<String, Integer> hm1 = sortByValue(m);
            for (Map.Entry<String, Integer> h : hm1.entrySet())
            {
                System.out.println("Key = " + h.getKey() +
                        ", Value = " + h.getValue());
            }
        }
    }
```

Build completed successfully in 2 s 218 ms (3 minutes ago)

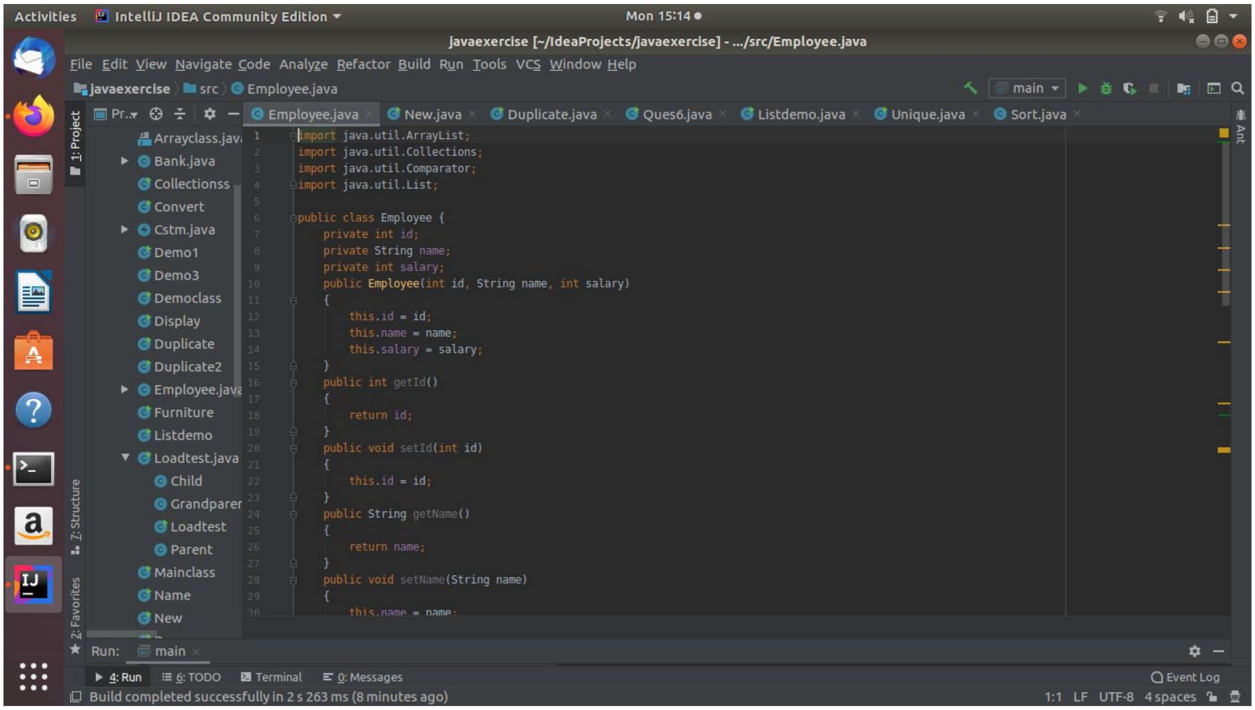


```
            }
        });

        HashMap<String, Integer> new1 = new LinkedHashMap<String, Integer>();
        for (Map.Entry<String, Integer> aa : list)
        {
            new1.put(aa.getKey(), aa.getValue());
        }
        return new1;
    }
    public static void main(String[] args)
    {
        HashMap<String, Integer> m = new HashMap<String, Integer>();
```

```
/home/vageesha/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
Key = g, Value = 0
Key = d, Value = 1
Key = h, Value = 2
Key = t, Value = 3
Key = f, Value = 4
Key = a, Value = 9

Process finished with exit code 0
```

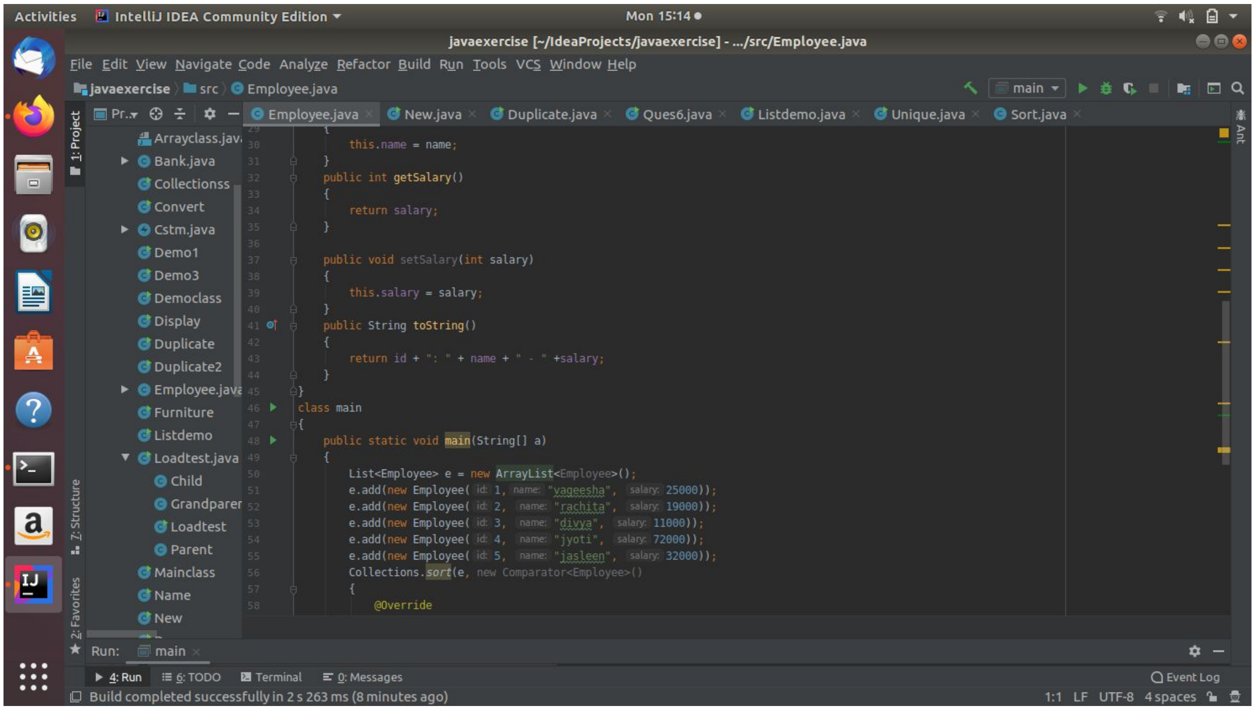
Build completed successfully in 2 s 10 ms (moments ago)

5. **Write a program to sort Employee objects based on highest salary using Comparator. Employee class{ Double Age; Double Salary; String Name**



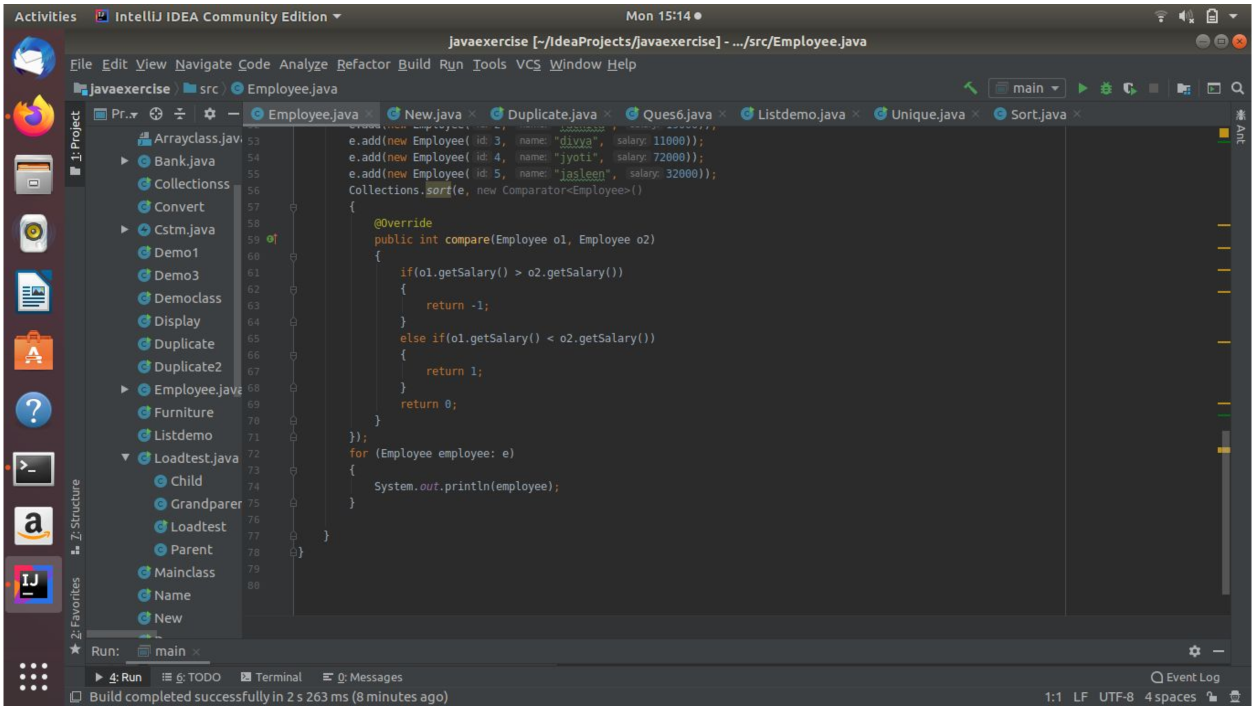
```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;

public class Employee {
    private int id;
    private String name;
    private int salary;
    public Employee(int id, String name, int salary)
    {
        this.id = id;
        this.name = name;
        this.salary = salary;
    }
    public int getId()
    {
        return id;
    }
    public void setId(int id)
    {
        this.id = id;
    }
    public String getName()
    {
        return name;
    }
    public void setName(String name)
    {
        this.name = name;
```

Build completed successfully in 2 s 263 ms (8 minutes ago)



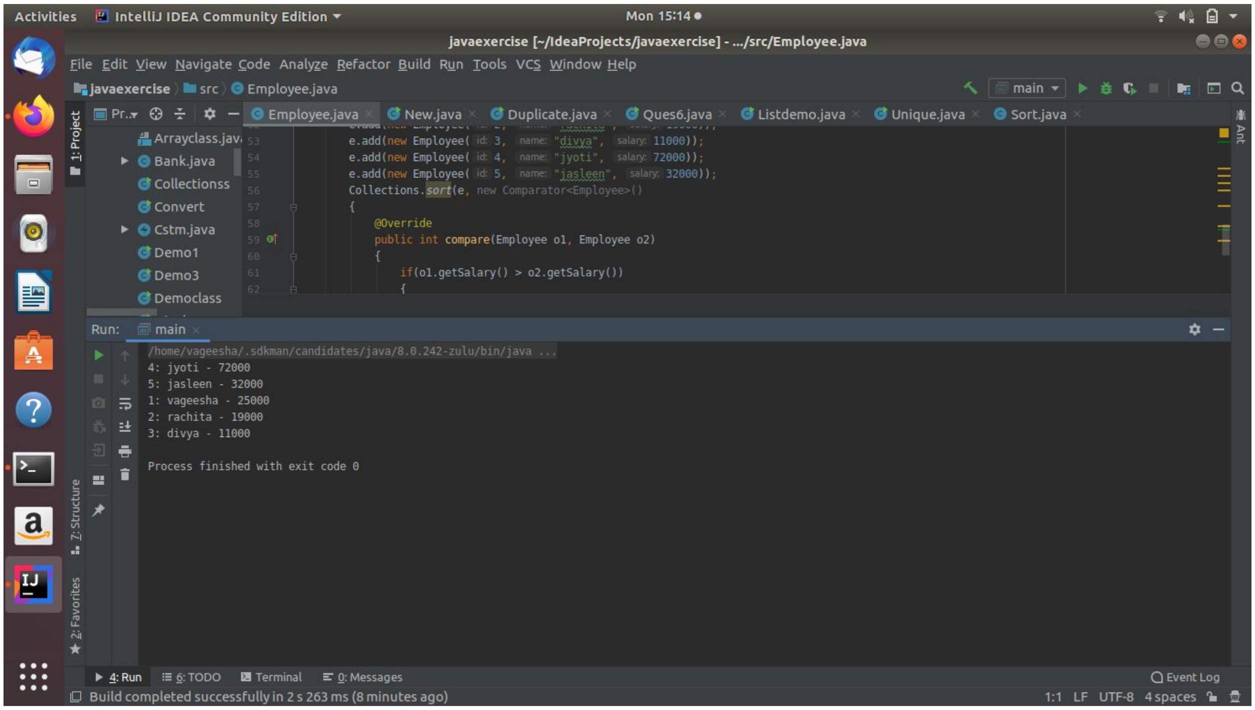
```
        {
            this.name = name;
        }
        public int getSalary()
        {
            return salary;
        }

        public void setSalary(int salary)
        {
            this.salary = salary;
        }
        public String toString()
        {
            return id + ": " + name + " - " +salary;
        }
}
class main
{
    public static void main(String[] a)
    {
        List<Employee> e = new ArrayList<Employee>();
        e.add(new Employee( id: 1,  name: "vageesha",  salary: 25000));
        e.add(new Employee( id: 2,  name: "rachita",  salary: 19000));
        e.add(new Employee( id: 3,  name: "divya",  salary: 11000));
        e.add(new Employee( id: 4,  name: "jyoti",  salary: 72000));
        e.add(new Employee( id: 5,  name: "jasleen",  salary: 32000));
        Collections.sort(e, new Comparator<Employee>()
        {
            @Override
```

Build completed successfully in 2 s 263 ms (8 minutes ago)



```
        e.add(new Employee( id: 3,  name: "divya",  salary: 11000));
        e.add(new Employee( id: 4,  name: "jyoti",  salary: 72000));
        e.add(new Employee( id: 5,  name: "jasleen",  salary: 32000));
        Collections.sort(e, new Comparator<Employee>()
        {
            @Override
            public int compare(Employee o1, Employee o2)
            {
                if(o1.getSalary() > o2.getSalary())
                {
                    return -1;
                }
                else if(o1.getSalary() < o2.getSalary())
                {
                    return 1;
                }
                return 0;
            }
        });
        for (Employee employee: e)
        {
            System.out.println(employee);
        }

    }
}
```

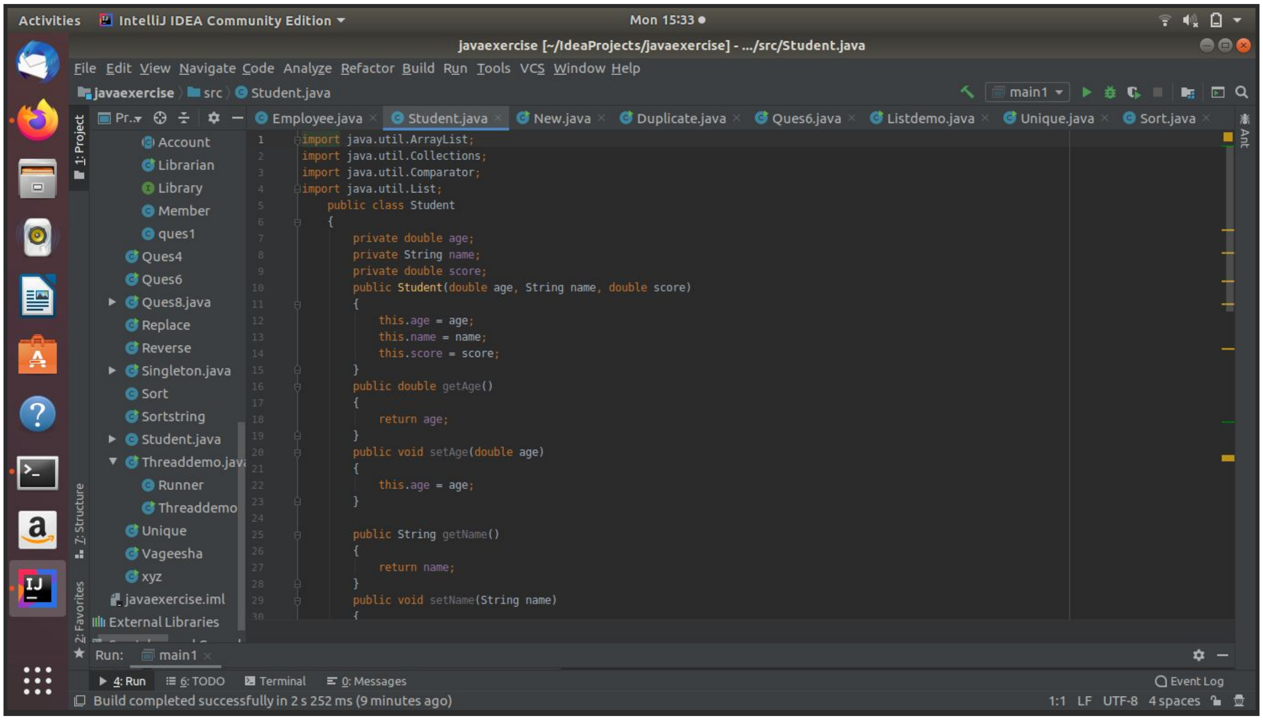
Build completed successfully in 2 s 263 ms (8 minutes ago)


javaexercise [~/IdeaProjects/javaexercise] - .../src/Employee.java
Build completed successfully in 2 s 263 ms (8 minutes ago)

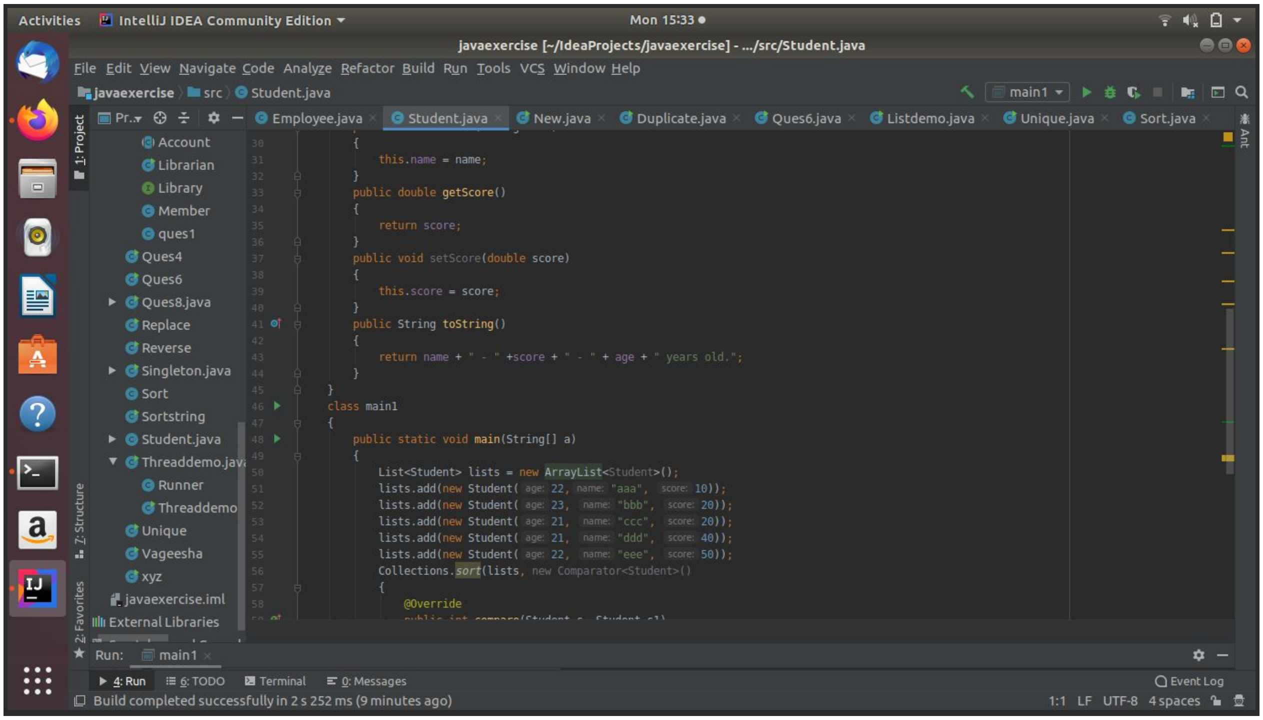
```
        e.add(new Employee( id: 3, name: "divya", salary: 11000));
        e.add(new Employee( id: 4, name: "jyoti", salary: 72000));
        e.add(new Employee( id: 5, name: "jasleen", salary: 32000));
        Collections.sort(e, new Comparator<Employee>()
        {
            @Override
            public int compare(Employee o1, Employee o2)
            {
                if(o1.getSalary() > o2.getSalary())
                {
```

```
/home/vageesha/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
4: jyoti - 72000
5: jasleen - 32000
1: vageesha - 25000
2: rachita - 19000
3: divya - 11000

Process finished with exit code 0
```

6. **Write a program to sort the Student objects based on Score , if the score are same then sort on First Name . Class Student{ String Name; Double Score; Double Age**


javaexercise [~/IdeaProjects/javaexercise] - .../src/Student.java
Build completed successfully in 2 s 252 ms (9 minutes ago)

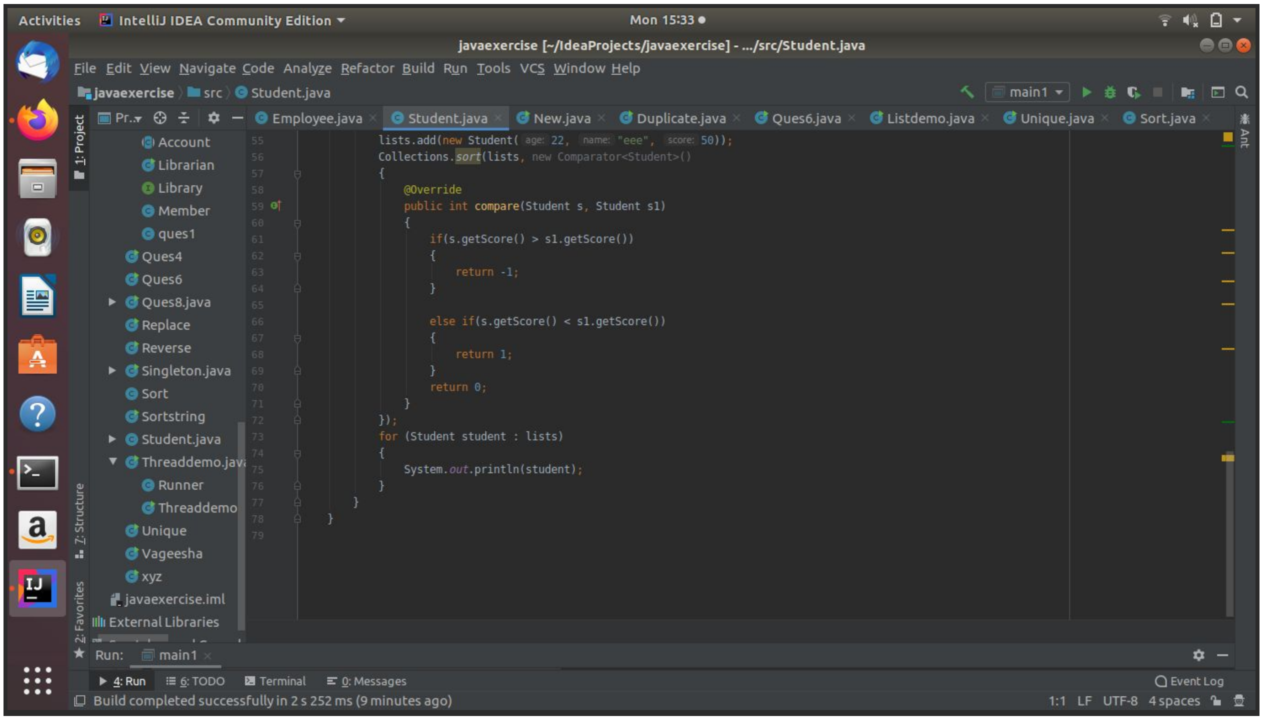
```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;
    public class Student
    {
        private double age;
        private String name;
        private double score;
        public Student(double age, String name, double score)
        {
            this.age = age;
            this.name = name;
            this.score = score;
        }
        public double getAge()
        {
            return age;
        }
        public void setAge(double age)
        {
            this.age = age;
        }

        public String getName()
        {
            return name;
        }
        public void setName(String name)
        {
```



javaexercise [~/IdeaProjects/javaexercise] - .../src/Student.java

```java
        {
            this.name = name;
        }
        public double getScore()
        {
            return score;
        }
        public void setScore(double score)
        {
            this.score = score;
        }
        public String toString()
        {
            return name + " - " +score + " - " + age + " years old.";
        }
    }
    class main1
    {
        public static void main(String[] a)
        {
            List<Student> lists = new ArrayList<Student>();
            lists.add(new Student( age: 22, name: "aaa", score: 10));
            lists.add(new Student( age: 23, name: "bbb", score: 20));
            lists.add(new Student( age: 21, name: "ccc", score: 20));
            lists.add(new Student( age: 21, name: "ddd", score: 40));
            lists.add(new Student( age: 22, name: "eee", score: 50));
            Collections.sort(lists, new Comparator<Student>()
            {
                @Override
```

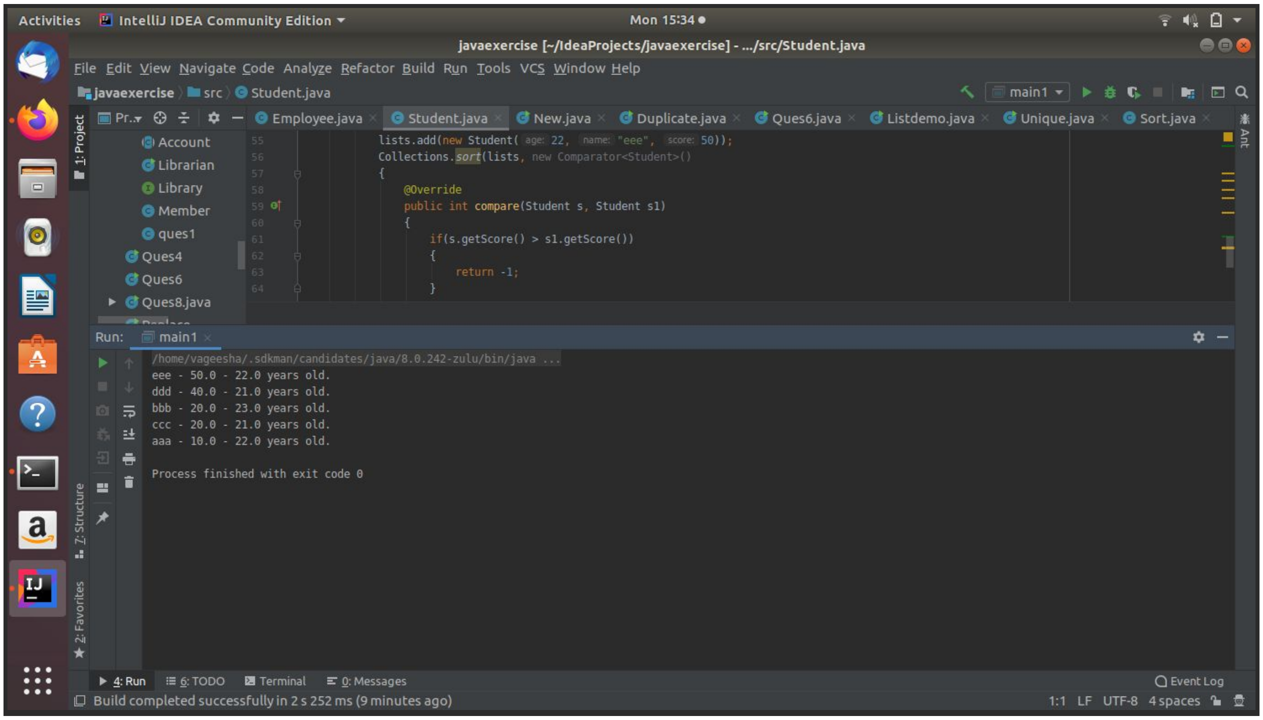
Build completed successfully in 2 s 252 ms (9 minutes ago)



javaexercise [~/IdeaProjects/javaexercise] - .../src/Student.java

```java
            lists.add(new Student( age: 22, name: "eee", score: 50));
            Collections.sort(lists, new Comparator<Student>()
            {
                @Override
                public int compare(Student s, Student s1)
                {
                    if(s.getScore() > s1.getScore())
                    {
                        return -1;
                    }

                    else if(s.getScore() < s1.getScore())
                    {
                        return 1;
                    }
                    return 0;
                }
            });
            for (Student student : lists)
            {
                System.out.println(student);
            }
        }
    }
```

Build completed successfully in 2 s 252 ms (9 minutes ago)



```
        lists.add(new Student( age: 22,  name: "eee",  score: 50));
        Collections.sort(lists, new Comparator<Student>()
        {
            @Override
            public int compare(Student s, Student s1)
            {
                if(s.getScore() > s1.getScore())
                {
                    return -1;
                }
```

```
/home/vageesha/.sdkman/candidates/java/8.0.242-zulu/bin/java ...
eee - 50.0 - 22.0 years old.
ddd - 40.0 - 21.0 years old.
bbb - 20.0 - 23.0 years old.
ccc - 20.0 - 21.0 years old.
aaa - 10.0 - 22.0 years old.

Process finished with exit code 0
```

Build completed successfully in 2 s 252 ms (9 minutes ago)

7. **Print the elements of an array in the decreasing frequency if 2 numbers have same frequency then print the one which came first.**

8. **Design a Data Structure SpecialStack that supports all the stack operations like push(), pop(), isEmpty(), isFull() and an additional operation getMin() which should return minimum element from the SpecialStack. (Expected complexity O(1))**