package Homework;  
  
  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.Objects;  
import java.util.Scanner;  
  
abstract class Customer {  
 String type;  
 String maKH;  
 String tenKH;  
 Date ngHD;  
 int sLuongDien;  
 float donGia;  
 float thanhTien;  
  
 public Customer() {  
 this.maKH = "";  
 this.tenKH = "";  
 this.ngHD = new Date();  
 this.sLuongDien = 0;  
 this.donGia = 0.0F;  
 this.thanhTien = 0.0F;  
 type = "";  
 }  
  
 public String get\_maKH(){  
 return maKH;  
 }  
  
 public String get\_tenKH(){  
 return tenKH;  
 }  
  
 public String get\_type(){  
 return type;  
 }  
  
 public Date get\_ngHD(){  
 return ngHD;  
 }  
 public Customer(String maKH, String tenKH, String ngHD, int sLuongDien, float donGia, float thanhTien) {  
 this.maKH = maKH;  
 this.tenKH = tenKH;  
 this.ngHD = new Date();  
 this.sLuongDien = sLuongDien;  
 this.donGia = donGia;  
 this.thanhTien = thanhTien;  
 type = "";  
 }  
  
 public abstract void tinhThanhTien();  
  
 public void inThongTin() {  
 System.*out*.println("Mã khách hàng: " + maKH);  
 System.*out*.println("Tên khách hàng: " + tenKH);  
 System.*out*.println("Ngày hóa đơn: " + ngHD);  
 System.*out*.println("Số lượng điện: " + sLuongDien);  
 System.*out*.println("Đơn giá điện: " + donGia);  
 System.*out*.println("Thành tiền: " + thanhTien);  
 }  
  
 protected void nhapThongTin() throws ParseException {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Nhập mã khách hàng: ");  
 maKH = scanner.nextLine();  
 if (Objects.*equals*(maKH, "")) {  
 System.*out*.println("Mã khách hàng không được để trống");  
 System.*exit*(0);  
 }  
 System.*out*.println("Nhập tên khách hàng: ");  
 tenKH = scanner.nextLine();  
 System.*out*.println("Nhập ngày hóa đơn dưới dạng yyyy-MM-dd: ");  
 String ngayHoaDon = scanner.nextLine();  
 try{  
 SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");  
 ngHD = format.parse(ngayHoaDon);  
 }catch (Exception e){  
 System.*out*.println("Ngày hóa đơn không đúng định dạng");  
 System.*exit*(0);  
 }  
 System.*out*.println("Nhập số lượng điện: ");  
 sLuongDien = scanner.nextInt();  
 System.*out*.println("Nhập đơn giá: ");  
 donGia = scanner.nextFloat();  
 }  
  
}  
  
class SinhHoat extends Customer {  
 int dinhMuc;  
  
 public SinhHoat() {  
 super();  
 this.dinhMuc = 0;  
 type = "Sinh hoạt";  
 }  
  
  
 @Override  
 public void tinhThanhTien() {  
 if (this.sLuongDien <= this.dinhMuc) {  
 this.thanhTien = sLuongDien \* donGia;  
 } else {  
 thanhTien = donGia \* dinhMuc + (sLuongDien - dinhMuc) \* donGia \* 2;  
 }  
 }  
  
 @java.lang.Override  
 public void inThongTin() {  
 super.inThongTin();  
 System.*out*.println("Định mức: " + dinhMuc);  
 }  
  
 public void nhapThongTin() {  
 try {  
 super.nhapThongTin();  
 } catch (ParseException e) {  
 throw new RuntimeException(e);  
 }  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Nhập định mức: ");  
 dinhMuc = scanner.nextInt();  
 }  
}  
  
class KinhDoanh extends Customer {  
 public KinhDoanh() {  
 super();  
 type = "Kinh doanh";  
 }  
 public KinhDoanh(String maKH, String tenKH, String ngHD, int sLuongDien, float donGia, float thanhTien) {  
 super(maKH, tenKH, ngHD, sLuongDien, donGia, thanhTien);  
 }  
  
 public void tinhThanhTien() {  
 if (sLuongDien <= 400) {  
 thanhTien = sLuongDien \* donGia;  
 } else {  
 thanhTien = (float) ((sLuongDien - 400) \* donGia \* 1.05 + 400 \* donGia);  
 }  
 }  
  
 public void nhapThongTin() {  
 try {  
 super.nhapThongTin();  
 } catch (ParseException e) {  
 throw new RuntimeException(e);  
 }  
 }  
}  
  
class SanXuat extends Customer {  
 String loaiDien;  
 public SanXuat() {  
 super();  
 this.loaiDien = "";  
 type = "Sản xuất";  
 }  
 public SanXuat(String maKH, String tenKH, String ngHD, int sLuongDien, float donGia, float thanhTien, String loaiDien) {  
 super(maKH, tenKH, ngHD, sLuongDien, donGia, thanhTien);  
 this.loaiDien = loaiDien;  
 }  
  
  
 @Override  
 public void tinhThanhTien() {  
 if (loaiDien.equals("2 pha")) {  
 if (sLuongDien <= 200) {  
 thanhTien = sLuongDien \* donGia;  
 } else {  
 thanhTien = (float) (200 \* donGia + (sLuongDien - 200) \* donGia \* 0.98);  
 }  
 }  
 if (loaiDien.equals("3 pha")) {  
 if (sLuongDien <= 150) {  
 thanhTien = sLuongDien \* donGia;  
 } else {  
 thanhTien = (float) (150 \* donGia + (sLuongDien - 150) \* donGia \* 0.97);  
 }  
 }  
  
 }  
  
 @java.lang.Override  
 public void inThongTin() {  
 super.inThongTin();  
 System.*out*.println("Loại điên" + loaiDien);  
  
 }  
  
 public void nhapThongTin() {  
 try {  
 super.nhapThongTin();  
 } catch (ParseException e) {  
 throw new RuntimeException(e);  
 }  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Nhập loại điện: ");  
 loaiDien = scanner.nextLine();  
 }  
}  
class QuanLyKhachHang{  
 private final ArrayList<Customer> customers;  
  
 private int index;  
  
 private int soLuongKhachHang;  
  
 public QuanLyKhachHang(int soLuongKhachHang){  
 index = 0;  
 customers = new ArrayList<Customer>(soLuongKhachHang);  
 }  
 public void input( int type) {  
 try{  
 switch (type) {  
 case 1:  
 Customer sinhHoat = new SinhHoat();  
 customers.add(index, sinhHoat);  
 customers.get(index).nhapThongTin();  
 break;  
 case 2:  
 customers.add(index, new KinhDoanh());  
 customers.get(index).nhapThongTin();  
 break;  
 case 3:  
 customers.add(index, new SanXuat());  
 customers.get(index).nhapThongTin();  
 break;  
 }  
 }catch (Exception e){  
 System.*out*.println(e);  
 System.*out*.println("Nhập thông tin không đúng");  
 System.*exit*(0);  
 }  
 checkValidWhenTheSameCustomer(index);  
 index ++;  
 }  
  
 public void displayEnergy(String month, String year) {  
 SimpleDateFormat sdf = new SimpleDateFormat("MM"); *//* SimpleDateFormat yearFormat = new SimpleDateFormat("yyyy");  
 for (int i = 0; i < index; i++){  
 String monthString = sdf.format(customers.get(i).get\_ngHD());  
 String yearString = yearFormat.format(customers.get(i).get\_ngHD());  
 if (monthString.equals(month) && yearString.equals(year)) {  
 customers.get(i).tinhThanhTien();  
 System.*out*.println("------------------------------------------------");  
 customers.get(i).inThongTin();  
 System.*out*.println("------------------------------------------------");  
 }  
 }  
 }  
  
 public void checkValidWhenTheSameCustomer (int customer\_position){  
 for (int i = 0; i <= customer\_position; i++){  
 if (i == customer\_position){  
 continue;  
 }  
 if (customers.get(i).get\_maKH().equals(customers.get(customer\_position).get\_maKH()) && customers.get(i).get\_tenKH().equals(customers.get(customer\_position).get\_tenKH())){  
 if (!Objects.*equals*(customers.get(i).get\_type(), customers.get(customer\_position).get\_type())){  
 System.*out*.println("Khách hàng trùng tên và mã khách hàng nhưng khác loại khách hàng");  
 System.*exit*(0);  
 }  
 }  
 }  
  
 }  
}  
public class Lab2\_1 {  
 public static void main(String[] args){  
 Scanner scanner = new Scanner(System.*in*);  
 int soLuongKhachHang;  
 System.*out*.println("Nhập số lượng khách hàng: ");  
 soLuongKhachHang = scanner.nextInt();  
 QuanLyKhachHang quanLyKhachHang = new QuanLyKhachHang(soLuongKhachHang);  
 while(true){  
 System.*out*.println("Nhập loại khách hàng: 1. Sinh hoạt, 2. Kinh doanh, 3. Sản xuất");  
 System.*out*.println("--------------------------------");  
 System.*out*.println("\nNhập 4 để xuất hóa đơn tiền điện");  
 int type = scanner.nextInt();  
 if (type == 4){  
 System.*out*.println("Nhập tháng: ");  
 String month = scanner.next();  
 System.*out*.println("Nhập năm: ");  
 String year = scanner.next();  
 quanLyKhachHang.displayEnergy(month, year);  
 }else{  
 quanLyKhachHang.input(type);  
 }  
 }  
 }  
}