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

Факультет компьютерных наук  
Департамент программной инженерии

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Доцент департамента больших данных и информационного поиска факультета компьютерных наук, к.ф.-м.н.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. Л. Чернышев «\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2019 г. | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия» профессор департамента программной инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. В. Шилов «\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

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

**ПРОГРАММА ДЛЯ МОДЕЛИРОВАНИЯ ДВИЖЕНИЯ ТОЧЕК НА**

**ОРИЕНТИРОВАННОМ МЕТРИЧЕСКОМ ГРАФЕ, С УСЛОВИЕМ**

**СИНХРОНИЗАЦИИ В ВЕРШИНАХ**

**Пояснительная записка**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.04.13-01 ПЗ 01-1-ЛУ**

Исполнитель

студент группы БПИ196

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / А. А. Баранова /

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г.

**Москва 2019**

УТВЕРЖДЕНRU.17701729.04.13-01 ПЗ 01-1-ЛУ

**ПРОГРАММА ДЛЯ МОДЕЛИРОВАНИЯ ДВИЖЕНИЯ ТОЧЕК НА**

**ОРИЕНТИРОВАННОМ МЕТРИЧЕСКОМ ГРАФЕ, С УСЛОВИЕМ**

**СИНХРОНИЗАЦИИ В ВЕРШИНАХ**

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

**Пояснительная записка**

**RU.17701729.04.13-01 ПЗ 01-1**

**Москва 2019**

**Содержание**

[1. Введение 4](#_Toc40398297)

[1.1. Наименование программы 4](#_Toc40398298)

[1.2. Краткая характеристика области применения 4](#_Toc40398299)

[2. Основания для разработки 5](#_Toc40398300)

[2.1 Документы, на основании которых ведется разработка 5](#_Toc40398301)

[2.2 Наименование темы разработки 5](#_Toc40398302)

[3. Назначение разработки 6](#_Toc40398303)

[3.1. Функциональное назначение 6](#_Toc40398304)

[3.2. Эксплуатационное назначение 6](#_Toc40398305)

[4. Требования к программе 7](#_Toc40398306)

[4.1. Требования к функциональным характеристикам 7](#_Toc40398307)

[4.1.1. Требования к составу выполняемых функций 7](#_Toc40398308)

[4.1.2. Требования к организации входных данных 8](#_Toc40398309)

[4.1.3. Требования к организации выходных данных 8](#_Toc40398310)

[4.2. Требования к интерфейсу 8](#_Toc40398311)

[4.3. Требования к надежности 9](#_Toc40398312)

[4.3.1. Требования к обеспечению надежного (устойчивого) функционирования программы 9](#_Toc40398313)

[4.3.3. Отказы из-за некорректных действий оператора 9](#_Toc40398314)

[4.4. Условия эксплуатации 9](#_Toc40398315)

[4.4.1. Климатические условия 9](#_Toc40398316)

[4.4.2. Требовaния к квалификации оператора 9](#_Toc40398317)

[4.5. Требования к составу и параметру технических средств 9](#_Toc40398318)

[4.6. Требования к информационной и программной совместимости 10](#_Toc40398319)

[4.7. Требования к маркировке и упаковке 10](#_Toc40398320)

[4.8. Требования к транспортированию и хранению 10](#_Toc40398321)

[5. Требования к программной документации 11](#_Toc40398322)

[5.1. Состав программной документации должен включать в себя следующие компоненты: 11](#_Toc40398323)

[5.2. Специальные требования к программной документации 11](#_Toc40398324)

[6. Технико-экономические показатели 12](#_Toc40398325)

[6.1. Ориентировочная экономическая эффективность 12](#_Toc40398326)

[6.2. Предполагаемая потребность 12](#_Toc40398327)

[6.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами 12](#_Toc40398328)

[7. Стадии и этапы разработки 13](#_Toc40398329)

[7.1. Необходимые стадии разработки, этапы и содержание работ 13](#_Toc40398330)

[7.2. Сроки и исполнители 14](#_Toc40398331)

[8. Порядок контроля и приемки 15](#_Toc40398332)

[8.1. Виды испытаний 15](#_Toc40398333)

[8.2. Общие требования к приемке работы 15](#_Toc40398334)

[9. Источники 16](#_Toc40398335)

[ПРИЛОЖЕНИЕ 1 17](#_Toc40398336)

[ПРИЛОЖЕНИЕ 2 18](#_Toc40398337)

# **Введение**

* 1. **Наименование программы**

Наименование программы – «Программа для моделирования движения точек на ориентированном метрическом графе, с условием синхронизации в вершинах»

«The Program for Modeling the Movement of Points on Directed Metric Graph, with the Condition of Synchronization at the Vertices».

* 1. **Документы, на основании которых ведется разработка**

Основанием для разработки является приказ декана факультета компьютерных наук И.В. Аржанцева "Об утверждении тем, руководителей курсовых работ студентов образовательной программы «Программная инженерия» факультета компьютерных наук" № 2.3-02/1112-04 от 11.12.2019.

1. **НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ**
   1. **Назначение программы**
      1. **Функциональное назначение**

Функциональным назначением разрабатываемого приложения является работа с выбранным пользователем из архива ранее созданных, либо новым, построенным им в специальном редакторе, сильно связным ориентированным метрическим графом с целью моделирования движения на нем точек, с условием синхронизации в вершинах, а также получения минимального анализа поведения этих точек и визуализации происходящих на графе процессов.

* + 1. **Эксплуатационное назначение**

Программа может быть использована преподавателями, студентами или исследователями, работающими в различных областях математики и информатики при рассмотрении разнообразных задач на графах, или нейробиологии - при моделировании различных реальных процессов в нейронных сетях с использованием данной математической модели. Таким образом, продукт разработки позволит решать задачи, возникающие при составлении теоретических моделей в ходе научных исследований.

* 1. **Краткая характеристика области применения**

Программа предназначена для моделирования движения точек на сильно связных ориентированных метрических графах, с условием синхронизации в вершинах. Программный инструмент позволяет строить графы и моделировать на них движение точек двух разных типов: стандартного (синхронизация в вершинах по установленным порогам) и модели песка (синхронизация по степени вершины); получать визуализацию процесса распространения точек на графе, изображение зависимости числа точек от времени в виде графика, изображение графика распределения размеров лавин при моделировании движения песка; сохранять анимацию процесса в формате GIF-изображения.

Данная программа несет научно-образовательный характер и может использоваться исследователями, работающими в различных областях математики и информатики при рассмотрении разнообразных задач на графах или изучении абелевой модели песка, или нейробиологии – при моделировании различных процессов в нейронных сетях с использованием данной математической модели.

1. **ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ**
   1. **Постановка задачи на разработку программы**

Разрабатываемая программа должна:

1. иметь отдельное поле для построения ориентированного метрического графа без петель, осуществлять на нем отрисовку графа и его редактирование;
2. выполнять сохранение построенного графа в файл формата .dgmm [см. Приложение 2];
3. выполнять открытие ранее созданного в этой программе графа из файла формата .dgmm [см. Приложение 2];
4. производить моделирование движения точек с условием синхронизации в вершинах по выбранному правилу, демонстрировать анимацию этого движения, предоставлять пользователю дополнительную запрошенную информацию.
   1. **Описание алгоритма и функционирования программы**
      1. **Алгоритм моделирования движения точек на ориентированном метрическом графе с условием синхронизации в вершинах**

Возможность моделирования движения точек предоставляется членами класса MovementModeling. В класс передаются сильно связный граф, тип моделирования, список выбранных дополнительных действий, а также ссылка на поверхность рисования и экземпляр GraphDrawing для отрисовки графа.

После вызова метода, запускающего движение, производится обработка переданных данных и непосредственно запуск моделирования. Далее управление передается обработчикам событий класса.

По тику основного таймера происходит обновление состояния системы и данных:

1. Обновление состояния и изображения:
2. Вызов события, оповещающего о начале обновления.
3. Исследование движущихся в данный момент точек на предмет окончания движения.
4. Исследование вершин на готовность выпустить точки.
5. Запуск новых точек.
6. Проверка состояния системы на предмет окончания движения и запуск события, оповещающего об этом, если движение окончено.
7. Обновление графиков (в случае их построения), если после действий п.1 общее количество точек на графе было изменено.
8. Сохранение текущего изображения в объект GifBitmapEncoder для возможности последующего сохранения процесса движения точек (в случае, если данная опция была выбрана).

Процесс можно приостановить нажатием кнопки Stop верхнего меню и возобновить нажатием кнопки Continue, а также полностью сбросить, кликнув по кнопке Reset.

* + 1. **Алгоритм проверки ориентированного метрического графа на сильную связность**

Проверка графа на сильную связность осуществляется с использованием упрощенного алгоритма Косарайю для поиска областей сильной связности в ориентированном графе. В процессе работы алгоритма выполняются следующие шаги:

1. С первой вершины графа запускается поиск в глубину.
2. По мере выполнения поиска в глубину заполняется массив bool[] visited. После посещения вершины элементу с ее индексом в массиве присваивается значение true.
3. По окончанию первого обхода, если visited содержит хотя бы одно значение false, метод проверки возвращает false.
4. Если при первом обходе все вершины были посещены, граф инвертируется.
5. На инвертированном графе с первой вершины вызывается поиск в глубину.
6. См. п.2.
7. По окончанию обхода, если visited содержит хотя бы одно значение false, метод проверки возвращает false.
8. Если все вершины при втором обходе были посещены, метод проверки возвращает true.

Данный алгоритм определяет, является ли ориентированный граф 𝐺 = (𝑉, 𝐸) сильно связным, за определенное время 𝑂(|𝑉| + |𝐸|).

Его сложность связана со сложностью алгоритма поиска в глубину, который, в

свою очередь, должен быть использован дважды, а также со сложностью

нахождения обратного графа. Оба эти алгоритма имеют линейную сложность и

именно поэтому Алгоритм Косарайю работает за линейное время.

* + 1. **Алгоритм парсинга введенных значений длин ребер.**

Парсинг введенных математических вражений производится путем перевода введенной строки в польскую запись (польскую нотацию) [см. Список литературы №], особенность которой состоит в том, что такая запись может быть интерпретирована без неоднозначности. В ходе парсинга выражения выполняются следующие действия:

1. На вход подается строка, содержащая введенное выражение.
2. Производится анализ расстановки скобок и перевод букв в нижний регистр.
3. Инициализируется стек, в котором будут храниться операторы, и объект StringBuilder, в который будет собираться выражение в польской нотации.
4. В цикле по позиции в строке выполняется лексический и синтаксический анализ текущей части выражения:
   1. Лексический анализ (получает на вход строку с выражением и ссылку на текущую позицию в строке):
      1. Проверяет, является ли текущий символ оператором:

Если это оператор, определяет его арность и переходит на следующую позицию, возвращает строковое представления оператора в особом формате.

* + 1. Проверяет, является ли текущий символ буквой или одной из поддерживаемых функций или констант.
       1. Передвигается далее по строке, пока она не закончилась и текущий символ – буква. В цикле добавляет в строку-результат метода текущий символ.
       2. Если полученная строка является одной из поддерживаемых функций, возвращает ее строковое представление в особом формате.
       3. Если полученная строка является одной из поддерживаемых констант, возвращает ее строковое представление в особом формате.
       4. Выбрасывает ArgumentException с сообщением «Unknown token».
    2. Проверяет, является ли текущий символ цифрой или десятичным разделителем.
       1. Если текущий символ – цифра, передвигается далее по строке, пока она не закончилась и текущий символ – цифра. В цикле добавляет в строку-результат метода текущий символ.
       2. Если текущий символ – десятичный разделитель, очищает строку-результат.
       3. Если указатель на позицию в строке не вышел за ее пределы и текущий символ – десятичный разделитель, добавляет десятичный разделитель в строку и передвигается далее по строке, пока она не закончилась и текущий символ – цифра. В цикле добавляет в строку-результат метода текущий символ.
       4. Читает научную (суффиксальную) нотацию.
       5. Возвращает полученное в строке-результате число в специальном формате.
    3. Выбрасывает ArgumentException с сообщением "Unknown token in expression".
  1. Синтаксический анализ (получает на вход строку-результат лексического анализа, ссылку на StringBuilder, в котором собирается выражение, и ссылку на стек):

По первому символу, указывающему на вид результата лексического анализа, производит следующие действия:

* + 1. Если это число, просто добавляет его в строку выражения.
    2. Если это функция или открывающая скобка – помещает в стек.
    3. Если это закрывающая скобка, вынимает элементы из стека, пока не дойдет до открывающей скобки и добавляет каждый элемент в строку-результат. Добавляет следующий элемент стека в строку-результат, если это указатель на функцию.
    4. Если это не число, функция или скобка, вынимает объекты из стека, пока их приоритетность меньше либо равна приоритетности последнего элемента стека, добавляет вынимаемые элементы в строку-результат. После помещает исходную (полученную после лексического анализа) строку в стек.
    5. Возвращает собранную строку-результат.

1. Вынимает все собранные в стеке операторы и добавляет их в результирующее выражение.
2. Производит подсчет выражения в польской нотации:
   1. Определяет стек операндов
   2. В цикле, пока позиция в строке не выходит за ее пределы производится лексический и синтаксический анализ всего выражения
      1. Лексический анализ (получает на всход строку с выражением в польской нотации и ссылку на указатель позиции в строке)
         1. Добавляет в строку-результат метода текучий символ и переходит к следующему
         2. Проходит по выражения до следующего маркера, добавляя символы в строку-результат.
         3. Возвращает полученную строку
      2. Синтаксический анализ (получает на вход ссылку на стек оперендов и результат лексического анализа)
         1. Если лексический анализ вернул операнд, добавляем его в стек.
         2. Если это операция, в зависимости от арности применяем ее к одному или двум последним операндам в стеке.
         3. Добавляет в стек результат вычислений.
   3. В результате в стеке остается единственный элемент – результат вычисления, введенного пользователем выражения.
   4. **Обоснование выбора алгоритма решения задачи.**

Алгоритмы 1-4 являются алгоритмами построения по формулам, описывающими зубчатые колеса, для них нет точных алгоритмов построения поэтому они были написаны самостоятельно с учетом механики фреймворка.

Алгоритмов для определения пересечения цилиндров не так много (мною было найдено 3) и все они отличаются сложным математическим аппаратом. В связи с этим был выбран тот, который использует более простой математический аппарат. Это позволило написать код, который легко отлаживать и поддерживать.

* 1. **Возможные взаимодействия программы с другими программами**

В целом программа работает самостоятельно. Однако для сохранения, редактирования записи и печати решения задачи необходим текстовый редактор, например Microsoft Word. Можно скопировать решение задачи из программы в графический редактор и выполнять необходимые действия по сохранению, редактированию и печати.

1. **Список литературы.**
2. Chapter 6. Neuronal Networks: A Discrete Model // Mathematical Concepts and Methods in Modern Biology: Using Modern Discrete Models / Robeva, R., Hodge, T. (редакторы). – USA: Academic Press, 2013. – 179-211 с.
3. Yerzhan Kalzhani. Проект MathParserTK: Math Parser .NET C# [Электронный ресурс] / GitHub. Режим доступа: URL: https://github.com/kirnbas/MathParserTK, свободный. (дата обращения: 15.05.20).
4. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
5. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с.
6. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с.
7. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с.
8. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с.
9. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с.
10. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с.
11. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с.
12. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с.
13. Большакова Е. А. Свойство бисвязности ориентированного графа: Дипломная работа, Санкт-Петербургский государственный университет, Математико-механический факультет, Кафедра исследования операций / Большакова Е. А., научный руководитель: Доктор физ.-мат. наук, профессор Романовский Иосиф Владимирович – Санкт-Петербург, СПбГУ, 2016. – 12-16 с.
14. Евстигнеев В. А. Толковый словарь по теории графов в информатике и программировании / Евстигнеев В.А., Касьянов В.Н. – Россия, Наука, Сибирское предприятие РАН, 1999.
15. Калинин, Н. С. Модель пересыпания песка и дивизоры на графах [Электронный ресурс]: курс лекций — Электрон. дан. — Дубна: Летняя школа «Современная математика», 2017. — Режим доступа: URL: <https://www.mccme.ru/dubna/2017/courses/kalinin.html>, свободный. (дата обращения: 14.05.20).

**ПРИЛОЖЕНИЕ 1**

**ТЕРМИНОЛОГИЯ**

Ниже приведен список необходимых терминов для ознакомления.

**Вершина** – базовое понятие. Точка, где могут сходиться/выходить рёбра и/или дуги.

**Ребро** – базовое понятие. Ребро соединяет две вершины графа.

**Дуга** – ребро, имеющее направление. Упорядоченная пара вершин (v, w), где вершину v называют началом, а w - концом дуги. Можно сказать, что дуга v → w ведет от вершины v к вершине w, при этом вершина w смежная с вершиной v.

**Граф** – базовое понятие. Включает множество вершин и множество рёбер, являющееся подмножеством декартова квадрата множества вершин (то есть каждое ребро соединяет ровно две вершины).

**Орграф** – ориентированный граф G = (V, E) есть пара множеств, где V - множество вершин (узлов), E – множество дуг.

**Сильно связный граф** – ориентированный граф, в котором все вершины сильно связаны, т. е. существует путь из первой во вторую и из второй в первую и т. д.

**Метрический граф** – граф, каждое ребро которого имеет заданное вещественное время прохождения.

**Инцидентность** — отношение между ребром (дугой) и его концевыми вершинами, т.е. ребро e = (a,b) инцидентно вершинам a и b и вершины a, b инцидентны ребру e = (a,b).

**ПРИЛОЖЕНИЕ 2**

**Описание формата .dgmm**

Файл формата .dgmm содержит информацию об орграфе, созданном в программе, в сереализованном с использованием XML сериализации виде.

Программа позволяет открыть файл формата .dgmm, и при этом осуществляет корректное открытие только файлов данного формата, созданных или измененных в этой программе.

**ПРИЛОЖЕНИЕ 1**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ, СТРУКТУР, ИНТЕРФЕЙСОВ И ПЕРЕЧИСЛЕНИЙ**

Таблица 1

|  |  |
| --- | --- |
| Класс/Структура/  Интерфейс/Перечисление | Назначение |
| MainWindow | Класс наследник Form, представляющий главное окно приложения. Содержит обработчики пользовательских событий, в нем так же содержатся и взаимодействуют объекты других классов и поля и методы необходимые для взаимодействия этих объектов. |
| Arc | Структура, представляющая дуги графа. Содержит индексы начальной и конечной вершины в орграфе, а также длину дуги. |
| Vertex | Структура, представляющая вершину орграфа. Содержит координаты вершины на плоскости. |
| Digraph | Класс, представляющий ориентированный метрический граф. Содержит списки вершин и дуг орграфа, списки значений порогов вершин, периодов восстановления и начальных состояний, а также методы для корректного удаления и добавления вершин и дуг в граф. |
| DigraphChangedEventArgs | Класс наследник EventArgs для реализации стандартного шаблона событий класса Digraph. События типа EventHandler<DigraphChangedEventArgs> оповещают об изменениях орграфа, таких как добавление и удаление вершин и ребер. Класс содержит индекс добавленного/удаленного элемента. |
| ICommand | Интерфейс, представляющий команды, совершаемые пользователем. Содержит методы выполнения действия и выполнения действия, обратному к этому действию. |
| AddArcCommand | Класс, реализующий интерфейс ICommand. Содержит методы для добавления дуги в граф и ее корректного удаления. |
| AddVertexCommand | Класс, реализующий интерфейс ICommand. Содержит методы для добавления вершины в граф и ее корректного удаления. |
| ChangeArcLengthCommand | Класс, реализующий интерфейс ICommand. Содержит методы для изменения длины дуги графа и ее возвращения к предыдущему значению. |
| ChangeColorCommand | Класс, реализующий интерфейс ICommand. Содержит методы для изменения цвета отрисовки дуг и вершин графа и возвращения этого цвета к предыдущему значению. |
| EraseArcCommand | Класс, реализующий интерфейс ICommand. Содержит методы для удаления дуги из графа и ее корректного возвращения. |
| EraseVertexCommand | Класс, реализующий интерфейс ICommand. Содержит методы для удаления вершины из графа и ее корректного возвращения. |
| MoveDigraphCommand | Класс, реализующий интерфейс ICommand. Содержит методы для перемещения графа на плоскости и его корректного в предыдущую позицию. |
| MoveVertexCommand | Класс, реализующий интерфейс ICommand. Содержит методы для перемещения вершины графа и ее корректного возвращения в предыдущую позицию. |
| ResizeDigraphCommand | Класс, реализующий интерфейс ICommand. Содержит методы для изменения размера графа и его корректного возвращения к предыдущему размеру. |
| CommandsManager | Класс, содержащий два стека команд: стек выполненных и стек отмененных. Реализует возможность отмены последних совершенных пользователем изменений при построении графа и возвращения отмененных изменений. |
| ConnectivityCheck | Класс, предоставляющий методы для проверки ориентированного графа на сильную связность. |
| GraphDrawing | Класс, предоставляющий методы для отрисовки графа и его элементов во всех необходимых для работы приложения формах. |
| MathParser | Класс, предоставляющий методы для парсинга строкового представления математического выражения. |
| ChartWindow | Класс наследник Form, представляющий окно, в которомво время моделирования движения отображается запрошенный пользователем графк. |
| MovementModeling | Класс, предоставляющий возможность моделирования движения точек на ориентированном метрическом графе с условием синхронизации в вершинах. |
| MovementTickEventArgs | Класс наследник EventArgs, используется для реализации стандартного шаблона событий класса MovementModeling. Содержит число, указывающее прошедшее с начала движения количество миллисекунд. |
| MovementModelingMode | Перечисление, содержащее набор дополнительных действий, запрашиваемых у программы при моделировании движения. |
| MovementModelingType | Перечисление, содержащее набор типов моделирования движения. |
| SandpileChartType | Перечисление, содержащее набор типов графиков, построение которых возможно при моделировании движения песка. |
| DigraphBuilding | Класс, предоставляющий статические методы для удаления вершин и ребер. |
| DigraphInformationDemonstration | Класс, представляющие статические методы для отображении различной информации о графе в таблицах DataGridView. |
| RandomDigraphGeneratorForm | Класс наследник Form, представляющий окно для выбора параметров построения случайного графа, реализующий построение этого графа. |
| SquareLatticeForm | Класс наследник Form, представляющий окно для выбора параметров построения квадратной решетки, реализующий построение такого графа. |
| TriangularLatticeForm | Класс наследник Form, представляющий окно для выбора параметров построения треугольной решетки, реализующий построение такого графа. |

**ПРИЛОЖЕНИЕ 2**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ МЕТОДОВ И СВОЙСТВ**

Таблица 2.1

Описание полей методов и свойств класса MainWindow.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| AboutApp | private | string | Строка с краткой информацией о программе и разработчике. | |
| commandsManager | private | CommandsManager | Предоставляет возможность отмены и возврата действий пользователя при построении графа. | |
| digraph | private | Digraph | Ориентированный метрический граф, с которым ведется работа. | |
| resizeCoefficient | private | double | Коэффициент масштабирования при изменении размера графа. | |
| graphDrawing | private | GraphDrawing | Предоставляет методы для отрисовки графа и его компонент. | |
| isControlPressed | private | bool | Показывает, нажата ли в данный момент клавиша Control. | |
| isOnMovement | private | bool | Показывает, находится ли программа в данный момент в режиме моделирования движения. | |
| isPressed | private | bool | Показывает, зажата ли в данный момент вершина на поле. | |
| movement | private | MovementModeling | Моделирует движение точек на графе. | |
| movedVertex | private | Vertex | Передвигаемая вершина. | |
| movedVertexIndex | private | int | Индекс передвигаемой вершины. | |
| Rnd | private | Random | Генерирует случайные значения. | |
| ticks | private | DateTime | Показывает, как долго была зажата вершина. | |
| vEnd | private | int | Индекс конечной вершины, выбранной для добавления ребра. | |
| vStart | private | int | Индекс начальной вершины, выбранной для добавления ребра. | |
| xCoefficient | private | int | Коэффициент смещения графа по оси Х. | |
| yCoefficient | private | int | Коэффициент смещения графа по оси Y. | |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| AboutToolStripMenuItem\_Click | private | void | Object, EventArgs | Предоставляет краткую справку о программе и разработчике. |
| AddVertexToGridAdjacencyMatrix | private | void | int | Добавляет вершину в матрицу смежности. |
| AddVertexToGridParameters | private | void | int | Добавляет вершину в матрицу параметров. |
| ArcLength\_TextChanged | private | void | Object, EventArgs | Делает поле для ввода длины недоступным, если выбранного ребра не существует. |
| ArcName\_TextChanged | private | void | Object, EventArgs | Определяет существует ли выбранная дуга и выводит ее длину или сообщение об ошибке. |
| ArcsColorDialogOpen\_Click | private | void | Object, EventArgs | Открывает ColorDialog для выбора цвета отрисовки дуг. |
| ArcsColorPanel\_Click | private | void | Object, EventArgs | Переводит фокус на панель. |
| ArcsColorPanel\_Enter | private | void | Object, EventArgs | Делает кнопку открытия ColorDialog видимой. |
| ArcsColorPanel\_Leave | private | void | Object, EventArgs | Делает кнопку открытия ColorDialog невидимой. |
| BasicTypeCheckBox\_CheckedChanged | private | void | Object, EventArgs | Поддерживает один из типов моделирования выбранным. |
| Build\_Click | private | void | Object, EventArgs | Открывает пустой редактор для создания нового графа. |
| ChangeDrawingElementsState | private | void | bool | Меняет видимость элементов управления, относящихся к редактору графов. |
| ChangeMainMenuState | private | void | bool | Меняет видимость элементов управления, относящихся к главному меню. |
| ChangeWindowStateForMovementModeling | private | void | bool | Подготавливает окно к моделированию движения. |
| ChartCheckBox\_CheckedChanged | private | void | Object, EventArgs | Отображает или скрывает варианты графиков для модели песка. |
| CheckConnectivity | private | bool | – | Проверяет граф на валидность. |
| ClearButton\_Click | private | void | Object, EventArgs | Удаляет созданный граф и возвращает редактор к исходному состоянию. |
| CursorButton\_Click | private | void | Object, EventArgs | Выбирает курсор как текущий инструмент. |
| DataToolStripMenuItem\_Click | private | void | Object, EventArgs | Открывает диалог для сохранения данных графа. |
| DeleteButton\_Click | private | void | Object, EventArgs | Выбирает ластик как текущий инструмент. |
| DigraphOpenFileDialog | private | OpenFileDialog | – | Создает OpenFileDialog для открытия графа из файла. |
| Down\_Click | private | void | Object, EventArgs | Двигает граф вниз. |
| DrawingSurface\_MouseClick | private | void | Object, MouseEventArgs | В зависимости от выбранного инструмента и состояния программы производит определенные действия в построении и редактировании графа. |
| DrawingSurface\_MouseDoubleClick | private | void | Object, MouseEventArgs | Удаляет нажатую вершину или ребро. |
| DrawingSurface\_MouseDown | private | void | Object, MouseEventArgs | Если текущий инструмент – курсор, готовится двигать вершину. |
| DrawingSurface\_MouseMove | private | void | Object, MouseEventArgs | Если выбрана вершина для передвижения, двигает ее. |
| DrawingSurface\_MouseUp | private | void | Object, MouseEventArgs | В зависимости произошедшего ранее производит определенные действия в построении и редактировании графа. |
| EdgeButton\_Click | private | void | Object, EventArgs | Выбирает ластик как текущий инструмент. |
| EnlargeButton\_Click | private | void | Object, EventArgs | Увеличивает изображение графа. |
| ExitToolStripMenuItem\_Click | private | void | Object, EventArgs | Закрывает приложение. |
| FindArcVertices | private | bool | Object, EventArgs | Ищет выбранные для построения дуги вершины. |
| FolderBrowserDialogForGraphSaving | private | FolderBrowserDialog | – | Создает FolderBrowserDialog для полного сохранения графа. |
| GetChartTypes | private | SandpileChartType[] | – | Возвращает массив выбранных типов графиков для моделирования движения песка. |
| GetModelingModes | private | MovementModelingMode[] | – | Возвращает массив выбранных действий для моделирования. |
| GraphBuilder\_KeyDown | private | void | Object, KeyEventArgs | Осуществляет действия, связанные с конкретными клавишами. |
| GraphBuilder\_SizeChanged | private | void | Object, EventArgs | Подстраивает элементы окна под новый размер. |
| GridAdjacencyMatrix\_CellClick | private | void | Object, DataGridViewCellEventArgs | Выводит информацию о выбранной дуге в контролы, отвечающие за их редактирование. |
| GridParameters\_CellValueChanged | private | void | Object, DataGridViewCellEventArgs | Проверяет корректность нового значения и меняет соответствующий параметр. |
| Left\_Click | private | void | Object, EventArgs | Двигает граф влево. |
| MainMenuToolStripMenuItem\_Click | private | void | Object, EventArgs | Осуществляет переход к главному меню. |
| MainWindow | public | – | – | Конструктор класса. |
| MainWindow\_KeyUp | private | void | Object, KeyEventArgs | Вызывает команды движения графа, если клавиша изменила его положение или размер. |
| MainWindow\_MouseWheel | private | void | Object, MouseEventArgs | Меняет положение или размер графа по вращению колесика мыши. |
| Movement\_PreviewKeyDown | private | void | Object, PreviewKeyDownEventArgs | Переводит фокус, чтобы предотвратить некорректные действия по нажатию клавиши. |
| MovementEndedSandpileEventHandler | private | void | Object, EventArgs | Предлагает пользователю добавить песок в вершину после окончания движения. |
| MovementToolStripMenuItem\_Click | private | void | Object, EventArgs | Запускает моделирование движения. |
| NewProjectToolStripMenuItem\_Click | private | void | Object, EventArgs | Открывает пустой редактор для создания нового графа. |
| OkWeight\_Click | private | void | Object, EventArgs | Меняет длину дуги. |
| Open\_Click | private | void | Object, EventArgs | Открывает диалог для открытия графа из файла. |
| OpenProjectToolStripMenuItem\_Click | private | void | Object, EventArgs | Открывает диалог для открытия графа из файла. |
| RadiusTrackBar\_ValueChanged | private | void | Object, EventArgs | Меняет радиус вершин. |
| RandomAddingCheckBox\_CheckedChanged | private | void | Object, EventArgs | Меняет обработчик события окончания движения песка. |
| RandomAddingLabel\_Click | private | void | Object, EventArgs | Добавляет песчинку в случайную вершину. |
| RandomGraph\_Click | private | void | Object, EventArgs | Открывает окно, позволяющее построить случайный сильно связный граф. |
| RedoButton\_Click | private | void | Object, EventArgs | Возвращает последнее отмененное действие. |
| ReduceButton\_Click | private | void | Object, EventArgs | Уменьшает изображение графа. |
| RefreshVariables | private | void | – | Сбрасывает все произведенные изменения. |
| RemoveVertexFromGridAdjacencyMatrix | private | void | int | Убирает вершину из матрицы смежности. |
| RemoveVertexFromGridParameters | private | void | int | Убирает вершину из таблицы параметров. |
| ResetToolStripMenuItem\_Click | private | void | Object, EventArgs | Сбрасывает процесс моделирования и возвращает окно в состояние редактора. |
| Right\_Click | private | void | Object, EventArgs | Двигает граф вправо. |
| SandpileChartType1\_CheckedChanged | private | void | Object, EventArgs | Поддерживает хотя бы один из типов выбранным. |
| SandpileChartType2\_CheckedChanged | private | void | Object, EventArgs | Поддерживает хотя бы один из типов выбранным. |
| SandpilePalette\_SelectionChanged | private | void | Object, EventArgs | Снимает выбор ячейки. |
| SandpileTypeCheckBox\_CheckedChanged | private | void | Object, EventArgs | Поддерживает один из типов моделирования выбранным. |
| SaveAllToolStripMenuItem\_Click | private | void | Object, EventArgs | Открывает диалог для выбора папки для полного сохранения графа. |
| SaveFileDialogForDataSaving | private | SaveFileDialog | Object, EventArgs | Создает SaveFileDialog для сохранения данных графа. |
| SaveFileDialogForGifSaving | private | SaveFileDialog | Object, EventArgs | Создает SaveFileDialog для сохранения GIF-изображения. |
| SaveFileDialogForImageSaving | private | SaveFileDialog | Object, EventArgs | Создает SaveFileDialog для сохранения изображения графа. |
| SaveGif | private | void | Object, EventArgs | Открывает диалог для сохранения Gif-изображения. |
| SaveGraph | private | DialogResult | string, string | Открывает диалог для сохранения графа полностью. |
| SaveImageToolStripMenuItem\_Click | private | void | Object, EventArgs | Открывает диалог для сохранения изображения графа. |
| SelectStock | private | void | int, int | Ищет выбранную вершину и помечает ее как стоковую. |
| SelectVertexToAddSand | private | void | int, int | Ищет выбранную вершину и добавляет в нее песчинку. |
| SquareLattice\_Click | private | void | Object, EventArgs | Открывает окно для построения квадратной решетки. |
| StockLabel\_Click | private | void | Object, EventArgs | Запускает моделирование движения песка. |
| StopToolStripMenuItem\_Click | private | void | Object, EventArgs | Приостанавливает движение. |
| SubscribeToDigraphEvents | private | void | – | Подписывает обработчики на события класса Digraph. |
| TriangleLattice\_Click | private | void | Object, EventArgs | Открывает окно для построения треугольной решетки. |
| UndoButton\_Click | private | void | Object, EventArgs | Отменяет последнее действие в редакторе. |
| Up\_Click | private | void | Object, EventArgs | Двигает граф вверх. |
| UpdateDigraphInfo | private | void | – | Полностью обновляет все отображаемые данные графа. |
| UpdateElapsedTime | private | void | Object, MovementTickEventArgs | Обновляет прошедшее с начала движения время. |
| UpdateImage | private | void | – | Обновляет изображение графа. |
| VertexButton\_Click | private | void | Object, EventArgs | Выбирает инструмент добавления вершин как текущий. |
| VertexColorDialogOpen\_Click | private | void | Object, EventArgs | Открывает ColorDialog для выбора цвета отрисовки вершин. |
| VerticesColorPanel\_Click | private | void | Object, EventArgs | Переносит фокус на панель. |
| VerticesColorPanel\_Enter | private | void | Object, EventArgs | Делает кнопку открытия ColorDialog видимой. |
| VerticesColorPanel\_Leave | private | void | Object, EventArgs | Делает кнопку открытия ColorDialog невидимой. |
| WheelStopped | private | void | Object, EventArgs | Выполняет команду движения графа после остановки движения колесиком. |

Таблица 2.2

Описание полей методов и свойств структуры Arc.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Поля | | | | | |
| Имя | Модификатор доступа | Тип | Назначение | | |
| startVertex | private | int | Вершина начала дуги. | | |
| endVertex | private | int | Вершина конца дуги. | | |
| length | private | double | Длина дуги. | | |
| Методы | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| Arc | public | – | int, int, double | | Конструктор |
| ToString | public | string | – | | Возвращает строковое представление дуги. |
| Свойства | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| StartVertex | public | int | get/set | Вершина начала дуги. | |
| EndVertex | public | int | get/set | Вершина конца дуги. | |
| Length | public | double | get/set | Длина дуги. | |

Таблица 2.3

Описание полей методов и свойств структуры Vertex.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| X | public | int | X координата вершины | |
| Y | public | int | Y координата вершины | |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Vertex | public | – | int, int | Конструктор |

Таблица 2.4

Описание полей методов и свойств класса Digraph.cs

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Методы | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение |
| AddArc | public | void | Arc, int | | Добавляет ребро в конкретное место в списке. |
| AddVertex | public | void | Vertex, int, int, int, int | | Добавляет вершину в конкретное место в списке. |
| Digraph | public | – | – | | Конструктор |
| RemoveArc | public | void | int | | Удаляет ребро. |
| RemoveVertex | public | void | int | | Удаляет вершину. |
| ResetStock | public | void | – | | Очищает сток. |
| SetTimeTillTheEndOfRefractoryPeriod | public | void | – | | Устанавливает таймеры, отсчитывающие время до конца периода восстановления. |
| Свойства | | | | | |
| Имя | Модификатор доступа | Тип | Доступ | Назначение | |
| AdjacencyMatrix | public | double[,] | get | Матрица смежности. | |
| Arcs | public | List<Arc> | get/set | Список дуг. | |
| RefractoryPeriods | public | List<int> | get/set | Список периодов восстановления вершин. | |
| State | public | List<int> | get/set | Список начальных состояний вершин. | |
| Stock | public | List<int> | get/set | Список индексов стоковых вершин | |
| Thresholds | public | List<int> | get/set | Список порогов вершин. | |
| TimeTillTheEndOfRefractoryPeriod | public | List<Timer> | get/set | Список таймеров, отсчитывающих время до конца периода восстановления. | |
| Vertices | public | List<Vertex> | get/set | Список вершин. | |

Таблица 2.5

Описание полей методов и свойств класса DigraphChangedEventArgs.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| Index | public | int | Индекс удаленного элемента. | |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| DigraphChangedEventArgs | public | – | int | Конструктор |

Таблица 2.6

Описание полей методов и свойств интерфейса ICommand.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Execute | – | void | – | Выполняет команду. |
| UnExecute | – | void | – | Отменяет команду. |

Таблица 2.7

Описание полей методов и свойств класса AddArcCommand.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| arc | private | Arc | Добавляемая дуга. | |
| digraph | private | Digraph | Орграф, с которым ведется работа. | |
| index | private | int | Индекс добавляемой дуги. | |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| AddArcCommand | public | – | Digraph, Arc | Конструктор |
| Execute | public | void | – | Добавляет дугу в граф. |
| UnExecute | public | void | – | Удаляет добавленную дугу. |

Таблица 2.8

Описание полей методов и свойств класса AddVertexCommand.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| digraph | private | Digraph | Орграф, с которым ведется работа. | |
| index | private | int | Индекс добавляемой вершины. | |
| refractoryPeriod | private | int | Период восстановления вершины. | |
| state | private | int | Начальное состояние вершины. | |
| threshold | private | int | Порог вершины. | |
| vertex | private | Vertex | Добавляемая вершина. | |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| AddVertexCommand | public | – | Digraph, Vertex | Конструктор |
| Execute | public | void | – | Добавляет вершину в граф. |
| UnExecute | public | void | – | Удаляет добавленную вершину. |

Таблица 2.9

Описание полей методов и свойств класса ChangeArcLengthCommand.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| digraph | private | Digraph | Орграф, с которым ведется работа. | |
| index | private | int | Индекс изменяемой дуги. | |
| newValue | private | double | Новая длина дуги. | |
| oldValue | private | double | Старая длина дуги. | |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| ChangeArcLengthCommand | public | – | Digraph, int, double, double | Конструктор |
| Execute | public | void | – | Изменяет длину дуги. |
| UnExecute | public | void | – | Возвращает старую длину дуги. |

Таблица 2.10

Описание полей методов и свойств класса ChangeColorCommand.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| newColor | private | Color | Новый цвет. | |
| oldColor | private | Color | Старый цвет. | |
| target | private | GraphDrawing | Объект, производящий отрисовку графа. | |
| type | private | Type | Тип объекта, цвет которого меняется. | |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| ChangeColorCommand | public | – | GraphDrawing, Type, Color, Color | Конструктор |
| Execute | public | void | – | Изменяет цвет отрисовки компонента графа. |
| UnExecute | public | void | – | Возвращает старый цвет отрисовки. |

Таблица 2.11

Описание полей методов и свойств класса EraseArcCommand.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| arc | private | Arc | Удаляемая дуга. | |
| digraph | private | Digraph | Орграф, с которым ведется работа. | |
| index | private | int | Индекс удаляемой дуги. | |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| EraseArcCommand | public | – | Digraph, Arc | Конструктор |
| Execute | public | void | – | Удаляет дугу из графа. |
| UnExecute | public | void | – | Восстанавливает дугу. |

Таблица 2.12

Описание полей методов и свойств класса EraseVertexCommand.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| arcIndices | private | List<int> | Список индексов инцидентных дуг. | |
| digraph | private | Digraph | Орграф, с которым ведется работа. | |
| incidentArcs | private | List<Arc> | Список инцидентных дуг. | |
| index | private | int | Индекс удаляемой вершины. | |
| refractoryPeriod | private | int | Период восстановления вершины. | |
| state | private | int | Начальное состояние вершины. | |
| threshold | private | int | Порог вершины. | |
| vertex | private | Vertex | Удаляемая вершина. | |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| EraseVertexCommand | public | – | Digraph, Vertex | Конструктор |
| Execute | public | void | – | Удаляет вершину из графа. |
| UnExecute | public | void | – | Восстанавливает вершину. |

Таблица 2.13

Описание полей методов и свойств класса MoveDigraphCommand.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| digraph | private | Digraph | Орграф, с которым ведется работа. | |
| xCoefficient | private | int | Коэффициент смещения по оси X. | |
| yCoefficient | private | int | Коэффициент смещения по оси Y. | |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Execute | public | void | – | Двигает граф. |
| MoveDigraphCommand | public | – | Digraph, int, int | Конструктор |
| UnExecute | public | void | – | Возвращает граф на прежнее место. |

Таблица 2.14

Описание полей методов и свойств класса MoveVertexCommand.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| digraph | private | Digraph | Орграф, с которым ведется работа. | |
| index | private | int | Индекс двигаемой вершины. | |
| oldPoint | private | Point | Старые координаты вершины. | |
| newPoint | private | Point | Новые координаты вершины. | |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Execute | public | void | – | Двигает вершину. |
| MoveVertexCommand | public | – | Digraph, int, Point, Point | Конструктор |
| UnExecute | public | void | – | Возвращает вершину на прежнюю точку. |

Таблица 2.15

Описание полей методов и свойств класса ResizeDigraphCommand.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| coefficient | private | double | Коэффициент изменения размера. | |
| digraph | private | Digraph | Орграф, с которым ведется работа. | |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Execute | public | void | – | Меняет размер графа. |
| ResizeDigraphCommand | public | – | Digraph, double | Конструктор |
| UnExecute | public | void | – | Возвращает графу прежний размер. |

Таблица 2.16

Описание полей методов и свойств структуры CommandsManager.cs

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Методы | | | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | | Назначение | |
| CommandsManager | public | – | – | | Конструктор | |
| Execute | public | void | ICommand | | Выполняет команду и помещает ее в стек Undo. | |
| Redo | public | void | – | | Возвращает последнюю отмененную команду. | |
| Undo | public | void | – | | Отменяет последнюю команду. | |
| Свойства | | | | | | |
| Имя | Модификатор доступа | Тип | | Доступ | | Назначение |
| CanRedo | public | bool | | get | | Указывает, есть ли команды в стеке Redo. |
| CanUndo | public | bool | | get | | Указывает, есть ли команды в стеке Undo. |
| RedoStack | private | Stack<ICommand> | | get | | Стек команд для отмены. |
| UndoStack | private | Stack<ICommand> | | get | | Стек команд для возврата. |

Таблица 2.17

Описание полей методов и свойств класса ConnectivityCheck.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля |  |  |  | |
| Имя | Модификатор доступа | Тип | Назначение | |
| adjacencyList | private | List<int> | Список смежности графа. | |
| numberOfVertices | private | int | Количество вершин графа. | |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| AddArcs | public | void | Arc | Добавляет дугу. |
| ConnectivityCheck | public | – | int | Конструктор. |
| DFS | private | void | Int, bool[] | Поиск в глубину и запоминание посещенных вершин. |
| GetInvertedGraph | private | ConnectivityCheck | – | Инвертирует граф. |
| IsGraphValid | public | bool | Digraph | Определяет валидность графа для моделирования движения точек. |
| IsStronglyConnected | public | bool | – | Проверят граф на сильную связность. |

Таблица 2.18

Описание полей методов и свойств класса GraphDrawing.cs

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Поля | |  | |  | |  | | |
| Имя | | Модификатор доступа | | Тип | | Назначение | | |
| \_radius | | private | | int | | Радиус вершин. | | |
| arcsPen | | private | | Pen | | Используется для отрисовки дуг. | | |
| brush | | private | | Brush | | Используется для отрисовки надписей. | | |
| graphics | | private | | Graphics | | Используется для рисования фигур. | | |
| font | | private | | Font | | Шрифт нумерации вершин. | | |
| highlightPen | | private | | Pen | | Используется для выделения вершин. | | |
| highlightSandpilePen | | private | | Pen | | Используется для выделения вершин. | | |
| incidenceList | | private | | List<Arc>[] | | Список смежности графа. | | |
| sandpileFont | | private | | Font | | Шрифт, использующийся при отрисовки графа в процессе моделирования движения песка. | | |
| sandpilePalette | | private | | Color[] | | Палитра цветов для обозначения состояния вершин в процессе моделирования движения песка. | | |
| verticesPen | | private | | Pen | | Используется для отрисовки вершин. | | |
| Методы | | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| ClearTheSurface | | public | | void | | – | | Очищает изображение. |
| Dispose | | public | | void | | – | | Освобождает все используемые классом освобождаемые объекты |
| DrawArc | | public | | void | | Vertex, Vertex, Arc, int, int, double | | Рисует дугу. |
| DrawDot | | public | | void | | PointF | | Рисует точку на графе. |
| DrawGraph | | public | | void | | Digraph, int, int double | | Полностью рисует граф. |
| DrawGraphSandpile | | public | | void | | Digraph, bool, int, int double | | Полностью рисует граф, используя Sandpile палитру. |
| DrawVertex | | public | | void | | int, int, int, Pen | | Рисует вершину. |
| DrawVertices | | public | | void | | Digraph, int, int double | | Рисует все вершины графа. |
| DrawVerticesSandpile | | public | | void | | Digraph, int, int double | | Рисует все вершины графа, используя Sandpile палитру. |
| GetGradientColors | | private | | Color[] | | Color, Color, int | | Возвращает массив градиентных цветов заданного размера с заданным начальным и конечным цветом. |
| GetSandpilePalette | | private | | Color[] | | int | | Возвращает палитру цветов для отрисовки графа в процессе моделирования движения песка. |
| GraphDrawing | | public | | – | | int, int | | Конструктор. |
| HighlightVertex | | public | | void | | Vertex | | Подсвечивает вершину. |
| HighlightVertexToAddSand | | public | | void | | Vertex | | Подсвечивает вершину, выбранную для добавления песка. |
| UnhighlightVertex | | public | | void | | Vertex | | Снимает выделение с вершины. |
| Свойства | | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Доступ | | Назначение | |
| ArcsColor | public | | Color | | get/set | | Цвет отрисовки дуг. | |
| BackColor | public | | Color | | get/set | | Цвет фона. | |
| Image | public | | Bitmap | | get/set | | Созданное изображение. | |
| R | public | | int | | get/set | | Радиус вершины. | |
| SandpilePalette | public | | Color[] | | get/set | | Палитра цветов для обозначения состояния вершин в процессе моделирования движения песка. | |
| Size | public | | Size | | get/set | | Размер изображения | |
| VerticesColor | public | | Color | | get/set | | Цвет отрисовки вершин. | |

Таблица 2.19

Описание полей методов и свойств класса ChartWindow.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| AvalancheSizesDistributionChartPrepare | public | void | – | Меняет настройки соответствующих контролов для отображения графика распределения лавин. |
| ChartWindow | public | – | – | Конструктор. |
| SaveAll\_Click | private | void | Object, EventArgs | Открывает диалог для сохранения графика целиком. |
| SaveData\_Click | private | void | Object, EventArgs | Открывает диалог для сохранения данных графика. |
| SaveImage\_Click | private | void | Object, EventArgs | Открывает диалог для сохранения изображения графика. |

Таблица 2.18

Описание полей методов и свойств класса MovementModeling.cs

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Поля | |  | |  | |  | | |
| Имя | | Модификатор доступа | | Тип | | Назначение | | |
| avalancheSize | | private | | int | | Размер текущей лавины. | | |
| digraph | | private | | Digraph | | Орграф, с которым ведется работа. | | |
| distributionChart | | private | | ChartWindow | | Окно с графиком распределения лавин. | | |
| DrawingSurface | | public | | PictureBox | | Поверхность для отображения анимации. | | |
| GraphDrawing | | public | | GraphDrawing | | Используется для отрисовки графа. | | |
| incidenceList | | private | | List<Arc>[] | | Список инцидентности. | | |
| involvedArcs | | private | | List<Arc> | | Список вершин, по которым движутся точки. | | |
| mainStopwatch | | private | | Stopwatch | | Отсчитывает время процесса. | | |
| mainTimer | | private | | Timer | | По тику таймера осуществляется обновление. | | |
| modes | | private | | MovementModelingMode[] | | Массив дополнительных действий, необходимых при моделировании. | | |
| MovementGif | | public | | GifBitmapEncoder | | Используется для отрисовки вершин. | | |
| numberOfDotsChart | | private | | ChartWIndow | | Окно с графиком изменения числа точек. | | |
| releaseCondition | | private | | Predicate<int> | | Условие выпускания вершинами точек. | | |
| SandpileChartTypes | | public | | SandpileChartType[] | | Массив необходимых для построения типов графика. | | |
| speed | | private | | Double | | Скорость точек. | | |
| stateChange | | private | | Action<int> | | Изменение состояния вершины после выпуска точек. | | |
| stopwatches | | private | | List<Stopwatch> | | Указывают время, прошедшее с момента выпуска конкретной точки. | | |
| type | | private | | MovementModelingType | | Тип моделирования. | | |
| Методы | | | | | | | | |
| Имя | | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| AddAvalancheSize | | private | | void | | – | | Очищает изображение. |
| AddNumberOfDotsChartPoint | | private | | void | | – | | Освобождает все используемые классом освобождаемые объекты |
| ChangeChartInterval | | private | | void | | Vertex, Vertex, Arc, int, int, double | | Рисует дугу. |
| CheckDotsNumber | | private | | void | | PointF | | Рисует точку на графе. |
| GetIncidenceList | | public | | List<Arc>[] | | Digraph, int, int double | | Полностью рисует граф. |
| GetPoint | | public | | PointF | | Digraph, bool, int, int double | | Полностью рисует граф, используя Sandpile палитру. |
| GetTime | | public | | double | | int, int, int, Pen | | Рисует вершину. |
| Go | | public | | void | | Digraph, int, int double | | Рисует все вершины графа. |
| MovementModeling | | public | | void | | Digraph, int, int double | | Рисует все вершины графа, используя Sandpile палитру. |
| PrepareBasicCharts | | private | | void | | Color, Color, int | | Возвращает массив градиентных цветов заданного размера с заданным начальным и конечным цветом. |
| PrepareSandpileCharts | | private | | void | | int | | Возвращает палитру цветов для отрисовки графа в процессе моделирования движения песка. |
| ProcessDots | | private | | void | | int, int | | Конструктор. |
| ProcessVertices | | private | | void | | Vertex | | Подсвечивает вершину. |
| ReleaseDots | | private | | void | | Vertex | | Подсвечивает вершину, выбранную для добавления песка. |
| StartMovementModeling | | public | | void | | Vertex | | Снимает выделение с вершины. |
| StartNewTimers | | private | | void | |  | |  |
| Stop | | public | | void | |  | |  |
| TickAddFrame | | private | | void | |  | |  |
| TickModeling | | private | | void | |  | |  |
| UpdateChart | | private | | void | |  | |  |
| Свойства | | | | | | | | |
| Имя | Модификатор доступа | | Тип | | Доступ | | Назначение | |
| ArcsColor | public | | Color | | get/set | | Цвет отрисовки дуг. | |
| BackColor | public | | Color | | get/set | | Цвет фона. | |
| Image | public | | Bitmap | | get/set | | Созданное изображение. | |
| R | public | | int | | get/set | | Радиус вершины. | |
| SandpilePalette | public | | Color[] | | get/set | | Палитра цветов для обозначения состояния вершин в процессе моделирования движения песка. | |
| Size | public | | Size | | get/set | | Размер изображения | |
| VerticesColor | public | | Color | | get/set | | Цвет отрисовки вершин. | |

Таблица 1.2

Описание полей методов и свойств класса RandomDigraphGeneratorForm.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля |  |  |  | |
| Имя | Модификатор доступа | Тип | Назначение | |
| height | private | int | Высота поля отображения графа. | |
| Rnd | private | Random | Генератор случайных значений. | |
| width | private | int | Ширина поля отображения графа. | |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Button2\_Click | private | void | Object, EventArgs | Генерирует случайный сильно связный граф с выбранным количеством вершин. |
| Cancel\_Click | private | void | Object, EventArgs | Закрывает окно. |
| RandomGraphGeneratorForm | public | – | int, int | Конструктор класса |
| VNRandom\_Click | private | void | Object, EventArgs | Присваивает количеству вершин случайное значение. |
| Свойства | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| Digraph | public | Digraph | Сгенерированный граф. | |

Таблица 1.3

Описание полей методов и свойств класса SquareLatticeForm.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля |  |  |  | |
| Имя | Модификатор доступа | Тип | Назначение | |
| \_rnd | private | Random | Генератор случайных значений. | |
| height | private | int | Высота поля отображения графа. | |
| width | private | int | Ширина поля отображения графа. | |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| AddArcs | private | void | – | Добавляет дуги в формируемую решетку. |
| AddVertices | private | void | – | Добавляет вершины в формируемую решетку. |
| Cancel\_Click | private | void | Object, EventArgs | Закрывает окно. |
| OK\_Click | private | void | Object, EventArgs | Строит квадратную решетку заданных размеров. |
| ParamsCheckBox\_CheckedChanged | private | void | Object, EventArgs | Определяет необходимость заполнения параметров графа случайными значениями. |
| SquareLatticeForm | public | – | int, int | Конструктор класса |
| Свойства | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| SquareLatticeDigraph | public | Digraph | Построенная квадратная решётка. | |

Таблица 1.4

Описание полей методов и свойств класса TriangularLatticeForm.cs

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Поля |  |  |  | |
| Имя | Модификатор доступа | Тип | Назначение | |
| \_rnd | private | Random | Генератор случайных значений. | |
| height | private | int | Высота поля отображения графа. | |
| width | private | int | Ширина поля отображения графа. | |
| Методы | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| AddArcs | private | void | – | Добавляет дуги в формируемую решетку. |
| AddVertices | private | void | – | Добавляет вершины в формируемую решетку. |
| Cancel\_Click | private | void | Object, EventArgs | Закрывает окно. |
| OK\_Click | private | void | Object, EventArgs | Строит треугольную решетку заданных размеров. |
| ParamsCheckBox\_CheckedChanged | private | void | Object, EventArgs | Определяет необходимость заполнения параметров графа случайными значениями. |
| TriangularLatticeForm | public | – | int, int | Конструктор класса |
| Свойства | | | | |
| Имя | Модификатор доступа | Тип | Назначение | |
| TriangularLatticeDigraph | public | Digraph | Построенная квадратная решётка. | |