```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

import mypackage.HeapAdaptablePriorityQueue;
import mypackage.Record;

public class Main {
   public static void main(String[] args){
      String filePath = System.getProperty("user.dir") + "/" + args[0];

      HeapAdaptablePriorityQueue<String, Double, Record> priQueue = new
HeapAdaptablePriorityQueue<>();

      priQueue = readTxtFileToQueue(filePath, priQueue);



      printMenu();
      while (true) {
         Scanner scan = new Scanner(System.in);
         int userIn = scan.nextInt();



         //1. Insert
         if(userIn == 1){
            System.out.println("Enter the name of the record to read");
            String fileName = scan.next();
```

```
    String path = System.getProperty("user.dir") + "/" + fileName;
    priQueue = readTxtFileToQueue(path, priQueue);
    System.out.println("Input file is read successfully");




}
// 2. Peek
else if(userIn == 2){
    Record min = priQueue.min().getValue();
    System.out.println("The patient detail with the highest priority is as follows:");
    System.out.println("Patient's first name: " + min.getFirstName());
    System.out.println("Patient's last name:" + min.getLastName());
    System.out.println("Date of birth of the patient: " + min.getDateOfBirth());
    System.out.println("Address: " + min.getAddress());
    System.out.println("City: " + min.getCity());
    System.out.println("County: " + min.getCounty());
    System.out.println("State: " + min.getState());
    System.out.println("Zip code: " + min.getZip());
    System.out.println("Phone Number (1st Preference): " + min.getPhone1());
    System.out.println("Phone Number (1st Preference): " + min.getPhone2());
    System.out.println("Email address: " + min.getEmail());
    System.out.println("UNOS Status: " + min.getUnosStatus());
    System.out.println("Date listed on " + min.getUnosStatus() + ": " + min.getDateListed());
    min.printUnosHistory();
}

// 3. nextPatient
else if(userIn == 3){
```

```
    Record min = priQueue.removeMin().getValue();
    System.out.println("The patient removed from the heap is as follows:");
    System.out.println("Patient's first name: " + min.getFirstName());
    System.out.println("Patient's last name:" + min.getLastName());
    System.out.println("Date of birth of the patient: " + min.getDateOfBirth());
    System.out.println("Address: " + min.getAddress());
    System.out.println("City: " + min.getCity());
    System.out.println("County: " + min.getCounty());
    System.out.println("State: " + min.getState());
    System.out.println("Zip code: " + min.getZip());
    System.out.println("Phone Number (1st Preference): " + min.getPhone1());
    System.out.println("Phone Number (1st Preference): " + min.getPhone2());
    System.out.println("Email address: " + min.getEmail());
    System.out.println("UNOS Status: " + min.getUnosStatus());
    System.out.println("Date listed on " + min.getUnosStatus() + ": " + min.getDateListed());
    min.printUnosHistory();
}

// 4. removePaient
else if(userIn == 4){
    Scanner scan2 = new Scanner(System.in);
    String firstName;
    String lastName;
    String dob;
    String address;
    String city;
    String county;
    String state;
    String zip;
```

```
String phone1;
String phone2;
String email;
String unosStatus;

 System.out.println("Please enter the patient information to remove from the queue: ");
 System.out.print("Please enter the patient's first name: ");
 firstName = scan2.nextLine();
 System.out.print("Please enter the patient's last name: ");
 lastName = scan2.nextLine();
 System.out.print("Please enter the patient's date of birth: ");
 dob = scan2.nextLine();
 System.out.print("Please enter the patient's address: ");
 address = scan2.nextLine();
 System.out.print("Please enter the patient's city: ");
 city = scan2.nextLine();
 System.out.print("Please enter the patient's county: ");
 county = scan2.nextLine();
 System.out.print("Please enter the patient's state: ");
 state = scan2.nextLine();
 System.out.print("Please enter the patient's zip code: ");
 zip = scan2.nextLine();
 System.out.print("Please enter the patient's phone number (1st Preference): ");
 phone1 = scan2.nextLine();
 System.out.print("Please enter the patient's phone number (2nd Preference): ");
 phone2 = scan2.nextLine();
 System.out.print("Please enter the patient's email address: ");
 email = scan2.nextLine();
 System.out.print("Please enter the patient's UNOS Status: ");
```

```
            unosStatus = scan2 .nextLine();


            HeapAdaptablePriorityQueue<String, Double, Record> temp = new
HeapAdaptablePriorityQueue<>();
            boolean patientFound = false;
            while(priQueue.size() > 0){
               Record value = new Record();
               value = priQueue.min().getValue();


               if(!value.getFirstName().equalsIgnoreCase(firstName) &&
!value.getLastName().equalsIgnoreCase(lastName)){
                  temp.insert(priQueue.min().getKey1(), priQueue.min().getKey2(), value);
               }
               else{
                  patientFound = true;
               }
               priQueue.removeMin();


            }
            priQueue = temp;


            if(!patientFound){
               System.out.println("Patient not found.");
            }
            else{
               System.out.println("Patient removed successfully!");
            }


         }
```

```
// 5. size
else if(userIn == 5){
   int size = priQueue.size();
   System.out.println("Number of records in the database: " + size);
}

// 6. updatePriority
else if(userIn == 6){
   Scanner scan3 = new Scanner(System.in);
   String firstName;
   String lastName;
   String dob;
   String address;
   String city;
   String county;
   String state;
   String zip;
   String phone1;
   String phone2;
   String email;
   String unosStatus;

    System.out.println("Please enter the patient information to change UNOS status: ");
    System.out.print("Please enter the patient's first name: ");
    firstName = scan3.nextLine();
    System.out.print("Please enter the patient's last name: ");
    lastName = scan3.nextLine();
    System.out.print("Please enter the patient's date of birth: ");
```

```
            dob = scan3.nextLine();
            System.out.print("Please enter the patient's address: ");
            address = scan3.nextLine();
            System.out.print("Please enter the patient's city: ");
            city = scan3.nextLine();
            System.out.print("Please enter the patient's county: ");
            county = scan3.nextLine();
            System.out.print("Please enter the patient's state: ");
            state = scan3.nextLine();
            System.out.print("Please enter the patient's zip code: ");
            zip = scan3.nextLine();
            System.out.print("Please enter the patient's phone number (1st Preference): ");
            phone1 = scan3.nextLine();
            System.out.print("Please enter the patient's phone number (2nd Preference): ");
            phone2 = scan3.nextLine();
            System.out.print("Please enter the patient's email address: ");
            email = scan3.nextLine();
            System.out.print("Please update the UNOS Status: ");
            unosStatus = scan3.nextLine();

            HeapAdaptablePriorityQueue<String, Double, Record> temp = new
HeapAdaptablePriorityQueue<>();
           boolean patientFound = false;
            while(priQueue.size() > 0){
              Record value = new Record();
              value = priQueue.min().getValue();

              if(!value.getFirstName().equalsIgnoreCase(firstName) &&
!value.getLastName().equalsIgnoreCase(lastName)){
```

```
        temp.insert(priQueue.min().getKey1(), priQueue.min().getKey2(),value);
    }
    else{
        patientFound = true;
        value.setDateListed();
        value.setUnosStatus(unosStatus);
        temp.insert(unosStatus, priQueue.min().getKey2(), value);
        System.out.println("The following patient detail has been updated:");
        System.out.println("Patient's first name: " + value.getFirstName());
        System.out.println("Patient's last name:" + value.getLastName());
        System.out.println("Date of birth of the patient: " + value.getDateOfBirth());
        System.out.println("Address: " + value.getAddress());
        System.out.println("City: " + value.getCity());
        System.out.println("County: " + value.getCounty());
        System.out.println("State: " + value.getState());
        System.out.println("Zip code: " + value.getZip());
        System.out.println("Phone Number (1st Preference): " + value.getPhone1());
        System.out.println("Phone Number (1st Preference): " + value.getPhone2());
        System.out.println("Email address: " + value.getEmail());
        System.out.println("UNOS Status: " + value.getUnosStatus());
        System.out.println("Date listed on " + value.getUnosStatus() + ": " + value.getDateListed());
        value.printUnosHistory();
    }
    priQueue.removeMin();
}
priQueue = temp;
if(!patientFound){
    System.out.println("Patient not found.");
}
```

```
        }
        // 7. exit
        else if(userIn == 7){
            scan.close();
            break;
        }

        else{

        }

    }




}


// Prints the option menu
public static void printMenu(){
    System.out.println("1.  insert");
    System.out.println("2.  peek");
    System.out.println("3.  nextPatient");
    System.out.println("4.  removePatient");
    System.out.println("5.  size");
    System.out.println("6.  updatePriority");
```

```
        System.out.println("7.  exit");
    }


    /*
     * Reads the given input text file into a priority queue
     *
     * @param filePath: The file path of the input file
     * @param list: the priority queue that the input file is being read into
     * @return list: the updated list
     */
    public static HeapAdaptablePriorityQueue<String, Double, Record> readTxtFileToQueue(String
filePath,
                                                HeapAdaptablePriorityQueue<String, Double, Record> list){


        try (BufferedReader br = new BufferedReader(new FileReader(filePath))) {
            String line;


            while ((line = br.readLine()) != null) {
                String[] values = line.split(";");
                if (values[0].equalsIgnoreCase("first_name")){}


                else{
                    // Split the line using semicolons as the separator
                    Record patientRecord = new Record(values[0], values[1], values[2], values[3], values[4],
values[5], values[6], values[7], values[8], values[9], values[10], values[11], values[12]);



                    list.insert(patientRecord.getUnosStatus(), patientRecord.getAge(), patientRecord);
```

```
                // System.out.println(list.min());
                // System.out.println(list.min().getValue().getFirstName());
            }



        }

    } catch (IOException e) {
        e.printStackTrace();
    }
  return list;
  }

}
```