Приложения

ПРИЛОЖЕНИЕ А

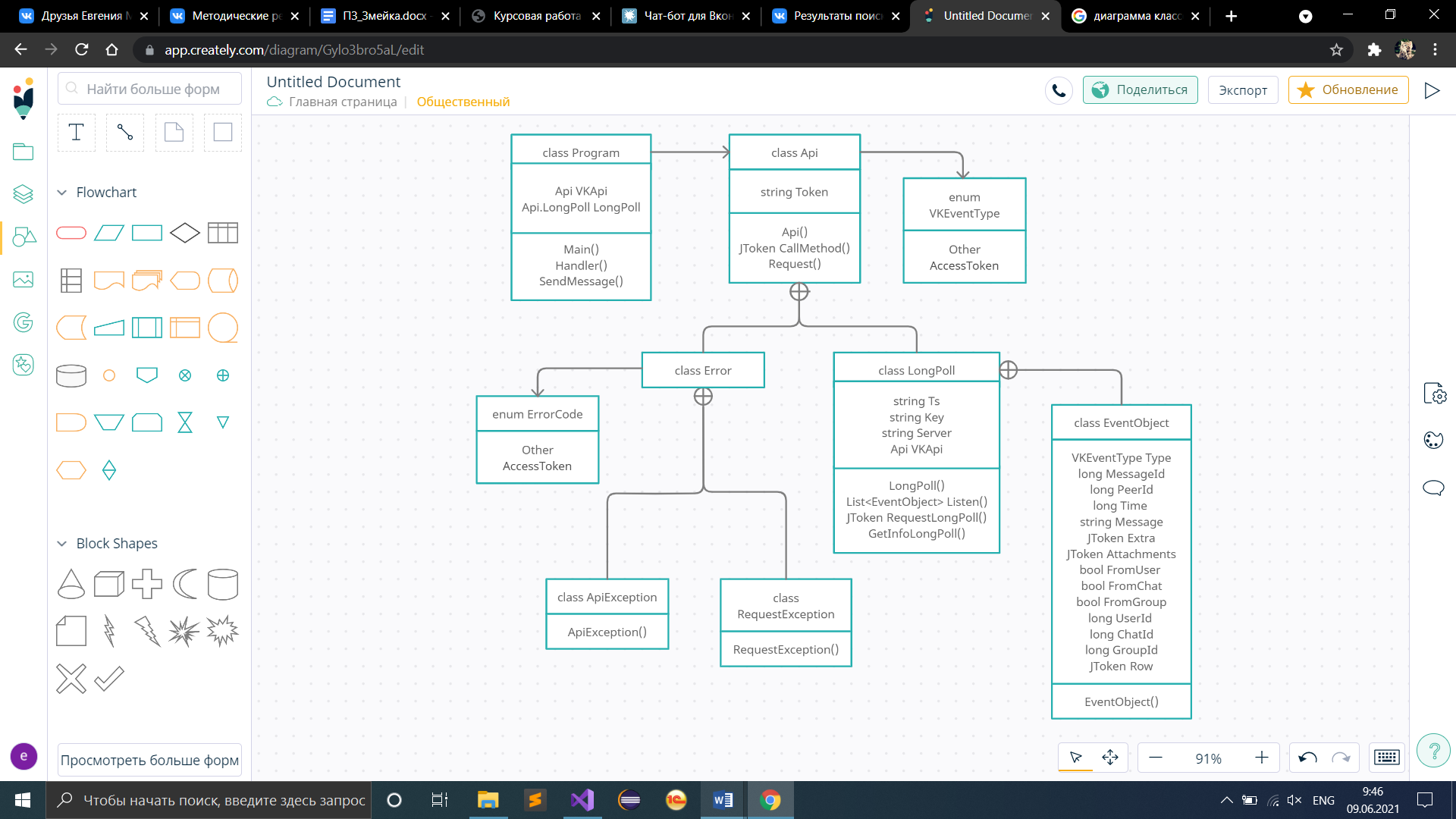


Рисунок A.1 – диаграмма классов

ПРИЛОЖЕНИЕ B

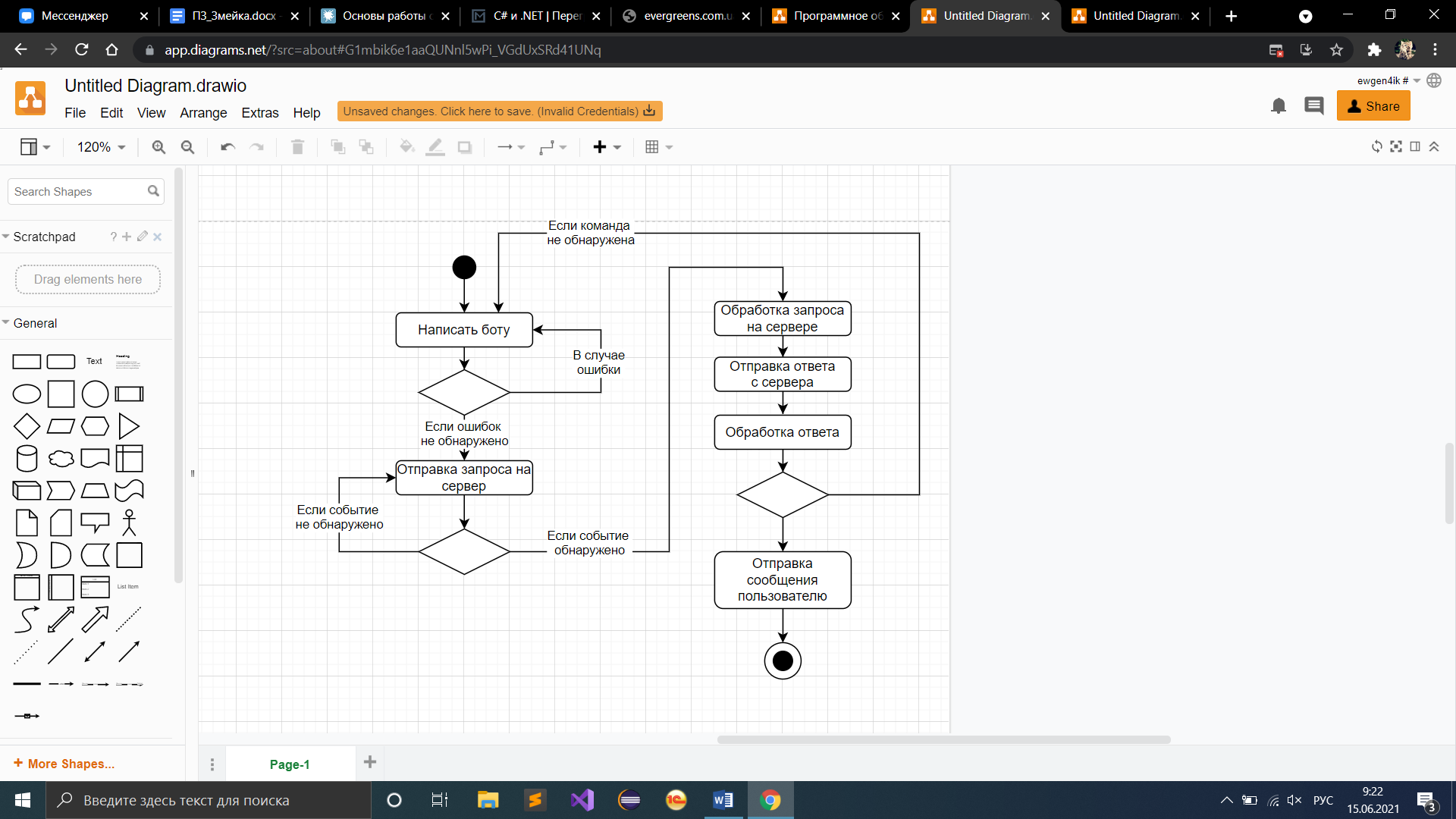


Рисунок B.1 – диаграмма деятельности

ПРИЛОЖЕНИЕ C

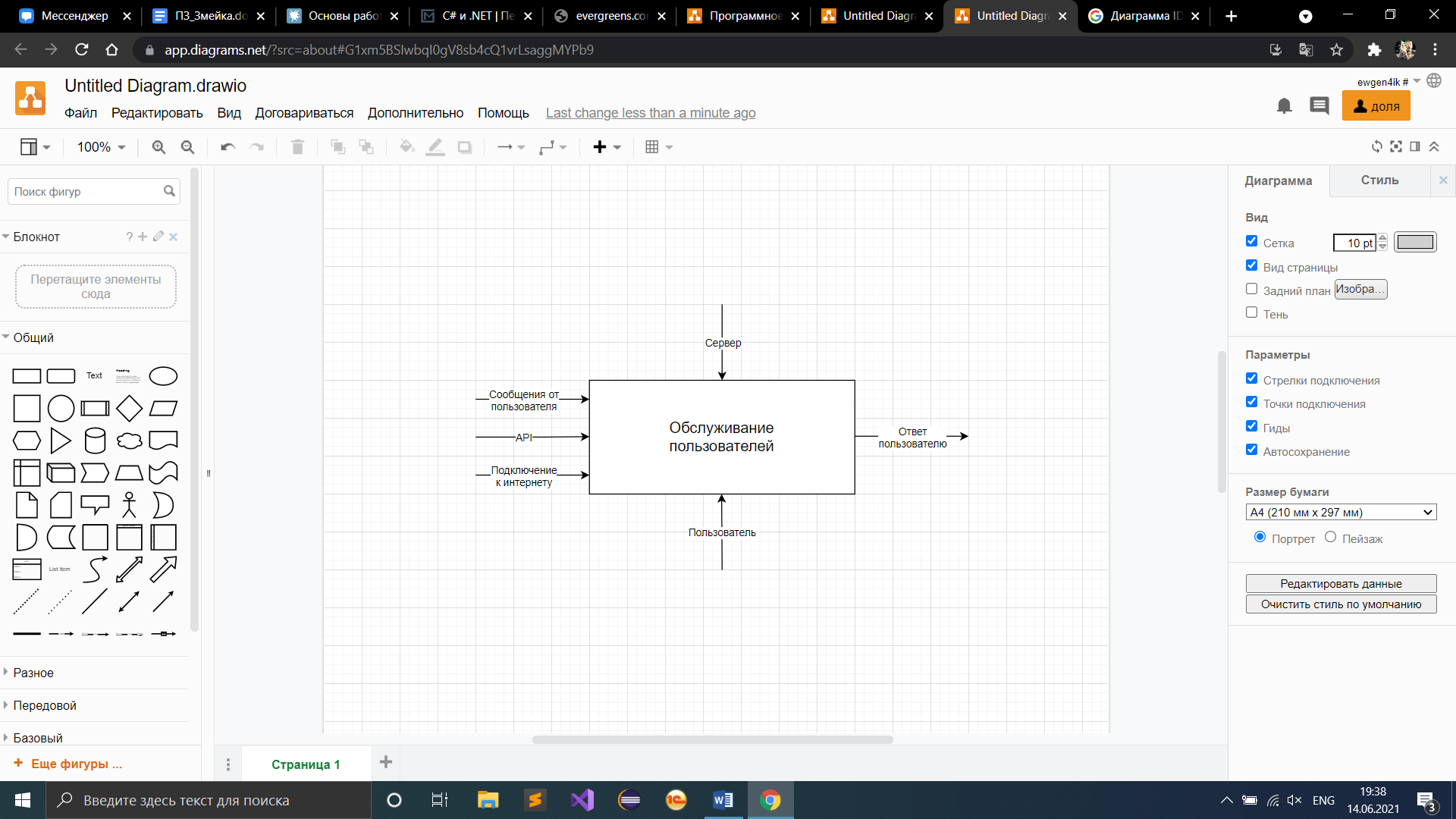
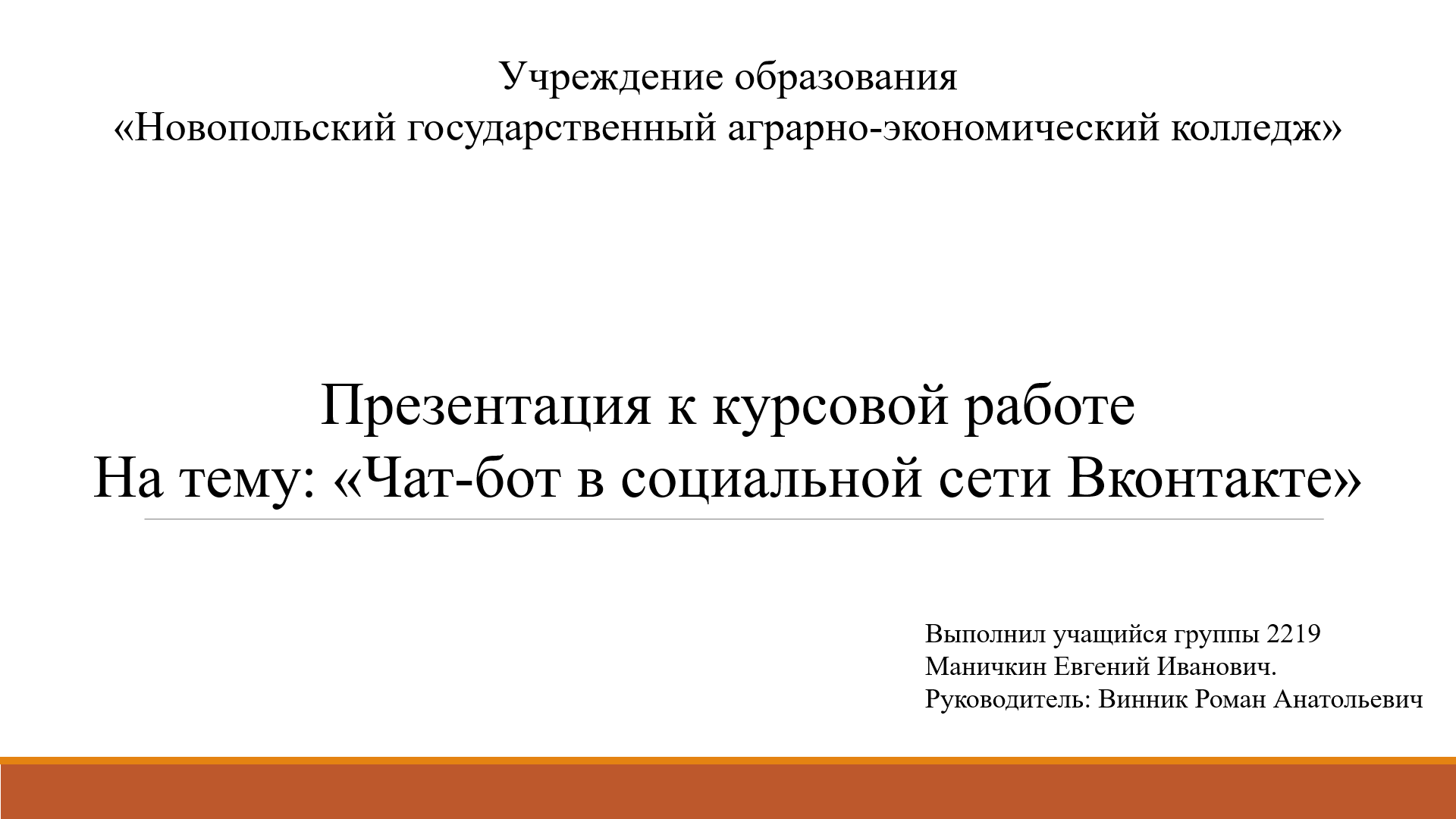


Рисунок C.1 – диаграмма нотации IDEF0

ПРИЛОЖЕНИЕ D



**Рисунок D.1 – созданная презентация**

ПРИЛОЖЕНИЕ E

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace LessonBot

{

public class Program

{

static Api VKApi = new Api("bf103c02ffda833c64adffd36eab0875ba080b5c46e4f6b4946e346e5d570a3897858579ec34306c2f039");

static Api.LongPoll LongPoll = new Api.LongPoll(VKApi);

private static void Main(string[] args)

{

Console.ForegroundColor = ConsoleColor.Green;

Console.WriteLine("Бот запущен!");

Console.ResetColor();

while (true)

{

List<Api.LongPoll.EventObject> events = LongPoll.Listen();

foreach (var evt in events)

{

Handler(evt);

}

}

}

private static void Handler(Api.LongPoll.EventObject evt)

{

try

{

if (evt.Type == Api.VKEventType.MessageNew)

{

string[] message = evt.Message.ToLower().Split(' ');

if (message[0] == "начать")

{

SendMessage(evt.PeerId, "Добрый день, меня зовут бот 🤖Вин, я бот автоответчик и помогаю абитуриентам и учащимся Новопольского государственного аграрно-экономического колледжа!\nНапишите мне \"Помощь\" для получения инструкций!");

}

if (message[0] == "помощь")

{

SendMessage(evt.PeerId, "📋Меню бота:\n👤Абитуриентам\n🎓Студентам\n📞Контакты");

Console.Write(evt.PeerId + " ");

}

if(message[0] == "абитуриентам")

{

SendMessage(evt.PeerId, "Отлично, может быть вы хотите узнать получше о нашем колледже? Вот меню которым вы можете воспользоваться!");

SendMessage(evt.PeerId, "📋Меню бота:\n👤Специальности;\n🏢Общежития;\n🏬Колледж;\n🏞Просторы;\n🏀Спорт;\n📝Балы;\n✅Дополнительно.");

}

if (message[0] == "специальности")

{

SendMessage(evt.PeerId, "👤Специальности:\n💰Бухгалтерский учет анализ и контроль;\nСрок обучения: 2 года 10 месяцев (на базе 9 классов) или 1 год 10 месяцев (на базе 11 классов) (дневная форма обучения)\n2 года 8 месяцев (на базе 11 классов) (заочная форма обучения)\n🖥Программное обеспечение информационных технологий;\nСрок обучения: 3 года 10 месяцев (на базе 9 классов) (дневная форма обучения)\n✉Документоведение и документационное обеспечение управления;\nСрок обучения: 2 года 10 месяцев (на базе 9 классов) (дневная форма обучения)\n🏤Организация гостиничных услуг;\nСрок обучения: 2 года 10 месяцев (на базе 9 классов) (дневная форма обучения).");

}

if (message[0] == "общежития")

{

SendMessage(evt.PeerId, "Наш колледж распологает 3-мя общежитиями.\n2-е общежитие - во втором общежитии 3 этажа, 1-2 этажи для работников колледжа, 3-й для учащихся последних курсов специальности ПОИТ.\n3-е ощежитие - в третьем общежитии имеется 5 этажей, в нем живут учащиеся поступившие после 9-ти классов. \n4-е общежитие - в 4-м общежитии как и в 3-ем 5 этажей, но проживают в нем учащиеся поступившие после 11-ти классов, а так же поступившие после 9-ти классов специальности(ДОУ ОГУ).");

}

if (message[0] == "колледж")

{

SendMessage(evt.PeerId, "Коледж имеет 2 корпуса, старый и новый.\nНовый корпус имеет 4 этажа, и более 40 аудиторий, 3-й этаж считается компьютерным, здесь в каждая аудитория распологает компьютерами. Старый же корпус обладает своей историей берущее свое начало еще с 19-го века, о этом вы можете прочитать на просторах интернета. В старом корпусе более 10-ти аудиторий.");

}

if (message[0] == "просторы")

{

SendMessage(evt.PeerId, "Немного фотографий Новопольского ГАЭКа \nhttps://sun9-48.userapi.com/impg/48cpxcmPKRTdoK8Vf1N6imTr9djkFsCK\_x6wfA/yUMAp0GFyGE.jpg?size=720x546&quality=96&sign=424e9f4832cb3590fdf19064dd73f5ba&type=album\nhttps://sun9-63.userapi.com/impg/X06KFOlep1Pca2DhRuHnlSQqjMvvs5lLNT7oeQ/vL0VAzEknSQ.jpg?size=960x1280&quality=96&sign=ac554a0583b012b837f4af7adf760837&type=album");

}

if (message[0] == "спорт")

{

SendMessage(evt.PeerId, "Важную роль нашего коледжа играет спорт. В расположении колледжа имеется 1 спортзал, 1 стадион, и одна спортивная площадка. Здесь занимаются такими видами спорта как футбол, волейбол, баскетбол, тенис, лыжный спорт, а так же множество различных спортивных мероприятия и соревнований которые проходят ежемесячно.");

}

if (message[0] == "балы")

{

SendMessage(evt.PeerId, "https://sun9-53.userapi.com/impg/C\_PEe4i18k9918usDb9TfheRBLQy1vDPDsTudQ/Nf3VbthWo0A.jpg?size=895x867&quality=96&sign=b9c1845bfeae79bc334b346dd441ca11&type=album");

}

if (message[0] == "дополнительно")

{

SendMessage(evt.PeerId, "Некоторы фото колледжа:\nАктовый зал. https://sun9-24.userapi.com/impg/JUj3Gen7TL2lSem7sqlWq3DH831DFKz34VWzYg/TN1w\_Q61ypM.jpg?size=1600x900&quality=96&sign=285469520ac5c2605fed3833f7d26ed5&type=album");

}

if(message[0] == "студентам")

{

}

if (message[0] == "контакты")

{

SendMessage(evt.PeerId, "🧠Директор: Лукашевич Сергей Владимирович т. 8-017-505-45-34;\n📕Зам. директора по учебной работе: Крумкач Тамара Николаевна т. 8-017-505-45-33;\n🚸Зам. директора по воспитательной работе: Бегун Лолита Михайловна т. 8-017-505-45-74;\n❗nЗаведующая практикой: Неведомская Наталья Мячеславовна т. 8-017-505-45-38;\n⛪Заведующая отделением: Волотовская Татьяна Евгеньевна т. 8-017-505-45-33;\n💪🏻Руководитель физической культуры: Наквас Евгений Александрович т. 8-017-505-45-25.");

Console.Write(evt.PeerId + " ");

}

}

}

catch (Exception e)

{

Console.WriteLine("Ошибка: " + e);

}

}

private static void SendMessage(long peerId, string message)

{

try

{

VKApi.CallMethod("messages.send", new Dictionary<string, string>

{

{"peer\_id", peerId.ToString()},

{"message", message}

});

}

catch

{

// ignored

}

}

}

}

using System;

using System.Collections.Generic;

using System.Net;

using System.IO;

using Newtonsoft.Json.Linq;

using System.Text;

using System.Collections.Specialized;

namespace LessonBot

{

class Api

{

string Token;

public Api(string token)

{

this.Token = token;

}

public JToken CallMethod(string methodName, Dictionary<string, string> parameters)

{

string address = $"https://api.vk.com/method/{methodName}?";

NameValueCollection postParameters = new NameValueCollection();

foreach(KeyValuePair<string, string> parameter in parameters)

{

postParameters.Add(parameter.Key, parameter.Value);

}

postParameters.Add("access\_token", Token);

postParameters.Add("v", "5.80");

JToken response;

try

{

response = JToken.Parse(Request(address, postParameters).ToString());

} catch { throw new Error.RequestException(); }

if(response["error"] == null)

{

return response["response"];

}

else

{

if(Convert.ToInt32(response["error"]["error\_code"]) == 5)

{

throw new Error.ApiException(response["error"]["error\_code"].ToString(), Error.ErrorCode.AccessToken);

}

else

{

throw new Error.ApiException(response["error"]["error\_code"].ToString(), Error.ErrorCode.Other);

}

}

}

public string Request(string address)

{

using (var client = new WebClient())

{

client.Encoding = Encoding.UTF8;

return client.DownloadString(address);

}

}

public string Request(string address, NameValueCollection parametrs)

{

using (var client = new WebClient())

return Encoding.UTF8.GetString(client.UploadValues(address, parametrs));

}

public class LongPoll

{

public string Ts;

public string Key;

public string Server;

private Api VKApi;

public LongPoll(Api vkApi)

{

VKApi = vkApi;

GetInfoLongPoll();

}

public List<EventObject> Listen()

{

while(true)

{

JToken response = RequestLongPoll();

if (Convert.ToInt32(response["failed"]) == 1)

{

Ts = response["ts"].ToString();

}

else if (Convert.ToInt32(response["failed"]) == 2 || Convert.ToInt32(response["failed"]) == 3)

{

GetInfoLongPoll();

}

else

{

Ts = response["ts"].ToString();

List<EventObject> updates = new List<EventObject>();

foreach (var update in response["updates"])

{

updates.Add(new EventObject(update));

}

return updates;

}

}

}

private JToken RequestLongPoll()

{

NameValueCollection parameters = new NameValueCollection() {

{"act", "a\_check"},

{"key", Key},

{"ts", Ts},

{"wait", "25"},

{"mode", "2"},

{"version", "3"}

};

return JToken.Parse(VKApi.Request($"https://{Server}", parameters));

}

private void GetInfoLongPoll()

{

JToken infoLongPoll = VKApi.CallMethod("messages.getLongPollServer", new Dictionary<string, string>(1) {

{"lp\_version", "3"}

});

Ts = infoLongPoll["ts"].ToString();

Key = infoLongPoll["key"].ToString();

Server = infoLongPoll["server"].ToString();

}

public class EventObject

{

public VKEventType Type = VKEventType.Other;

public long MessageId = 0;

public long PeerId = 0;

public long Time = 0;

public string Message = "";

public JToken Extra;

public JToken Attachments;

public bool FromUser = false;

public bool FromChat = false;

public bool FromGroup = false;

public long UserId = 0;

public long ChatId = 0;

public long GroupId = 0;

public JToken Row;

public EventObject(JToken response)

{

Row = response;

if (Convert.ToInt32(response[0]) == 4)

{

Type = VKEventType.MessageNew;

MessageId = response[1].ToObject<long>();

PeerId = response[3].ToObject<long>();

Time = response[4].ToObject<long>();

Message = response[5].ToString();

Extra = response[6];

Attachments = response[7];

if(PeerId < 0)

{

FromGroup = true;

GroupId = -PeerId;

}

else if(PeerId > 2000000000)

{

FromChat = true;

ChatId = PeerId - 2000000000;

if(Extra != null)

{

UserId = Extra["from"].ToObject<long>();

}

}

else

{

FromUser = true;

UserId = PeerId;

}

}

}

}

}

public enum VKEventType

{

Other,

MessageNew

}

public class Error

{

public class RequestException : Exception { public RequestException() : base("Cannot send request.") { } }

public class ApiException : Exception

{

public ErrorCode Code;

public ApiException(string message, ErrorCode code) : base(message)

{

this.Code = code;

}

}

public enum ErrorCode

{

Other,

AccessToken

}

}

}

}