Main – главный класс, из которого начинается выполнение программы, сначала мы создаем двух клиентов, две компании и два диалога между ними соответственно, далее в каждом из диалогов отправляем сообщения от каждого из лиц, в конце выводим все диалоги и убеждаемся, что программа работает корректно.

package v\_4;  
  
public class Main {  
  
 public static void main(String[] args) {  
 Client client1 = new Client("Jack");  
 Client client2 = new Client("Mark");  
  
 Company company1 = new Company("Samsung");  
 Company company2 = new Company("Asos");  
  
 Conversation conversation1 = new Conversation(client1, company1);  
 Conversation conversation2 = new Conversation(client2, company2);  
  
 client1.sendMessage(0, "Добрый день");  
 company1.sendMessage(0, "Здравствуйте! Чем могу помочь?");  
  
 client2.sendMessage(0, "abcd");  
 company2.sendMessage(0, "efgh");  
 client2.sendMessage(0, "ijkl");  
 company2.sendMessage(0, "mnop");  
  
 conversation1.printAllMessanges();  
 System.*out*.println();  
 conversation2.printAllMessanges();  
 }  
}

Massage – класс, описывающий сообщение. Состоит из двух полей: имя отправителя и текст сообщения.

package v\_4;  
  
public class Message {  
 public String name;  
 public String message;  
 public Message(String message, String name){  
 this.message = message;  
 this.name = name;  
 }  
}

Client – класс, описывающий клиента. Состоит из двух полей: имя клиента и его диалоги – и имеет методы добавления диалога и отправки сообщения в выбранный диалог.

package v\_4;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class Client {  
 public String name;  
 public List<Conversation> conversations;  
 public Client(String name){  
 this.name = name;  
 conversations = new ArrayList<>();  
 }  
 public void addConversation(Conversation conversation){  
 conversations.add(conversation);  
 }  
 public void sendMessage(int numConv, String message){  
 conversations.get(numConv).sendMessage(new Message(message, name));  
 }  
}

Company – класс, описывающий компанию. Состоит из 2 полей: название компании и список ее диалогов. Имеет методы для добавления нового диалога и отправки сообщения в выбранный диалог.

package v\_4;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class Company {  
 public String name;  
 public List<Conversation> conversations;  
 public Company(String name){  
 this.name = name;  
 this.conversations = new ArrayList<>();  
 }  
 public void addConversation(Conversation conversation){  
 conversations.add(conversation);  
 }  
 public void sendMessage(int numConv, String message){  
 conversations.get(numConv).sendMessage(new Message(message, name));  
 }  
}

Conversation – класс, описывающий диалог. В нем находятся такие поля, как клиент и пользователь, между которыми и происходит диалог, и список всех сообщений между ними. Методы данного класса отвечают за отправку сообщения и вывод всего диалога.

package v\_4;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class Conversation {  
 public Company company;  
 public Client client;  
 private List<Message> messages;  
 public Conversation(Client client, Company company){  
 this.client = client;  
 this.company = company;  
 client.addConversation(this);  
 company.addConversation(this);  
 messages = new ArrayList<>();  
 }  
 public void sendMessage(Message message){  
 messages.add(message);  
 }  
 public void printAllMessanges(){  
 for (Message message : messages){  
 System.*out*.println(message.name + ": " + message.message);  
 }  
 }  
}