

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО

Преподаватель департамента программной
инженерии факультета компьютерных
наук

УТВЕРЖДАЮ

Академический руководитель
образовательной программы
«Программная инженерия» профессор
департамента программной
инженерии, канд. техн. наук

_____ Н. К. Чуйкин
«_____» _____ 2019 г.

_____ В. В. Шилов
«_____» _____ 2019 г.

РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА ДЛЯ
ВИЗУАЛИЗАЦИИ ГРАФА ДРУЗЕЙ СОЦИАЛЬНОЙ
СЕТИ "ВКОНТАКТЕ"

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.13-01 51 01-1-ЛУ

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Исполнитель: студент группы БПИ185
_____ А. А. Мануйлов
«_____» _____ 2019 г.

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА ДЛЯ
ВИЗУАЛИЗАЦИИ ГРАФА ДРУЗЕЙ СОЦИАЛЬНОЙ
СЕТИ "ВКОНТАКТЕ"**

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.13-01 51 01-1-ЛУ

Листов 16

Содержание

1	Введение	4
1.1	Наименование программы	4
1.2	Документы, на основании которых ведется разработка	4
2	Назначение и область применения	5
2.1	Назначение программы	5
2.1.1	Функциональное назначение	5
2.1.2	Эксплуатационное назначение	5
2.1.3	Область применения	5
3	Технические характеристики	6
3.1	Постановка задачи на разработку программы	6
3.2	Описание алгоритмов и функционирования программы	6
3.2.1	Описание алгоритмов программы	6
3.2.2	Возможные взаимодействия программы с другими программами .	7
3.3	Описание и обоснование выбора метода организации входных и выходных данных	7
3.3.1	Описание метода организации входных и выходных данных	7
3.3.2	Обоснование выбора метода организации входных и выходных данных	7
3.4	Описание и обоснование выбора состава технических и программных средств	7
3.4.1	Состав технических и программных средств	7
4	Технико-экономические показатели	8
4.1	Предполагаемая потребность	8
4.1.1	Экономические преимущества по сравнению с отечественными и зарубежными аналогами	8
	Список источников	9
5	Приложение: описание классов, полей и методов	10
6	Диаграмма классов	15
7	Лист регистрации изменений	16

1 Введение

1.1 Наименование программы

Программа для визуализации графа друзей социальной сети "Вконтакте"
Programm for Visualisation Graph of Friends in the Social Network "VK"

1.2 Документы, на основании которых ведется разработка

Приказ декана факультета компьютерных наук Национального Исследовательского университета «Высшая школа экономики» № 2.3-02/1012-0 2 «Об утверждении тем, руководителей курсовых работ студентов образовательной программы «Программная инженерия» факультета компьютерных наук» от 10.12.18

2 Назначение и область применения

2.1 Назначение программы

2.1.1 Функциональное назначение

Функциональным назначением программы является построение графа друзей социальной сети «ВКонтакте», демонстрация кластеризованности окружения пользователя, наглядное представление работы силового алгоритма.

2.1.2 Эксплуатационное назначение

Эксплуатационным назначением программы является предоставление пользователю возможности построить кластеризованный граф социальной сети своего аккаунта «ВКонтакте». Приложение может быть использовано для анализа кластерного разделения друзей «ВКонтакте».

2.1.3 Область применения

Данная программа применяется для визуализации структуры социального окружения пользователя и анализа взаимосвязей друзей.

3 Технические характеристики

3.1 Постановка задачи на разработку программы

Задача - разработка программы для построения графа друзей социальной сети «ВКонтакте», его дальнейшей кластеризации и применения к нему силового алгоритма LPA. Пользователь имеет возможность выбрать вершину графа с ее последующим выделением и получить информацию о выбранной вершине (имя выбранного пользователя и фотография). Программа реализована на платформе Windows Forms Application на языке программирования C#.

3.2 Описание алгоритмов и функционирования программы

3.2.1 Описание алгоритмов программы

Программа содержит основные алгоритмы:

1. Силовой алгоритм

Основная идея алгоритма заключается в проведении аналогии между графом на плоскости и физической системой. Цель силового алгоритма - расположение вершин графа максимально удобно для визуального восприятия, этой задаче прекрасно удовлетворяет ситуация минимальной энергии в системе. Связанные вершины притягиваются, как будто соединяющие их ребра это пружины (проводится аналогия с законом Гука), а не связанные вершины отталкиваются, как будто это заряженные противоположными зарядами частицы (аналогия с законом Кулона). Кроме того, все вершины притягиваются к центру поля (вершине пользователя программы), для того, чтобы граф не "разлетался" в стороны.

При запуске алгоритма происходит фиксированное количество итераций (200). Во время каждой итерации вычисляется суммарный вектор силы, действующий на каждую вершину со стороны других и после все вершины сдвигаются на небольшое расстояние в направлении этих сил.

Основное отличие алгоритма от модели реальной физической системы это формулы вычисления сил. Сила притяжения считается по логарифмической зависимости :

$F = \log_{10}\left(\frac{D}{C_2}\right) \cdot C_1$, где D - расстояние между вершинами, C_1 - константа нормировки, C_2 - идеальное расстояние между вершинами ($1/20$ диагонали экрана). Как видно из формулы, использование логарифма не дает вершинам "слипаться" и их притяжение становится более плавным, чем по линейному закону. Сила отталкивания считается по обратной квадратической зависимости от расстояния:

$F = \frac{-C_3}{D^2}$, где C_3 - константа нормировки, D - расстояние между вершинами.

Притяжение к центру же просто пропорционально расстоянию от вершины до центра.

2. Алгоритм кластеризации LPA

Алгоритм распространения меток (Label Propagation Algorithm) является алгоритмом строгой кластеризации (каждая вершина принадлежит только одному кластеру). Этот алгоритм достаточно прост в реализации и в исполнении, и к тому же он достаточно хорошо имитирует знакомства людей в реальной жизни.

Суть алгоритма в том, что на старте каждой вершине присваивается уникальный идентификатор, а затем, пока изменения в метках вершин не прекратятся, происходит следующее: для каждой вершины считается самая часто встречаемая метка среди ее соседей, и затем эта метка присваивается самой вершине. Если таких меток несколько, то из них выбирается случайно одна. Таким образом, все вершины оказываются разделены на кластеры, которые в последствии удобно воспринимать.

3.2.2 Возможные взаимодействия программы с другими программами

Программа предусматривает взаимодействие с vk api для авторизации пользователя и получения токена (ключа доступа), получения списка друзей пользователя, получения списка общих друзей и получения общей информации о пользователе.

3.3 Описание и обоснование выбора метода организации входных и выходных данных

3.3.1 Описание метода организации входных и выходных данных

Входные данные пользователь вводит через дополнительную форму в поля ввода пароля и логина на странице «Вконтакте». Выходными данными является изображение графа в формате .JPEG

3.3.2 Обоснование выбора метода организации входных и выходных данных

Ввод данных в поля ввода «Вконтакте», минуя код программы, - самый безопасный способ авторизации.

3.4 Описание и обоснование выбора состава технических и программных средств

3.4.1 Состав технических и программных средств

Для надёжной и бесперебойной работы программы требуется следующий состав технических средств:

- 1) персональный компьютер, оснащенный 64-разрядным (x64) процессором Pentium с тактовой частотой 1.6 ГГц и выше или аналогичный процессор;
- 2) 4 ГБ оперативной памяти или больше;
- 3) не менее 1 ГБ свободного места на жестком диске;
- 4) видеокарта и монитор, поддерживающие режим Super VGA с разрешением не менее чем 850x600 точек;
- 5) мышь Microsoft Mouse или совместимое указывающее устройство;
- 6) клавиатура;

4 Технико-экономические показатели

4.1 Предполагаемая потребность

4.1.1 Экономические преимущества по сравнению с отечественными и зарубежными аналогами

По сравнению с существующими аналогами данная программа является десктопной, что позволяет пользоваться ей не прибегая к работе браузеров. Ненагруженный интерфейс и интуитивное понимание делают программу легко осваиваемой и простой в использовании даже для начинающего пользователя.

Список источников

- [1] Vuslit [Электронный ресурс] URL: https://vuzlit.ru/949014/avtomaticheskoe_raspolzhenie_vershin_ploskosti (Дата обращения: 09.05.2019, режим доступа: свободный)
- [2] Полное руководство по языку программирования C# 7.0 и платформе .NET 4.7 Metanit [Электронный ресурс] URL: <https://metanit.com/sharp/tutorial/> (Дата обращения: 21.05.2019, режим доступа: свободный)
- [3] Руководство по языку C# MSDN [Электронный ресурс] URL: <https://docs.microsoft.com/ru-ru/dotnet/csharp/> (Дата обращения: 30.04.2019, режим доступа: свободный)
- [4] Руководство разработчикам по API ВКонтакте [Электронный ресурс] URL: https://vk.com/dev/first_guide (Дата обращения: 26.05.2019, режим доступа: свободный)
- [5] Герберт Шилдт. Полное руководство C# 4.0. 2014. -1056.

5 Приложение: описание классов, полей и методов

Таблица 1: Описание классов и их основного функционала

Класс	Функционал
Form1	Основное окно приложения, содержит контрол PictureBox для отрисовки графа, контрол PictureBox для отображения фотографии профиля авторизовавшегося пользователя, контрол PictureBox для отображения фотографии выбранного на графе пользователя, несколько контролов Label для вывода информации о пользователях, контрол menuStrip для выбора необходимой операции. Данный класс содержит все основные методы и события для работы программы.
browser	Вспомогательная форма для авторизации пользователя, получения и передачи в основную форму accessToken
Vertex	Класс, представляющий вершину на плоскости
Edge	Класс, Представляющий ребро на плоскости

Таблица 2: Описание полей класса Edge

Имя	Модификатор доступа	Тип	Назначение
V1	public	Vertex	Первая вершина ребра
V2	public	Vertex	Вторая вершина ребра

Таблица 3: Описание методов класса Edge

Имя	Модификатор доступа	Тип	Аргументы	Назначение
IsVertexPicked	public	bool	-	Проверяет выбрана ли одна из вершин

Таблица 4: Описание полей класса Vertex

Имя	Модификатор доступа	Тип	Назначение
Coord	public	PointF	Координаты вершины

Таблица 5: Описание методов класса Vertex

Имя	Модификатор доступа	Тип	Аргументы	Назначение
IsVertexCommon	public	bool	Vertex v	Проверяет связаны ли вершины

Таблица 6: Описание свойств класса Vertex

Имя	Модификатор доступа	Тип	Доступ	Назначение
Common_Vortexes	public	string[]	get	Проверяет связаны ли вершины
ID	public	string	get	id вершины (пользователя)
Group	public	string	get set	Принадлежность кластеру
IsPicked	public	bool	get set	Выбрана ли вершина

Таблица 7: Описание полей класса browser

Имя	Модификатор доступа	Тип	Назначение
client_id	private	string	id приложения вконтакте
accessToken	public	string	токен доступа
userId	public	string	id авторизовавшегося пользователя
address	private	string	URL для перехода на страницу авторизации

Таблица 8: Описание методов класса browser

Имя	Модификатор доступа	Тип	Аргументы	Назначение
webBrowser1_DocumentCompleted	public	void	-	Событие обработчик загрузки страницы Получает из URL токен и id

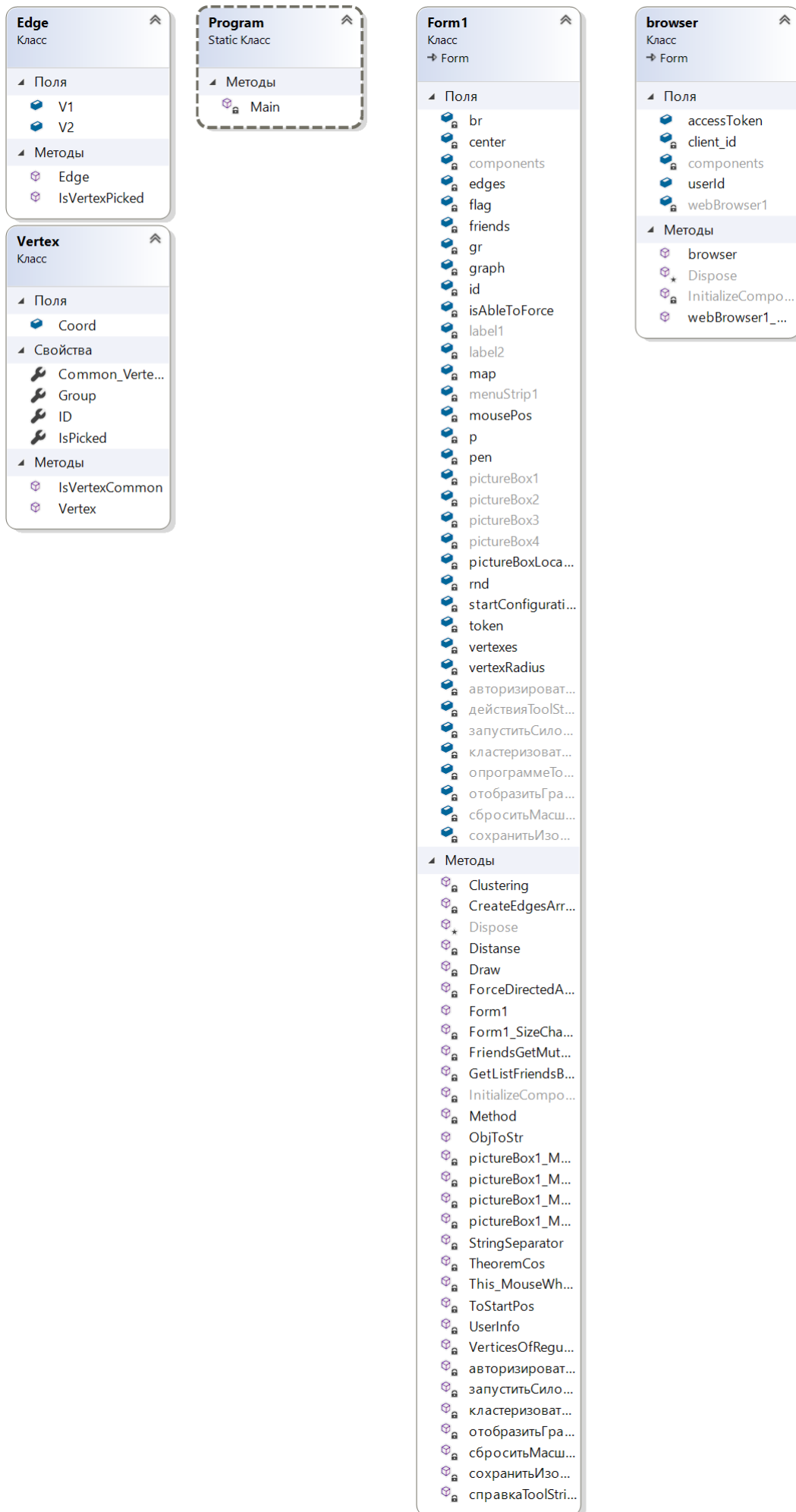
Таблица 9: Описание полей класса Form1

Имя	Модификатор доступа	Тип	Назначение
map	private	Bitmap	Битмап
gr	private	Graphics	Графика для рисования
br	private	SolidBrush	Кисть для рисования вершин
rnd	private	Random	Для произвольного расположения
mousePos	private	PointF	Временная точка для перерисовки
center	private	PointF	Координаты центральной точки
p	private	PointF	Временная точка для перерисовки
picture BoxLocation	private	Point	Расположение контрола pictureBox на форме во время его перемещения
vertexes	private	Vertex[]	Все вершины
edges	private	Edge[]	Все ребра
start Configuration	private	PointF[]	Точки в старт
pen	private	Pen	Ручка для рисования
graph	private	Dictionary <string, string[]>	Граф
friends	private	string[]	Список друзей
vertexRadius	private	float	Радиус отображаемой вершины
isAbleToForce	private	bool	Можно ли применить силовой алгоритм

Таблица 10: Описание методов класса Form1

Имя	Модификатор доступа	Тип	Аргументы	Назначение
ObjToStr	public	string	object obj	Кастует к строке
Draw	private	void	-	Рисует граф на PictureBox
Friends GetMutual	private	Dictionary <string, string[]>	-	Создает список общик друзей
GetList FriendsByID	private	string[]	-	Создает список друзей пользователя
UserInfo	private	string[]	string id	Возвращает общую информацию о id
Form1_ SizeChanged	private	void	-	Событие изменения размера формы
ForceDirected Algorithm	private	void	-	Силовой алгоритм размещения графа на плоскости
pictureBox_ MouseMove	private	void	-	Событие перемещения мышы
pictureBox_ MouseDown	private	void	-	Событие нажатия клавиши мышы
pictureBox_ MouseUp	private	void	-	Событие отжатия клавиши мышы
This_ MouseWheel	private	void	-	Обработчик события прокрутки колесика мышы
pictureBox1_ MouseDouble Click	private	void	-	Обработчик события двойного нажатия мышы
Clustering	private	void	-	алгоритм кластеризации LPA
TheoremCos	private	int	double R, double a	возвращает половину стороны многоугольника
Method	private	void		Обработчик события закрытия второй формы
Create EdgesArray	private	Edge[]	-	Создает массив ребер графа
ToStartPos	private	void	-	Возвращает граф в начальное состояние
VerticesOf Regular	private	PointF[]	int n	Ищет координаты центров кластеров
Distanse	private	double	PointF a, PointF b	Ищет расстояние между точками

6 Диаграмма классов



7 Лист регистрации изменений

[illegible]