|  |  |
| --- | --- |
| ***Подп. и дата*** |  |
| ***Инв. № дубл.*** |  |
| ***Взам. инв. №*** |  |
| ***Подп. и дата*** |  |
| ***Инв. № подл*** | RU.17701729.04.13-01 51 01-1-ЛУ |

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Образовательная программа бакалавриата «Программная инженерия»

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Доцент департамент больших данных и информационного поиска факультета компьютерных наук, к.ф.-м.н.    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. Л. Чернышев «\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2020 г. | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия» профессор департамента программной инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. В. Шилов «\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2020 г. |

**ПРОГРАММА ДЛЯ НАХОЖДЕНИЯ ВРЕМЕНИ НАСЫЩЕНИЯ ДЛЯ ОРИЕНТИРОВАННЫХ МЕТРИЧЕСКИХ ГРАФОВ**

**Программа и методика испытаний**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.04.13-01 51 01-1-ЛУ**

Исполнитель  
студент группы БПИ196  
\_\_\_\_\_\_\_\_\_\_ / Татаринов Н.А. /  
«\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2020 г.

**Москва 2020**

УТВЕРЖДЕНRU.17701729.04.13-01 51 01-1-ЛУ

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл | RU.17701729.04.13-01 51 01-1 |

**ПРОГРАММА ДЛЯ НАХОЖДЕНИЯ ВРЕМЕНИ НАСЫЩЕНИЯ ДЛЯ ОРИЕНТИРОВАННЫХ МЕТРИЧЕСКИХ ГРАФОВ**

**Программа и методика испытаний**

**RU.17701729.04.13-01 51 01-1**

**Листов 11**

**Москва 2020**

**Содержание**

[**1.** **ОБЪЕКТ ИСПЫТАНИЙ** 3](#_Toc41170253)

[**2.** **ЦЕЛЬ ИСПЫТАНИЙ** 3](#_Toc41170254)

[**3. ТРЕБОВАНИЯ К ПРОГРАМММЕ** 3](#_Toc41170255)

[**3.1. Требования к функциональным характеристикам** 3](#_Toc41170256)

[**3.6. Требования к информационной и программной совместимости** 4](#_Toc41170257)

[**3.7. Требования к маркировке и упаковке** 4](#_Toc41170258)

[**3.8. Требования к транспортированию и хранению** 5](#_Toc41170259)

[**5. СРЕДСТВА И ПРОЯДОК ИСПЫТАНИЙ** 6](#_Toc41170260)

[**5.1. Технические средства** 6](#_Toc41170261)

[**5.2. Программные средства** 6](#_Toc41170262)

[**5.3. Порядок проведения испытаний** 6](#_Toc41170263)

[**6. МЕТОДЫ ИСПЫТАНИЙ** 6](#_Toc41170264)

[**6.1. Испытание выполнения требований к функциональным характеристикам и надёжности** 6](#_Toc41170265)

[**6.1.1 Функциональные характеристики** 6](#_Toc41170266)

[**6.1.1.2 Внутренний функционал** 10](#_Toc41170267)

[**6.1.2 Надёжность.** 10](#_Toc41170268)

# **ОБЪЕКТ ИСПЫТАНИЙ**

Объектом испытаний является «Программа для нахождения времени насыщения для ориентированных метрических графов» – программа, реализующая алгоритм построения и визуализации сильно-связных ориентированных метрических графов, моделирующая распространение эпсилон-окрестностей по этим графам и позволяющая пользователю визуально найти время насыщение с некоторым приближением.

Задача программы заключается в обеспечении возможности построить и визуализировать сильно-связный ориентированный метрический граф, построить модель распространения эпсилон-окрестностей по этому графу и предоставить возможность по этой модели определить приблизительное значение времени насыщения.

# **ЦЕЛЬ ИСПЫТАНИЙ**

Целью испытаний является проверка программы на соответствие требованиям к функциональным характеристикам и надежности.

# **3. ТРЕБОВАНИЯ К ПРОГРАМММЕ**

## **3.1. Требования к функциональным характеристикам**

**3.1.1. Требования к составу выполняемых функций**

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

1) Задание сильно-связного ориентированного метрического графа с помощью графического интерфейса и сохранение его описания в файл.

2) Задание сильно-связного ориентированного метрического графа из файла.

3) Анализ входных данных на корректность.

4) Моделирование распространения эпсилон-окрестностей по рёбрам заданного графа.

5) Предоставление возможности визуального приблизительного определения времени насыщения для заданного графа.

**3.1.2. Требования к организации входных данных**

Данные на вход принимаются в виде файла, имя которого вводит пользователь, или с помощью графического интерфейса.

**3.1.3. Требования к организации выходных данных**

Программа должна сохранять заданный графическим путем граф в файл, имя которого выбирает пользователь.

**3.2. Требования к надежности**

Программа должна обрабатывать все исключительные ситуации, возникающие в процессе её работы, должна проверять корректность входных данных.

**3.3. Требования к интерфейсу**

Программа должна иметь оконный интерфейс WPF с возможностью ввода входных данных через файл или через окно и вывода выходного графа в текстовый файл.

Работа программы должна начинаться со стартового окна, которое будет предоставлять возможность выбрать способ задания графа.

**3.4. Условия эксплуатации**

Требуемая квалификация пользователя программы – оператор с базовыми навыками владения ПК и базовыми знаниями в области теории графов.

**3.5. Требования к составу и параметру технических средств**

Для нормального функционирования программы требуется компьютер, оснащенный следующими техническими компонентами:

1. процессор не ниже Intel Core i7-5500U CPU или совместимый с ним с тактовой частотой не ниже 2.4 ГГц;
2. 16 Гб ОЗУ или более;
3. монитор с разрешением не ниже 1920х1080;
4. клавиатура и мышь.

## **3.6. Требования к информационной и программной совместимости**

Для нормального функционирования программы требуется компьютер, оснащенный следующими программными компонентами:

1. операционная система Microsoft Windows 10;
2. библиотека Microsoft .NET Framework 4.8 и выше;
3. среда программирования – Microsoft Visual Studio 2019 и выше.

## **3.7. Требования к маркировке и упаковке**

Приложение располагается на жёстком диске Lenovo(D:), USB-накопителе или посредством сети Интернет. Установка программы не требуется (готова к запуску).

## **3.8. Требования к транспортированию и хранению**

Требования к транспортированию и хранению программы соответствуют стандартным требованиям к транспортированию и хранению соответствующих электронных носителей информации.

**4. ТРЕБОВАНИЯ К ПРОГРАММОЙ ДОКУМЕНТАЦИИ**

**5.1. Состав программной документации**

Состав программной документации должен включать в себя следующие компоненты:

1. Техническое задание (ГОСТ 19.201-78)
2. Программа и методика испытаний (ГОСТ 19.301-79)
3. Пояснительная записка (ГОСТ 19.404-79)
4. Руководство оператора (ГОСТ 19.505-79)
5. Текст программы (ГОСТ 19.401-78)

Вся документация должна быть составлена согласно ЕСПД (ГОСТ 19.101-77, 19.104-78, 19.105-78, 19.106-78 и ГОСТ к соответствующим документам (см. выше)) [3]. Вся документация сдаётся в электронном виде в составе курсовой работы в систему LMS НИУ ВШЭ.

**5.2. Специальные требования к программной документации**

Документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 и ГОСТами к каждому виду документа (см. п. 5.1.);

Пояснительная записка должна быть загружена в систему Антиплагиат через LMS «НИУ ВШЭ».

Документация и программа сдаются в электронном виде в формате .pdf или .docx. в архиве формата .zip или .rar;

За один день до защиты комиссии все материалы курсового проекта:

* техническая документация,
* программный проект,
* исполняемый файл,
* отзыв руководителя
* лист Антиплагиата

должны быть загружены одним или несколькими архивами в проект дисциплины «Курсовой проект 2019-2020» в личном кабинете в информационной образовательной среде LMS (Learning Management System) НИУ ВШЭ.

# **5. СРЕДСТВА И ПРОЯДОК ИСПЫТАНИЙ**

## **5.1. Технические средства**

Для нормального функционирования программы требуется компьютер, оснащенный следующими техническими компонентами:

1. процессор не ниже Intel Core i7-5500U CPU или совместимый с ним с тактовой частотой не ниже 2.4 ГГц;
2. 16 Гб ОЗУ или более;
3. монитор с разрешением не ниже 1920х1080;
4. клавиатура и мышь.

## **5.2. Программные средства**

Для нормального функционирования программы требуется компьютер, оснащенный следующими программными компонентами:

1. операционная система Microsoft Windows 10;
2. библиотека Microsoft .NET Framework 4.8 и выше;
3. среда программирования – Microsoft Visual Studio 2019 и выше.

## **5.3. Порядок проведения испытаний**

Порядок проведения испытаний должен быть следующим:

1. Проверка требований к функциональным характеристикам
2. Проверка требований к надёжности

# **6. МЕТОДЫ ИСПЫТАНИЙ**

## **6.1. Испытание выполнения требований к функциональным характеристикам и надёжности**

### **6.1.1 Функциональные характеристики**

#### 6.1.1.1 Внешний функционал

Все окна программы являются полноэкранными с отсутствием возможности изменить их размер.

1. Стартовое окно.

На стартовом окне (рис. 1) находятся 3 активные подписанные кнопки для выбора дальнейших действий и закрытый блок текста с указанием на необходимость сделать этот выбор. Кнопка «Выйти из программы» отвечает за выход из программы, если это необходимо (такая кнопка есть в каждом окне). Кнопка «Взять граф из файла» отправляет пользователя в окно для получения графа из файла. Кнопка «Нарисовать граф самому» отправляется пользователя в окно графического интерфейса.

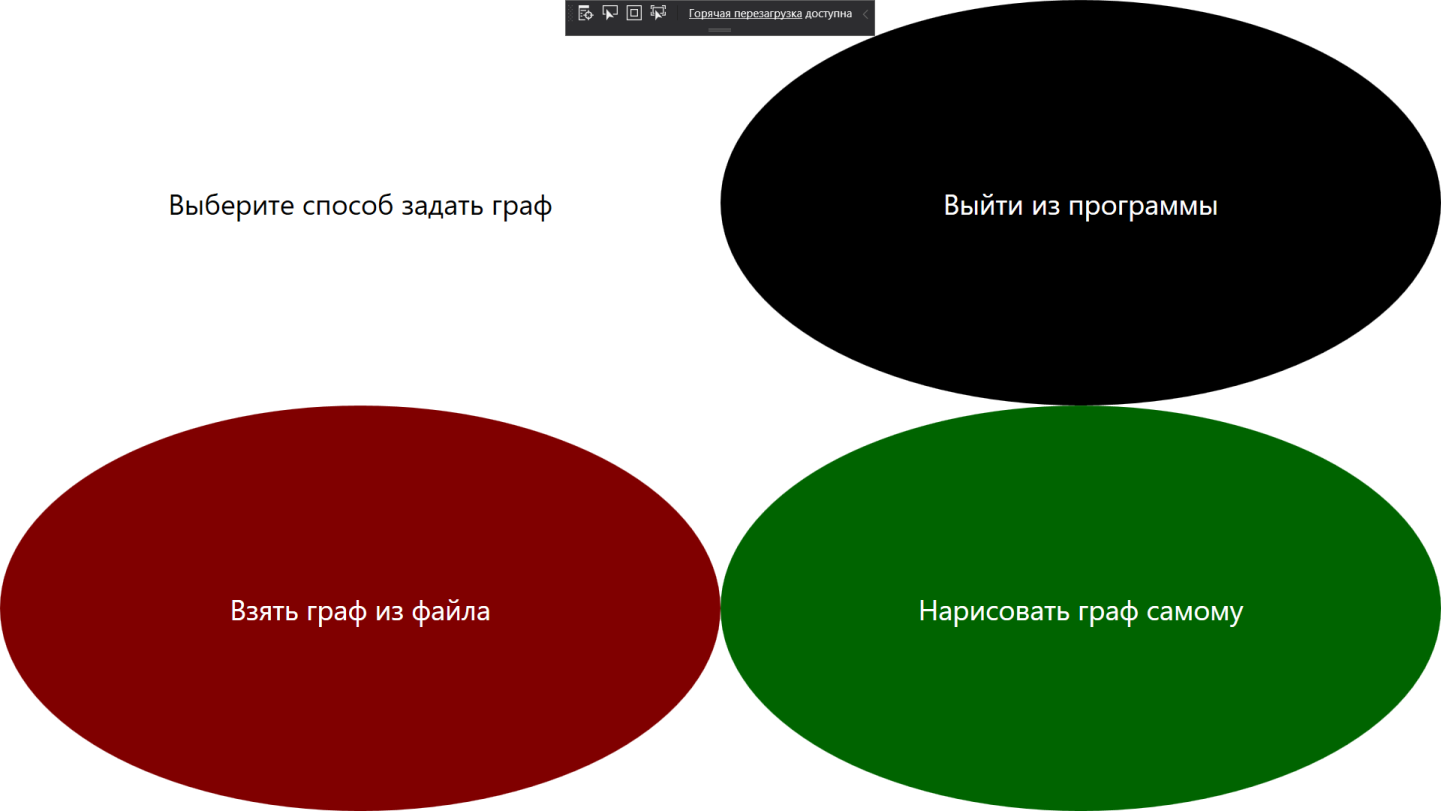


Рисунок 1. Стартовое окно

1. Окно для получения графа из файла.

На окне для получения графа из файла (рис. 2) находятся 2 активные подписанные кнопки и открытый блок текста, в который пользователь должен ввести путь к файлу для чтения из файла (об этом есть соответствующая подсказка). Функционал кнопки «Выйти из программы» описан ранее. Кнопка «Получить граф» запускает попытку чтения графа из указанного файла (обрабатываются все возможные ошибки). Если граф не удалось считать, поверх окна вылетает сообщение с указанием причины ошибки. Если граф успешно считан, пользователь отправляется в окно графического интерфейса.

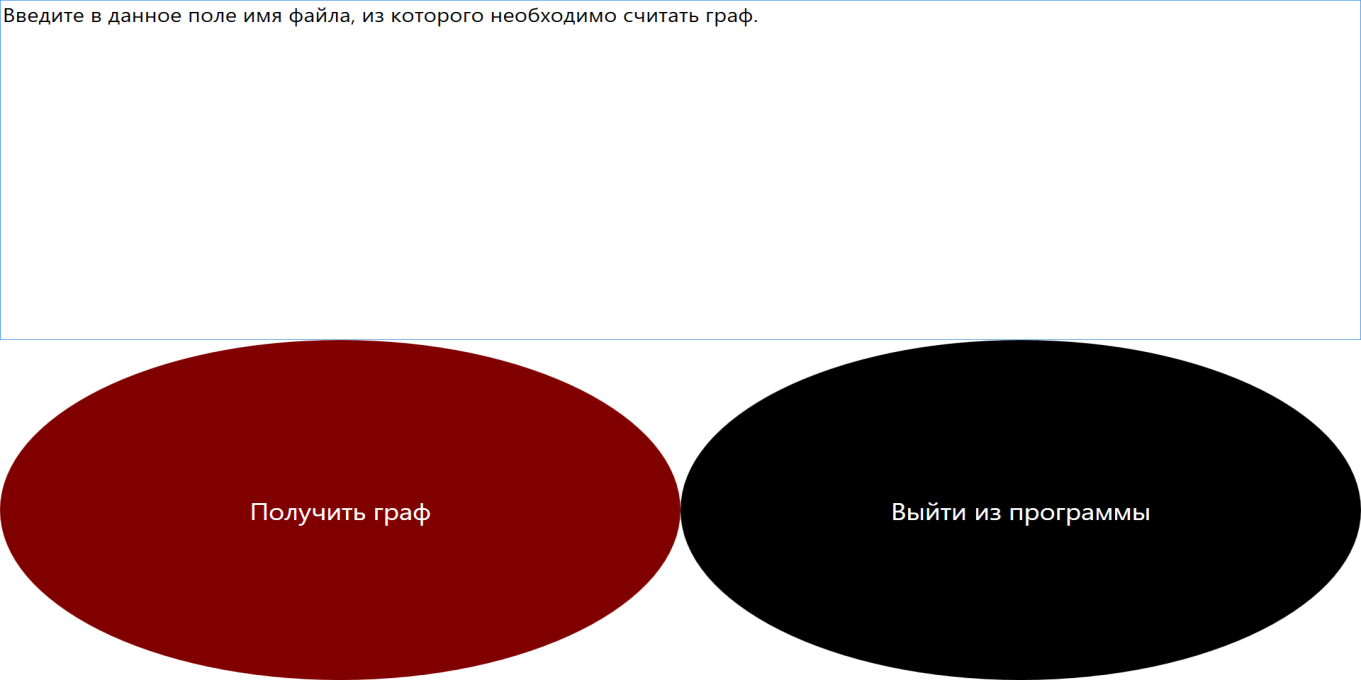


Рисунок 2. Окно для получения графа из файла

1. Окно графического интерфейса.

В окно графического интерфейса (рис. 3) можно попасть 2-мя путями. Первый – из стартового окна. В таком случае, место для рисования графа (объект типа Canvas) будет пустым. Второй – из окна для получения графа из файла. В таком случае на месте для рисования изначально будет находиться граф из файла.

На окне графического интерфейса располагают 4 кнопки, 3 открытых блока (с подсказками) и объект типа Canvas для отображения графа на нём. Функционал кнопки «Выйти из программы» описан ранее. Кнопки «Отмечаем вершины» и «Рисуем рёбра» не могут быть активны одновременно – если неактивна «Отмечаем вершины», то отмечаем вершины, а если «Рисуем р ёбра», то рисуем рёбра. Кнопка «Найти время насыщения» отправляет пользователя в окно моделирования. В первый текстовый блок необходимо записать величину эпсилон-окрестности, во второй – стартовую вершину (для моделирования), в третий – путь к файлу, в который будет сохраняться итоговый граф для моделирования.

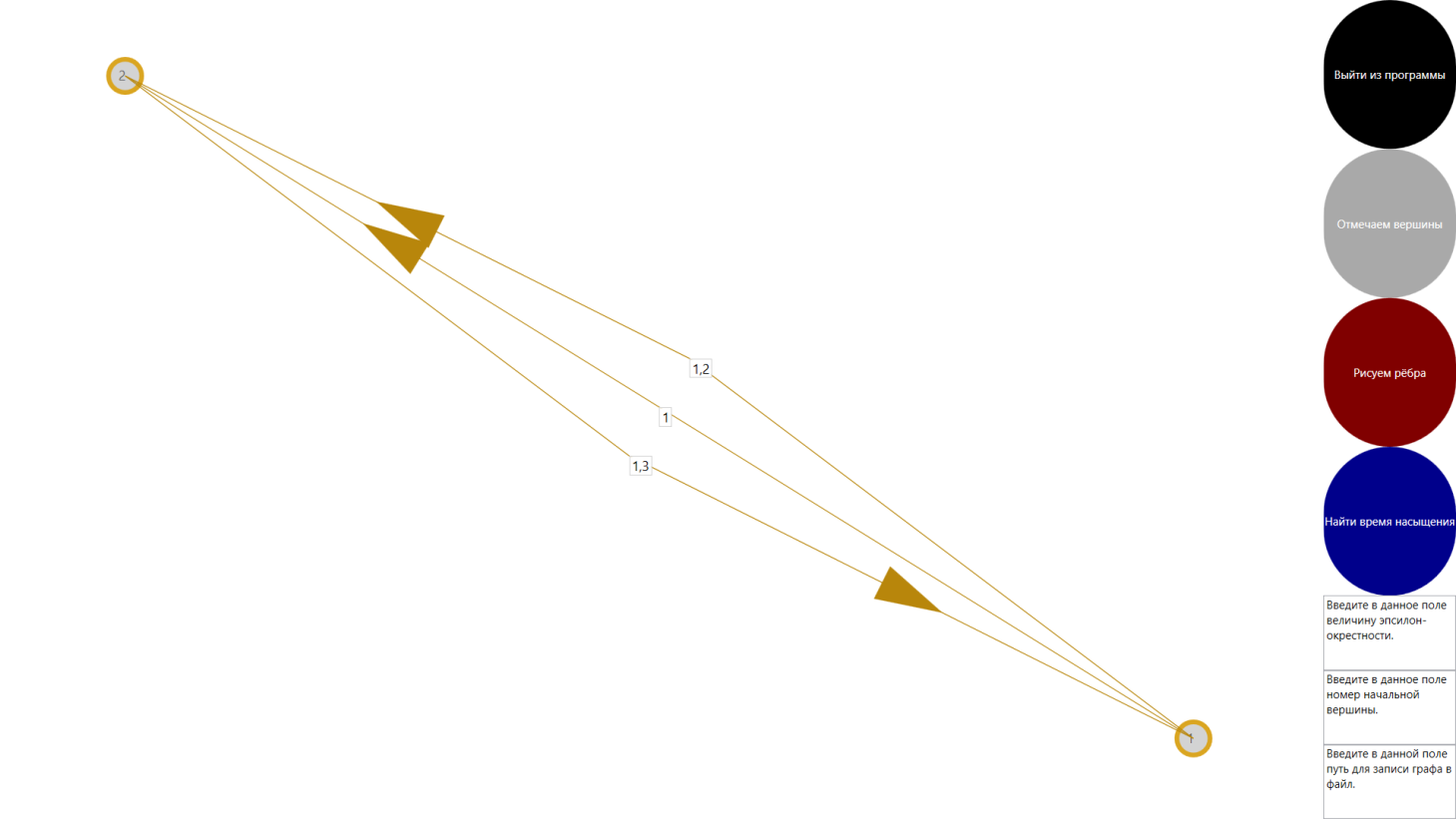


Рисунок 3. Окно графического интерфейса

1. Окно моделирования.

В окне для моделирования (рис. 4) находится объект типа Canvas, в котором происходит моделирование, кнопка «Запустить моделирование», при нажатии на которую запускается/приостанавливается моделирование, кнопка «Выйти из программы» (функционал описан ранее) и закрытый блок текста, в который выводится время движения окрестностей по графу (в единицах длины графа – для времени насыщения).

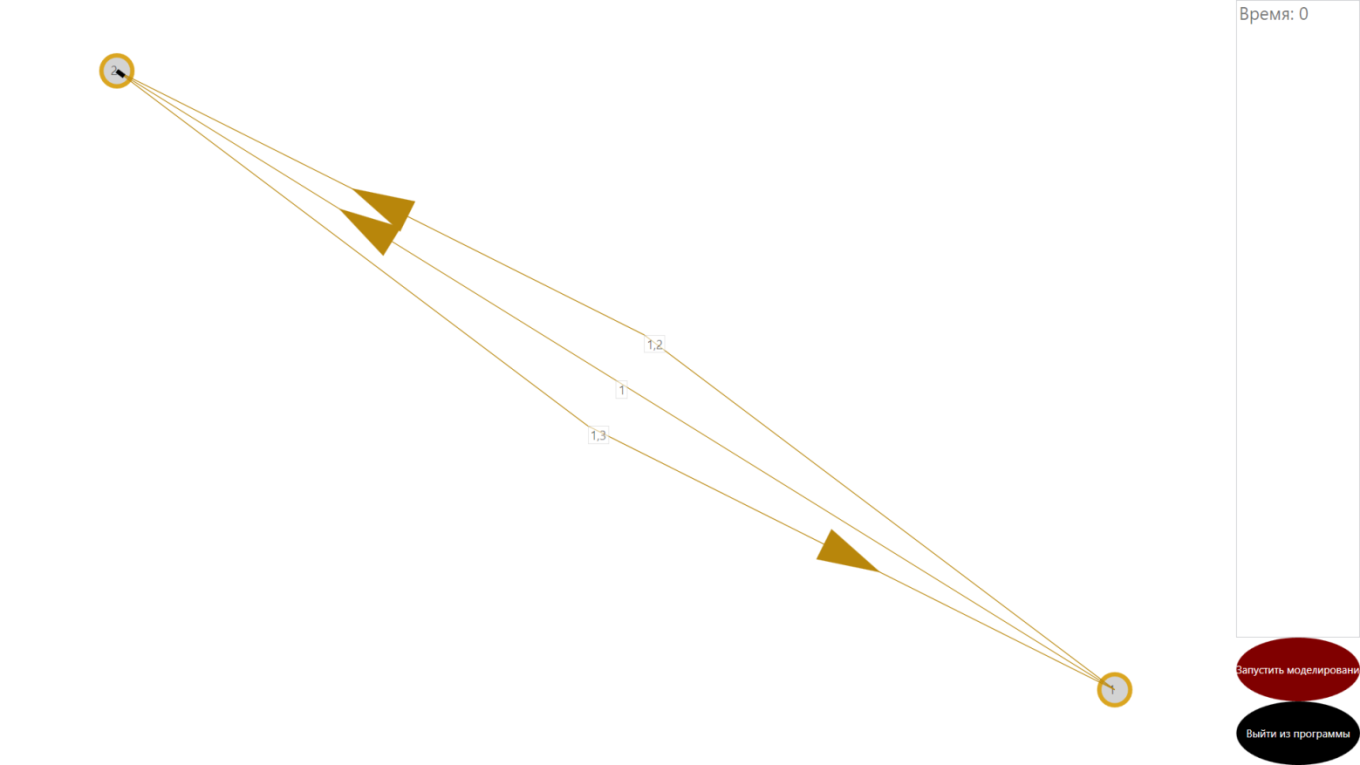


Рисунок 4. Окно моделирования

### **6.1.1.2 Внутренний функционал**

В стартовом окне не происходит никаких проверок – только переход в другие окна или выход.

В окне чтения из файла происходит проверка на существование файла и доступность его для чтения, затем проверка на корректность хранения графа и его считывание в соответствующем случае. Проверка на сильную связность не производится (так как пользователь имеет возможность добавить рёбра в графическом интерфейсе).

В графическом интерфейсе пользователь имеет возможность перемещать и добавлять вершины, добавлять рёбра и изменять их вес. Для запуска моделирования необходимо, чтобы граф был сильно-связным.

В окне моделирования пользователь может запускать и приостанавливать моделирование.

## **6.1.2 Надёжность.**

В процессе программы могут возникать ошибки при работе с файлами, но каждая из них обрабатывается, причём об этой ошибке выводится сообщение на экран. Кроме того, ошибки могут возникать при работе с графом. Первая ошибка – формат хранения графа для чтения. Если он не корректен, то об этом выводится сообщение на экран. Вторая – граф не является сильно-связным на момент запуска моделирования. Об этом также выводится сообщение на экран. Последний вид ошибок – ошибки моделирования (слишком большой время насыщения, или не хватает ресурсов компьютера для дальнейшего продолжения). В этом случае после сообщения о такой ошибке программа закроется через 10 секунд.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Изм. | Номера листов | | | | Всего листов (страниц) в документе | № документа | Входя- щий № сопро- водит. докум. и дата | Под- пись | Дата |
| изменен- ных | заменен- ных | новых | аннулиро- ванных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |