

```

/*
    Problem 4C:      DME Customer class for creating
                    customer objects
                    creates a package
                    Documentation version

    Programmer:     Wayne Spence
    Date:           September 16, 2003
    Program Name:   DME4CCustomer
*/

package DME;
import javax.swing.*;

/**
<b> DME4CCustomer </b> class for the creation of customer electrical service
objects with associated attributes and methods.
*/
public class DME4CCustomer
{
    //Declare instance variables
    private String customerNameString;
    private String customerTypeString;
    private int customerNumber;
    private int serviceType;
    private int meterReadingDay;
    private int meterReadingMonth;
    private int meterReadingYear;
    private int kilowattHours;

    private String outputLine = "";
    public static final String DME_PANE_TITLE = "Denton Municipal Utilities";

    /**
    Initialize (create default values for) instance variables that includes
    customer number and name, customer type, service type, meter reading day/month/year
    and kilowatt hours
    */
    public DME4CCustomer()
    {
        customerNameString = "";

```

```
customerTypeString = "R";
customerNumber= 0;
serviceType= 1;
meterReadingDay = 10;
meterReadingMonth = 9;
meterReadingYear = 2003;
kilowattHours = 0;
}

/**
Controls prompting the user for the customer number (an integer variable)
*/
public void setCustomerNumber()
{
    customerNumber = promptUser("customer number", customerNumber, 1, 99999);
}

/**
Controls prompting the user for the customer name (a string variable)
*/
public void setCustomerName()
{
    customerNameString = promptUser ("customer name", customerNameString);
}

/**
Controls prompting the user for the customer type (a string variable
with valid values of R, C or G)
*/
public void setCustomerType()
{
    customerTypeString = promptUser ("customer type (R, C or G)", "R");
}

/**
Controls prompting the user for the service type (an integer variable
with valid values of 1 or 2)
*/
public void setServiceType()
{
    serviceType = promptUser ("service type (1 or 2)", serviceType, 1, 2);
}
```

```
/**
Controls prompting the user for the meter reading day (an integer variable)
*/
public void setMeterReadingDay()
{
    meterReadingDay = promptUser ("meter reading day", meterReadingDay, 1, 31);
}

/**
Controls prompting the user for the meter reading month (an integer variable)
*/
public void setMeterReadingMonth()
{
    meterReadingMonth = promptUser ("meter reading month", meterReadingMonth, 1, 12);
}

/**
Controls prompting the user for the meter reading year (an integer variable)
*/
public void setMeterReadingYear()
{
    meterReadingYear = promptUser ("meter reading year", meterReadingYear, 2000, 2003);
}

/**
Controls prompting the user for the number of kilowatt hours used (an integer variable)
*/
public void setKilowattHours()
{
    kilowattHours = promptUser ("kilowatt hours usage (KWH)", kilowattHours, 1, 99999);
}

/**
Provides the current value of the customer number
*/
public int getCustomerNumber()
{
    return customerNumber;
}

/**
```

```
Provides the current value of the customer name
*/
public String getCustomerName()
{
    return customerNameString;
}

/**
Provides the current value of the customer type
*/
public String getCustomerType()
{
    return customerTypeString;
}

/**
Provides the current value of the service type
*/
public int getServiceType()
{
    return serviceType;
}

/**
Provides the current value of the meter reading day
*/
public int getMeterReadingDay()
{
    return meterReadingDay;
}

/**
Provides the current value of the meter reading month
*/
public int getMeterReadingMonth()
{
    return meterReadingMonth;
}

/**
Provides the current value of the meter reading year
*/
```

```

public int getMeterReadingYear()
{
    return meterReadingYear;
}

/**
Provides the current value of the kilowatt hours used
*/
public int getKilowattHours()
{
    return kilowattHours;
}

/**
Generic procedure for prompting the user for input and accepting an integer value.
Controls include a user prompt string, an initial (default) value, and the minimum
and maximum acceptable values.
*/
private static int promptUser (String userPrompt, int initialValue, int minValue, int maxValue)
{
    boolean goodData = false;
    String outputLine = "";
    String initialValueString = Integer.toString(initialValue);
    int intNumber = 0;

    while (!goodData)
    {
        String intNumberString =
            JOptionPane.showInputDialog(null, "What is the " + userPrompt + "?", initialValueString);
        try
        {
            intNumber = Integer.parseInt(intNumberString);
            if (intNumber < minValue || intNumber > maxValue)
            {
                outputLine = "The " + userPrompt + " should be between " +
                    minValue + " and " + maxValue + ".";
                displayMessage(outputLine);
            }
            else
                goodData = true;
        }
        catch (NumberFormatException e)
        {

```

```

        outputLine = "You have not entered digits for the " + userPrompt + ".";
        displayMessage(outputLine);
    }
}
return intNumber;
}

/**
Generic procedure for prompting the user for input and accepting a string value.
Controls include a user prompt string and an initial (default) value.
*/
private static String promptUser (String userPrompt, String initialValue)
{
    boolean goodData = false;
    String outputLine = "";
    String stringData = "";
    while (!goodData)
    {
        stringData =
            JOptionPane.showInputDialog(null, "What is the " + userPrompt + "?",
                initialValue);
        goodData = true;
    }
    return stringData;
}

/**
Generic procedure for producing user prompting. Controls include a prompting
message text.
*/
private static void displayMessage (String messageText)
{
    String errorTitle = "An error has occurred";
    messageText = messageText + "\nPlease, try again!";

    JOptionPane.showMessageDialog(null, messageText, errorTitle,
        JOptionPane.ERROR_MESSAGE);
}
}

```