

```

/*
    Application to provide Customer support from a direct access file
*/

import java.io.*;
import javax.swing.*;

public class CustomerDA
{
    static boolean fileExist = false;
    public String custNum, custNumber;
    public String firstName, middleName, lastName, custType, serType,
        lastMeterDate, lastMeterReading, currReading, currMo, currDy,
        currYr, currDate, outStanding;

    public String[] custNumRef = new String[200];

    private static int custCount = 0;
    boolean done = false;

    public CustomerDA()
    {
        File customerKeyIn = new File("customerKeyFile.txt");
        if (customerKeyIn.exists())
        {
            try
            {
                BufferedReader customerKey = new BufferedReader(new
                    FileReader(customerKeyIn));
                while (!done)
                {
                    custNumRef[custCount] = customerKey.readLine();
                    if (custNumRef[custCount] != null)
                    {
                        custCount++;
                    }
                    else
                    {
                        customerKey.close();
                    }
                }
            }
        }
    }
}

```

```

        }

        catch (IOException e)
        {
            done = true;
        }
    }
    else
    {
        System.out.println("File not found");
    }
}

public static int getCount()
{
    return custCount;
}

public String[] getCustomer(String custNum)
{
    String custData[] = new String[18];
    String foundStatus = "No";
    int recordLength = 164 * 2;
    int custLength = 0;
    String tempCustNum = "          " + custNum;
    custLength = tempCustNum.length();
    custNum = tempCustNum.substring(custLength - 6, custLength);
    try
    {
        RandomAccessFile CustomerDAFile = new RandomAccessFile(
            "CustomerDAFile.dat", "r");
        for(int i = 0; i < custCount; i++)
        {
            if (custNum.equals(custNumRef[i]))
            {
                int recordNumber = i;
                CustomerDAFile.seek(recordNumber * recordLength);
                int location = recordNumber * recordLength;

                custData[0] = parseField(CustomerDAFile, 6);
                custData[1] = parseField(CustomerDAFile, 15);
                custData[2] = parseField(CustomerDAFile, 15);
            }
        }
    }
}

```

```

        custData[3] = parseField(CustomerDAFile, 15);
        custData[4] = parseField(CustomerDAFile, 25);
        custData[5] = parseField(CustomerDAFile, 15);
        custData[6] = parseField(CustomerDAFile, 2);
        custData[7] = parseField(CustomerDAFile, 5);
        custData[8] = parseField(CustomerDAFile, 8);
        custData[9] = parseField(CustomerDAFile, 1);
        custData[10] = parseField(CustomerDAFile, 1);
        custData[11] = parseField(CustomerDAFile, 8);
        custData[12] = parseField(CustomerDAFile, 6);
        custData[13] = parseField(CustomerDAFile, 8);
        custData[14] = parseField(CustomerDAFile, 6);
        custData[15] = parseField(CustomerDAFile, 10);
        custData[16] = parseField(CustomerDAFile, 10);
        custData[17] = parseField(CustomerDAFile, 8);
        foundStatus = "Yes";
        CustomerDAFile.close();
    }
}
}
catch(IOException e)
{
    System.out.println("Error: " + e);
}
if (!foundStatus.equals("Yes"))
{
System.out.println ("Status " + foundStatus);
    custData[0] = "Bad";
}
return custData;
}

public static boolean fileExist()
{
    if(fileExist)
        return true;
    else
        return false;
}

public void putCustomer (String custNum, String lastName,
    String firstName, String middleName, String street, String city,

```

```

String state, String zip, String entryDate, String custType,
String serType, String lastMeterDate, String lastMeterRead,
String currMeterDate, String currMeterRead, String outstanding,
String currBalance, String lastPayDate)
{
String foundStatus = "No";
int recordLength = 164 * 2;
int custLength = 0;
String tempCustNum = "          " + custNum;
custLength = tempCustNum.length();
custNum = tempCustNum.substring(custLength - 6, custLength);
try
{
RandomAccessFile CustomerDAFile = new RandomAccessFile(
    "CustomerDAFile.dat", "rw");
for(int i = 0; i < custCount; i++)
{
    if (custNum.equals(custNumRef[i]))
    {
        int recordNumber = i;
        CustomerDAFile.seek(recordNumber * recordLength);
        String custData = parseField(CustomerDAFile, 163);
        recordNumber = i;
        CustomerDAFile.seek(recordNumber * recordLength);

        custNum = FixField(custNum, 6);
        lastName = FixField(lastName, 15);
        firstName = FixField(firstName, 15);
        middleName = FixField(middleName, 15);
        street = FixField(street, 25);
        city = FixField(city, 15);
        state = FixField(state, 2);
        zip = FixField(zip, 5);
        entryDate = FixField(entryDate, 8);
        custType = FixField(custType, 1);
        serType = FixField(serType, 1);
        lastMeterDate = FixField(lastMeterDate, 8);
        lastMeterRead = FixField(lastMeterRead, 6);
        currMeterDate = FixField(currMeterDate, 8);
        currMeterRead = FixField(currMeterRead, 6);
        outstanding = FixField(outstanding, 10);
        currBalance = FixField(currBalance, 10);
    }
}
}
}

```

```

        lastPayDate = FixField(lastPayDate, 8);

        String newCustData = custNum + lastName + firstName
            + middleName + street + city + state + zip +
            entryDate + custType + serType + lastMeterDate
            + lastMeterRead + currMeterDate + currMeterRead
            + outstanding + currBalance + lastPayDate;

        CustomerDAFile.writeChars(newCustData);
        foundStatus = "Yes";
        CustomerDAFile.close();
    }
}
if (foundStatus.equals("Yes"))
{
    JOptionPane.showMessageDialog(null, "Customer Number" + custNum.trim() + " is updated.",
        "New File Notification", JOptionPane.PLAIN_MESSAGE);
}
else
{
    JOptionPane.showMessageDialog(null, "Customer Number" + custNum.trim() + " is not
found.",
        "New File Notification", JOptionPane.ERROR_MESSAGE);
}
}

catch (IOException e)
{
    System.out.println ("Error: " + e);
}
}

public String parseField(RandomAccessFile CustomerDAFile, int fieldLength)
{
    String custField = "";
    try
    {
        for(int i = 0; i < fieldLength; i++)
        {
            char c = CustomerDAFile.readChar();
            custField += c;
        }
    }
}

```

```
    }  
    catch (IOException e)  
    {  
        System.out.println ("Error: " + e);  
    }  
    return custField;  
}
```

```
public static String FixField(String stringField, int stringLength)
```

```
{  
    String resultField = "";  
    int itemLength, beginPoint = 0;  
    resultField = stringField;  
    resultField = "          " + resultField;  
    itemLength = resultField.length();  
    beginPoint = itemLength - stringLength;  
    resultField = resultField.substring(beginPoint, itemLength);  
    return resultField;  
}  
}
```