Assignment 2

CSE 373

Design and Analysis of Algorithms

Section 9

Spring 2020

North South University

Submitted To: Shaikh Shawon Arefin Shimon (SAS3)

Name	:	Sams Uddin Ahamed
Student ID	:	1620129
Email Address	:	Sams.ahamed@northsouth.edu
Submission Date	:	5 March 2020

1. Knapsack Class:

```
<terminated> knapsack [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (Mar 5, 2020, 9:40:49 PM)
Bars that taken with the price and weight:
JAR_158 107 1.4927067
JAR_389 100 1.6084536
JAR_525 104 1.9564586
JAR_547 69 1.3585281
JAR_236 59 1.3473155
JAR_124 61 1.7789774
JAR_57 119 3.8520856
JAR_3 116 3.9883077
JAR_488 84 2.9475205
JAR_332 120 4.4265957
JAR_91 87 3.2555377
JAR_352 91 3.4154954
JAR_244 46 1.7710621
JAR_464 58 2.2768962
JAR_191 86 3.390126
JAR_430 50 2.1146166
JAR_34 106 4.5195284
JAR_38 110 5.2503347
JAR_357 110 5.3142376
JAR_418 111.63416 6.144497
==================
Total knapsack Value: 1794.6342
Total Weight of the knapsack: 62.20928
```

Figure 1: Outputs

```
package com.nsu.cse373.spring2020.ID1620129;

import java.util.List;
import java.io.BufferedReader;

import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.Arrays;
import java.util.Collections;
import java.util.Comparator;

public abstract class knapsack implements Comparator<String>{

public static void main(String[] args) throws IOException {

     String arr[] = new String[50000];

     File FileDes = new File("src\\com\\nsu\\cse373\\spring2020\\ID1620129\\data.txt");
     BufferedReader br = new BufferedReader(new FileReader(FileDes));

     String st;

     int i =0;
     while ((st = br.readLine()) != null) {

         arr[i] = st;
         i++;
     }

    String s = arr[0].trim();
```

```
  String[] words = s.split(" ");
  String firstWord = words[0];
  String lastWord = words[words.length - 1];
  int numberOfjars = Integer.valueOf(firstWord)+1;
  float knapsackSize = Float.valueOf(lastWord);


  List<String> nameList = Arrays.asList(arr);
Collections.sort(nameList.subList(1, numberOfjars), new Comparator<String>() {

  @Override
  public int compare(String arg0, String arg1) { //Custom comparator
     String fprice0 = arg0.split(" ")[1];
     String fprice1 = arg1.split(" ")[1];
     String fsize0 = arg0.split(" ")[2];
     String fsize1 = arg1.split(" ")[2];

     Int f = Integer.valueOf(fprice0);
     int f1 = Integer.valueOf(fprice1);
     float size1 = Float.parseFloat(fsize0);
     float size2 = Float.parseFloat(fsize1);

     Float unit = f/size1;
     Float unit1 = f1/size2;
     int m = Float.compare(unit, unit1);


  return m;
  }
});




   float sum=0;
   int l = numberOfjars-1;
   int count=0;
   String ItemsTaken[] = new String[10000];

   int iTaken = 0;

   while(sum<=knapsackSize) {  //Pushing Jars in to the knapsack Jar

   String temp = arr[l].trim();
   String[] wordsTemp = temp.split(" ");
   String jsizeTemp = wordsTemp[2];
   String jpriceTemp = wordsTemp[1];
   float jsizeFloatTemp = Float.valueOf(jsizeTemp); //Getting the JarSize
   int jprice = Integer.valueOf(jpriceTemp); //Getting the JarPrice


  if(sum>jsizeFloatTemp) {

     float need = knapsackSize-sum;

    if(need<=jsizeFloatTemp) {
      sum = sum+need;
      ItemsTaken[iTaken] = temp;
      Float unitP1 = jprice/jsizeFloatTemp;
      Float takenPrice = need*unitP1;
```

```
      ItemsTaken[iTaken] = ItemsTaken[iTaken].replace(jpriceTemp, Float.toString(takenPrice));
      ItemsTaken[iTaken] = ItemsTaken[iTaken].replace(jsizeTemp, Float.toString(need));

       iTaken++;

  }

else {
     sum = sum+jsizeFloatTemp;
     ItemsTaken[iTaken] = temp;
  ItemsTaken[iTaken] =  ItemsTaken[iTaken].replace(jsizeTemp, Float.toString(jsizeFloatTemp));

     iTaken++;

}


      count++;

}

if(sum<=jsizeFloatTemp) {

        float need1 = knapsackSize-sum;

       if(jsizeFloatTemp<=need1) {
         sum = sum + jsizeFloatTemp;

      ItemsTaken[iTaken] = temp;
   ItemsTaken[iTaken] = ItemsTaken[iTaken].replace(jsizeTemp, Float.toString(jsizeFloatTemp));

       iTaken++;

}

else {
        sum = sum+need1;

        ItemsTaken[iTaken] = temp;
        Float unitP = jprice/jsizeFloatTemp;
        Float takenPrice = need1*unitP;

      ItemsTaken[iTaken] = ItemsTaken[iTaken].replace(jpriceTemp, Float.toString(takenPrice));
      ItemsTaken[iTaken] = ItemsTaken[iTaken].replace(jsizeTemp, Float.toString(need1));

      iTaken++;

}



      count++;


}

 if(sum==knapsackSize) {
      break;
 }

l--;
```

```
}
```

```
  System.out.println("Jars that taken with the price and weight: ");
    for(int m=0;m<count;m++) {
       System.out.println(ItemsTaken[m]);
    }

  System.out.println("==================");

    float knapsackPriceTotal = 0;
    for(int o=0;o<count;o++) {

       String temp = ItemsTaken[o].trim();
       String[] wordsTemp = temp.split(" ");
       String priceS = wordsTemp[1];
       Float price = Float.valueOf(priceS);
       knapsackPriceTotal = knapsackPriceTotal + price;
     }

   System.out.println("Total knapsack Value: "+knapsackPriceTotal);
   System.out.println("Total Weight of the knapsack: "+sum+"\n");

}


}
```

2. CreatingData Class:

```
package com.nsu.cse373.spring2020.ID1620129;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;

class CreatingData {
//Run once If there's no Data.txt file

  public static void main(String[] args) {
// TODO Auto-generated method stub


    int NumberOfJar =(int) getRandomIntegerBetweenRange(200,1000);
    float knapsacksize = (float) getRandomIntegerBetweenRange(40,80);
    writingfirstLine(NumberOfJar,knapsacksize);


    for(int i=0;i<NumberOfJar;i++) {
       String jarName = "JAR_"+(i+1);
       String jarPrice = String.valueOf((int)getRandomIntegerBetweenRange(20,120));
       String jarSize= String.valueOf((float)getRandomIntegerBetweenRange(1.3,51.6));
       fileWriteForKnapsack("data.txt",jarName,jarPrice,jarSize);

   }
}




public static double getRandomIntegerBetweenRange(double min, double max){
       double x = (double)(Math.random()*((max-min)+1))+min;
       return x;
}


public static void writingfirstLine(int njars,float ksize) {


 try {
   File FileDes = new File("src\\com\\nsu\\cse373\\spring2020\\ID1620129\\data.txt");
   FileWriter writer = new FileWriter(FileDes, true);
   BufferedWriter bufferedWriter = new BufferedWriter(writer)

   bufferedWriter.write(String.valueOf(njars));
   bufferedWriter.write(" ");
   bufferedWriter.write(String.valueOf(ksize));
   bufferedWriter.newLine();
   bufferedWriter.close();
 } catch (IOException e) {
        e.printStackTrace();
  }


}
```

```
public static void fileWriteForKnapsack(String filename,String Jarname,String Jarprice,String
Jarsize) {

  try {
    File FileDes = new File("src\\com\\nsu\\cse373\\spring2020\\ID1620129\\"+filename);


    FileWriter writer = new FileWriter(FileDes, true);
    BufferedWriter bufferedWriter = new BufferedWriter(writer);

    bufferedWriter.write(Jarname);
    bufferedWriter.write(" ");
    bufferedWriter.write(Jarprice);
    bufferedWriter.write(" ");
    bufferedWriter.write(Jarsize);
    bufferedWriter.newLine();

    bufferedWriter.close();
  } catch (IOException e) {
      e.printStackTrace();
  }


}

}
```