Министерство образования и науки РФ

Институт информационных технологий и управления в технических системах

Севастопольский государственный университет

Лабораторная работа №5

по дисциплине: «ТСПП»

«**Исследование способов построения диаграмм последовательностей**»

Выполнил:

ст.гр. ИСб/22о

Воронин И.Ю.

Проверил:

Дрозин А.Ю.

г. Севастополь

1. Цель работы

Исследовать способы представления временных особенностей передачи и приема сообщений между объектами. Изучить способы представления объектов, сообщений и времени на диаграммах последовательностей.

1. Вариант задания

Для предметной области, проанализированной в лабораторной работе №1, составить диаграммы последовательностей.

1. Ход работы

Изобразим деятельность класса ClientPay, содержащего следующие классы в своей реализации:

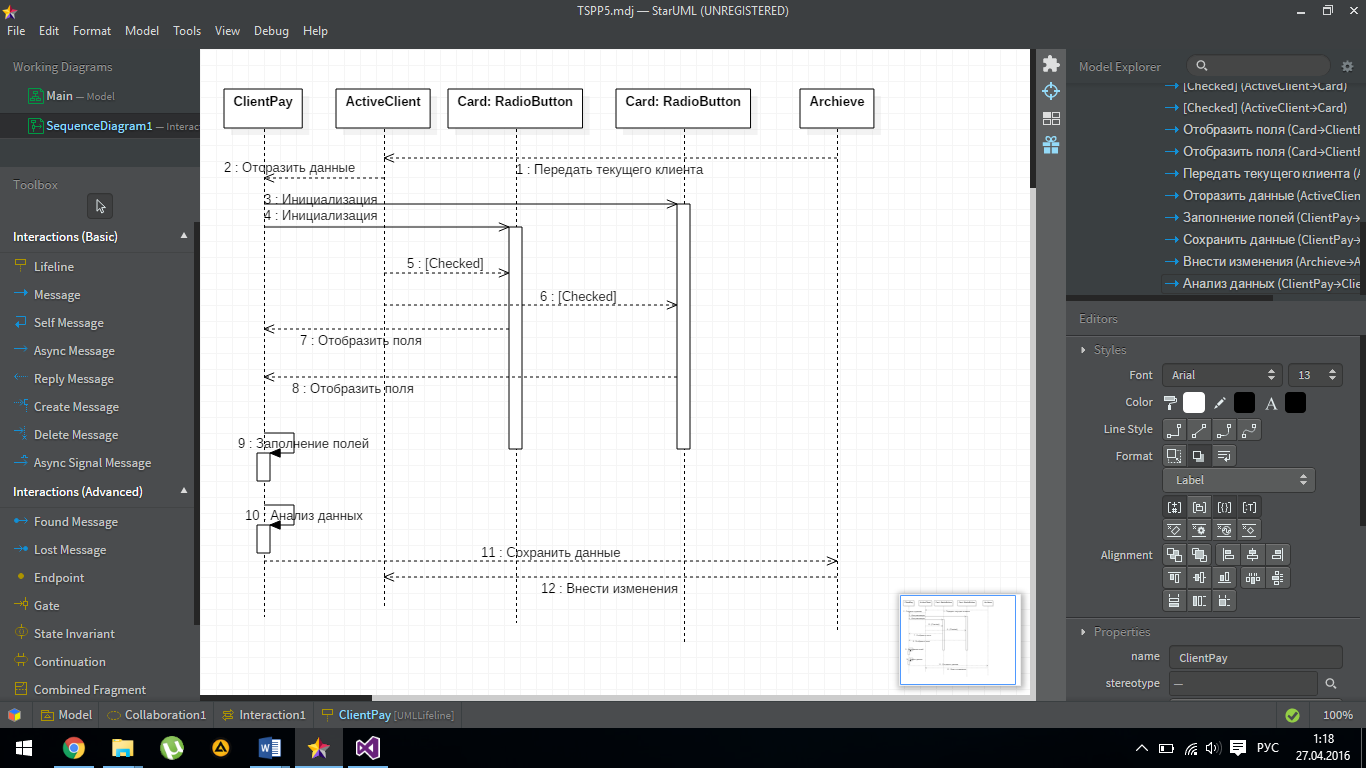


Рисунок 3.1 – Диаграмма последовательности.

На данной диаграмме рассматривается работа класса-формы, в которой активный пользователь выбирает как он желает совершить оплату: кредиткой или наличными.

4.Код программы

namespace InsurenceTerminal

{

public partial class ClientPay : Form

{

private int SMS\_Code;

public ClientPay()

{

InitializeComponent();

BalanceLabel.Text = ActiveClient.client.balance.Money + " руб.";

SMStoolTip.ShowAlways = false;

pictureBox1.Hide();

CashLabel.Hide();

CardLabel1.Hide();

CardLabel2.Hide();

CardLabel3.Hide();

CardMoneyTbx.Hide();

Card1Tbx.Hide();

Card2Tbx.Hide();

Card3Tbx.Hide();

Card4Tbx.Hide();

DefisLbl1.Hide();

DefisLbl2.Hide();

DefisLbl3.Hide();

SMSTbx.Hide();

SMSBtn.Hide();

InitializeComboBox();

}

private void InitializeComboBox()

{

String contractType = "";

foreach (Contract contract in Archieve.ContractList)

{

if (contract.client.Passport.Equals(ActiveClient.client.Passport) &&

contract.client.Login.Equals(ActiveClient.client.Login))

{

switch (contract.contractType)

{

case ContractType.car: contractType = "Машина"; break;

case ContractType.health: contractType = "Здоровье"; break;

case ContractType.house: contractType = "Дом"; break;

}

ContractComboBox.Items.Add(contractType + ": " + contract.Name);

}

}

}

private void PrevBtn\_Click(object sender, EventArgs e)

{

this.Dispose();

ClientMenu clientMenu = new ClientMenu(ActiveClient.client);

clientMenu.MdiParent = Start.ActiveForm;

clientMenu.Show();

}

private void CashRadioButton\_Click(object sender, EventArgs e)

{

if (CashRadioButton.Checked)

{

pictureBox1.Show();

CashLabel.Show();

CardLabel1.Hide();

CardLabel2.Hide();

CardLabel3.Hide();

CardMoneyTbx.Hide();

Card1Tbx.Hide();

Card2Tbx.Hide();

Card3Tbx.Hide();

Card4Tbx.Hide();

DefisLbl1.Hide();

DefisLbl2.Hide();

DefisLbl3.Hide();

SMSTbx.Hide();

SMSBtn.Hide();

}

}

private void CardRadioButton\_Click(object sender, EventArgs e)

{

if (CardRadioButton.Checked)

{

CardLabel1.Show();

CardLabel2.Show();

CardLabel3.Show();

CardMoneyTbx.Show();

Card1Tbx.Show();

Card2Tbx.Show();

Card3Tbx.Show();

Card4Tbx.Show();

DefisLbl1.Show();

DefisLbl2.Show();

DefisLbl3.Show();

SMSTbx.Show();

SMSBtn.Show();

pictureBox1.Hide();

CashLabel.Hide();

}

}

private void OKBtn\_Click(object sender, EventArgs e)

{

if(Card1Tbx.Text.Length == 4 &&

Card2Tbx.Text.Length == 4 &&

Card3Tbx.Text.Length == 4 &&

Card4Tbx.Text.Length == 4 &&

CardMoneyTbx.Text.Length > 0

){

MessageBox.Show(

"СМС сообщение с кодом отправлено.",

"Внимание",

MessageBoxButtons.OK,

MessageBoxIcon.None

);

Random random = new Random();

SMS\_Code = random.Next(9999);

SMStoolTip.SetToolTip(SMSBtn, "" + SMS\_Code);

}

}

private void NextBtn\_Click(object sender, EventArgs e)

{

String contractType = "";

foreach (Contract contract in Archieve.ContractList)

{

switch (contract.contractType)

{

case ContractType.car: contractType = "Машина"; break;

case ContractType.health: contractType = "Здоровье"; break;

case ContractType.house: contractType = "Дом"; break;

}

if (ContractComboBox.Text.Equals(contractType + ": " + contract.Name))

{

if (CashRadioButton.Checked)

{

Random random = new Random();

int Money = 50 \* random.Next(100);

ActiveClient.client.balance.Pay(contract,Money);

MessageBox.Show(

"Было внесено " + Money + " рублей",

"Внимание",

MessageBoxButtons.OK,

MessageBoxIcon.Asterisk

);

}

if (CardRadioButton.Checked)

{

if (Card1Tbx.Text.Length == 4 &&

Card2Tbx.Text.Length == 4 &&

Card3Tbx.Text.Length == 4 &&

Card4Tbx.Text.Length == 4 &&

CardMoneyTbx.Text.Length > 0 &&

SMSTbx.Text.Length > 0 &&

SMS\_Code.ToString().Equals(SMSTbx.Text)

)

{

ActiveClient.client.balance.Pay(

contract,

Convert.ToInt32(CardMoneyTbx.Text)

);

MessageBox.Show(

"Операция успешно совершена.",

"Внимание",

MessageBoxButtons.OK,

MessageBoxIcon.Asterisk

);

}

else

{

return;

}

//

}

Serializator.SerializeClients();

this.Dispose();

ClientMenu clientMenu = new ClientMenu(ActiveClient.client);

clientMenu.MdiParent = Start.ActiveForm;

clientMenu.Show();

return;

}

}

}

private void MoneyTbx\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(Char.IsDigit(e.KeyChar)) && e.KeyChar != Convert.ToChar(8))

{

e.Handled = true;

}

}

private void Card1Tbx\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(Char.IsDigit(e.KeyChar)) && e.KeyChar != Convert.ToChar(8))

{

e.Handled = true;

}

}

private void Card2Tbx\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(Char.IsDigit(e.KeyChar)) && e.KeyChar != Convert.ToChar(8))

{

e.Handled = true;

}

}

private void Card3Tbx\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(Char.IsDigit(e.KeyChar)) && e.KeyChar != Convert.ToChar(8))

{

e.Handled = true;

}

}

private void Card4Tbx\_KeyPress(object sender, KeyPressEventArgs e)

{

if (!(Char.IsDigit(e.KeyChar)) && e.KeyChar != Convert.ToChar(8))

{

e.Handled = true;

}

}

}

}