CoFollowersAmount // Количество водителей, которые могут подвести.

try {

if (CoFollowersAmount > 1) throws new TooManyDriversException;

if (CoFollowersAmount == 1) throws new DriverJustOneException;

if (CoFollowersAmount == 0) throws new DriverNotFoundException;

}

catch ( TooManyDriversException e) {

int i =0;

do {

If (CountDistance(Driver1, Client) > (CountDistance(Driver2, Client)) {FinalDriver = Driver2;}

else {FinalDriver = Driver1;

i++;

} while ( i!= CoFollowersAmount;

;

Message = “Вас может подвести водитель %s .”, Driver.Name;}

catch ( DriverJustOneException e) { Message = “Вас может подвести водитель %s .”, Driver.Name;}

catch ( DriverNotFoundException e) { Message = “Водителей едущих в вашу сторону нет.”;}

finally { SendMessage(Client.Phone\_Number, Message); // функция, которая отправляет клиенту сообщение. Вместо неё в прототипе будет использоваться вывод в консоль или на экран.

Message2 = “Вас попросил подвести %s . Координаты (%d, %d).”, Client.Name, Client.coordinates.x, Client.coordinates.y;

SendMessage(Driver.Phone\_Number, Message2);

}

**try {**

**if (DriverDelay == 1) throws new DriverDelayException;**

**}**

**catch ( DriverNotFoundException e) { Message = “Извините, но выбраный водитель отказался. Мы подберём и отправим вам другого.”;**

**SendMessage(Client.Phone\_Number , Message);**

**}**

**class DriverNotFoundException extends Exception{**

**public DriverNotFoundException(Throwable e) {   
        initCause(e);   
    }**

**}**

**class TooManyDriversException extends Exception{**

**public TooManyDriversException(Throwable e) {   
        initCause(e);   
    }**

**}**

**class DriverJustOneException extends Exception{**

**public DriverJustOneException n(Throwable e) {   
        initCause(e);   
    }**

**}**

**class DriverDelayException extends Exception{**

**public DriverDelayException (Throwable e) {   
        initCause(e);   
    }**

**}**